



HIGH PERFORMANCE COMPUTING 2010 TECHNOLOGY COMPASS

TECHNOLOGY COMPASS

TABLE OF CONTENTS AND INTRODUCTION

HIGH PERFORMANCE COMPUTING	PAGE 4
Performance Turns Into Productivity	Page 6
CLUSTER MANAGEMENT MADE EASY	PAGE 10
Moab Cluster Suite	Page 12
Moab Grid Suite	Page 16
Moab Adaptive HPC Suite	Page 20
INTEL CLUSTER READY	PAGE 22
A Quality Standard for HPC Clusters	Page 24
Intel Cluster Ready Builds HPC Momentum	Page 28
The transtec Benchmarking Center	Page 32
WINDOWS HPC SERVER 2008 R2	PAGE 34
Elements of the Microsoft HPC Solution	Page 36
Deployment, system management, and monitoring	Page 38
Job Scheduling	Page 40
Service-Oriented Architecture	Page 42
Networking and MPI	Page 44
Microsoft Office Excel Support	Page 46
PARALLEL NFS	PAGE 50
The New Standard for HPC Storage	Page 52
Panasas HPC Storage	Page 56
NVIDIA GPU COMPUTING	PAGE 60
The CUDA Architecture	Page 62
Codename "Fermi"	Page 66
Introducing NVIDIA Nexus	Page 72
GLOSSARY	PAGE 74

30 YEARS OF EXPERIENCE IN SCIENTIFIC COMPUTING

1980 marked the beginning of a decade where numerous startups were created, some of which later transformed into big players in the IT market. Technical innovations brought dramatic changes to the nascent computer market. In Tübingen, close to one of Germany's prime and oldest universities, transtec was founded.

In the early days, transtec focused on reselling DEC computers and peripherals, delivering high-performance workstations to university institutes and research facilities. In 1987, SUN/Sparc and storage solutions broadened the portfolio, enhanced by IBM/RS6000 products in 1991. These were the typical workstations and server systems for high performance computing then, used by the majority of researchers worldwide.

In the late 90s, transtec was one of the first companies to offer highly customized HPC cluster solutions based on standard Intel architecture servers, some of which entered the TOP500 list of the world's fastest computing systems.

Thus, given this background and history, it is fair to say that transtec looks back upon a 30 years' experience in scientific computing; our track record shows nearly 400 HPC cluster installations. With this experience, we know exactly what customers' demands are and how to meet them. High performance and ease of management – this is what customers require today. HPC systems are for sure required to peak-perform, as their name indicates, but that is not enough: they must also be easy to handle. Unwieldy design and operational complexity must be avoided or at least hidden from administrators and particularly users of HPC computer systems.

transtec HPC solutions deliver ease of management, both in the Linux and Windows worlds, and even where both worlds live in an integrated fashion! With Moab Adaptive HPC Suite, transtec is able to provide a combined, flexible, and dynamical "Linux plus Windows" solution for HPC environments wherever needed.

transtec HPC systems use the latest and most innovative technology. Their superior performance goes hand in hand with energy efficiency, as you would expect from any leading edge IT solution. We regard these basic characteristics.

This brochure focusses on where transtec HPC solutions excel. To name a few: Intel Cluster Ready certification as an independent quality standard for our systems, Panasas HPC storage systems for highest performance and best scalability required of an HPC storage system. Again, with these solutions, usability and ease of management are central issues that are addressed. Also, being NVIDIA Tesla Preferred Provider, transtec is able to provide customers with well-designed, extremely powerful solutions for Tesla GPU computing.

Last but not least, your decision for a transtec HPC solution means you opt for most intensive customer care and best service in HPC. Our experts will be glad to bring in their expertise and support to assist you at any stage, from HPC design to daily cluster operations.

Have fun reading the transtec HPC Compass 2010!



HIGH PERFORMANCE COMPUTING PERFORMANCE TURNS INTO PRODUCTIVITY

High Performance Computing (HPC) has been with us from the very beginning of the computer era. High-performance computers were built to solve numerous problems which the “human computers” could not handle. The term HPC just hadn’t been coined yet. More important, some of the early principles have changed fundamentally.

HPC systems in the early days were much different from those we see today. First, we saw enormous mainframes from large computer manufacturers, including a proprietary operating system and job management system. Second, at universities and research institutes, workstations made inroads and scientists carried out calculations on their dedicated Unix or VMS workstations. In either case, if you needed more computing power, you scaled up, i.e. you bought a bigger machine.

Today the term High-Performance Computing has gained a fundamentally new meaning. HPC is now perceived as a way to tackle complex mathematical, scientific or engineering problems. The integration of industry standard, “off-the-shelf” server hardware into HPC clusters facilitates the construction of computer networks of such power that one single system could never achieve. The new paradigm for parallelization is scaling out.

HIGH PERFORMANCE COMPUTING

PERFORMANCE TURNS INTO PRODUCTIVITY

Computer-supported simulations of realistic processes (so-called Computer Aided Engineering – CAE) has established itself as a third key pillar in the field of science and research alongside theory and experimentation. It is nowadays inconceivable that an aircraft manufacturer or a Formula One racing team would operate without using simulation software. And scientific calculations, such as in the fields of astrophysics, medicine, pharmaceuticals and bio-informatics, will to a large extent be dependent on supercomputers in the future. Software manufacturers long ago recognized the benefit of high-performance computers based on powerful standard servers and ported their programs to them accordingly.

The main advantages of scale-out supercomputers is just that: they are infinitely scalable, at least in principle. Since they are based on standard hardware components, such a supercomputer can be charged with more power whenever the computational capacity of the system is not sufficient any more, simply by adding additional nodes of the same kind. A cumbersome switch to a different technology can be avoided in most cases.

The primary rationale in using HPC clusters is to grow, to scale out computing capacity as far as necessary. To reach that goal, an HPC cluster returns most of the invest when it is continuously fed with computing problems.

The secondary reason for building scale-out supercomputers is to maximize the utilization of the system.

“transtec HPC solutions offer leading edge performance and energy efficiency. Apart from that, deciding for a transtec HPC solution means deciding for the most intensive customer care and the best service imaginable”

Dr. Oliver Tennert Director Marketing & HPC Solutions



VARIATIONS ON THE THEME: MPP AND SMP

Parallel computations exist in two major variants today. Applications running in parallel on multiple compute nodes are frequently so-called Massively Parallel Processing (MPP) applications. MPP indicates that the individual processes can each utilize exclusive memory areas. This means that such jobs are predestined to be computed in parallel, distributed across the nodes in a cluster. The individual processes can thus utilize the separate units of the respective node – especially the RAM, the CPU power and the disk I/O.

Communication between the individual processes is implemented in a standardized way through the MPI software interface (Message Passing Interface), which abstracts the underlying network connections between the nodes from the processes. However, the MPI standard (current version 2.0) merely requires source code compatibility, not binary compatibility, so an off-the-shelf application usually needs specific versions of MPI libraries in order to run. Examples of MPI implementations are OpenMPI, MPICH2, MVAPICH2, Intel MPI or – for Windows clusters – MS-MPI.

If the individual processes engage in a large amount of communication, the response time of the network (latency) becomes important. Latency in a Gigabit Ethernet or a 10GE network is typically around 10 μ s. High-speed interconnects such as InfiniBand, reduce latency by a factor of 10 down to as low as 1 μ s. Therefore, high-speed interconnects can greatly speed up total processing.

The other frequently used variant is called SMP applications. SMP, in this HPC context, stands for Shared Memory Processing. It involves the use of shared memory areas, the specific implementation of which is dependent on the choice of the underlying operating system. Consequently, SMP jobs generally only run on a single node, where they can in turn be multi-threaded and thus be parallelized across the number of CPUs per node. For many HPC applications, both the MPP and SMP variant can be chosen.

Many applications are not inherently suitable for parallel execution. In such a case, there is no communication between the individual compute nodes, and therefore no need for a high-speed network between them; nevertheless, multiple computing jobs can be run simultaneously and sequentially on each individual node, depending on the number of CPUs.

In order to ensure optimum computing performance for these applications, it must be examined how many CPUs and cores deliver the optimum performance.

We find applications of this sequential type of work typically in the fields of data analysis or Monte-Carlo simulations.

HIGH PERFORMANCE COMPUTING

PERFORMANCE TURNS INTO PRODUCTIVITY

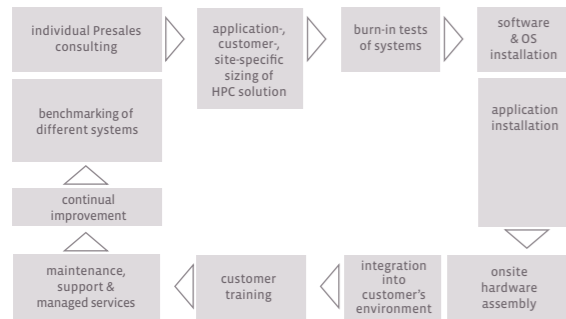
HIGH PERFORMANCE MEETS EFFICIENCY

Initially, massively parallel systems constitute a challenge to both administrators and users. They are complex beasts. Anyone building HPC clusters will need to tame the beast, master the complexity and present users and administrators with an easy-to-use, easy-to-manage system landscape.

Leading HPC solution providers such as transtec achieve this goal. They hide the complexity of HPC under the hood and match high performance with efficiency and ease-of-use for both users and administrators. The “P” in “HPC” gains a double meaning: “Performance” plus “Productivity”.

Cluster Management Software like Moab Cluster Suite, Moab Grid Suite, and Moab Adaptive HPC Suite provide the means to master and hide the inherent complexity of HPC systems. For administrators and users, HPC clusters are presented as single, large machines, with many different tuning parameters. The software also provides a unified view of existing clusters whenever unified management is added as a requirement by the customer at any point in time after the first installation. Thus, daily routine tasks such as job management, user management, queue partitioning and management, can be performed easily with either graphical or web-based tools, without any advanced scripting skills or technical expertise required from the administrator or user.

FIGURE 1 SERVICES AND CUSTOMER CARE FROM A TO Z



HPC @ TRANSTEC: SERVICES AND CUSTOMER CARE FROM A TO Z

transtec AG has 30 years of experience in scientific computing and is one of the earliest manufacturers of HPC clusters. For nearly a decade, transtec has delivered highly customized High Performance clusters based on standard components to academic and industry customers across Europe with all the high quality standards and the customer-centric approach that transtec is well known for.

Every transtec HPC solution is more than just a rack full of hardware – it is a comprehensive solution with everything the HPC user, owner, and operator need.

In the early stages of any customer's HPC project, transtec experts provide extensive and detailed consulting to the customer – they benefit from expertise and experience. Consulting is followed by benchmarking of different systems with either specifically crafted customer code or generally accepted benchmarking routines; this aids customers in sizing and devising the optimal and detailed HPC configuration.

Each and every piece of HPC hardware that leaves our factory undergoes a burn-in procedure of 24 hours or more if necessary. We make sure that any hardware shipped

meets our and our customers' quality requirements. transtec HPC solutions are turnkey solutions. By default, a transtec HPC cluster has everything installed and configured – from hardware and operating system to important middleware components like cluster management or developer tools and the customer's production applications. Onsite delivery means onsite integration into the customer's production environment, be it establishing network connectivity to the corporate network, or setting up software and configuration parts.

transtec HPC clusters are ready-to-run systems – we deliver, you turn the key, the system delivers high performance.

Every HPC project entails transfer to production: IT operation processes and policies apply to the new HPC system. Effectively, IT personnel is trained hands-on, introduced to hardware components and software, with all operational aspects of configuration management.

transtec services do not stop when the implementation projects ends. Beyond transfer to production, transtec takes care. transtec offers a variety of support and service options, tailored to the customer's needs. When you are in need of a new installation, a major reconfiguration or an update of your solution – transtec is able to support your staff and, if you lack the resources for maintaining the cluster yourself, maintain the HPC solution for you. From Professional Services to Managed Services for daily operations and required service levels, transtec will be your complete HPC service and solution provider. transtec's high standards of performance, reliability and dependability assure your productivity and complete satisfaction.



CLUSTER MANAGEMENT MADE EASY

If the administration and operation involved in an IT infrastructure were proportional to the number of systems, no one would consider purchasing an HPC cluster. From the beginnings of Beowulf clustering to today, a wide range of cluster management solutions have been developed that make HPC clustering manageable even for companies who have just one single, average-experienced administrator.

CLUSTER MANAGEMENT MADE EASY

MOAB CLUSTER SUITE

FIGURE 1 AUTOMATING TASKS, POLICIES, AND REPORTING

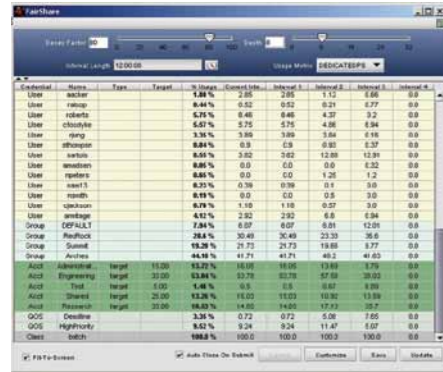


FIGURE 2 CONTROLS FOR TRUSTED MULTI-GROUP SHARING

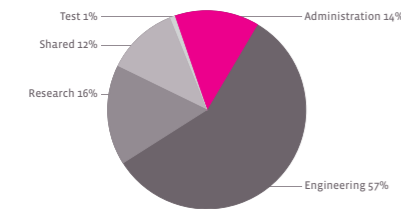
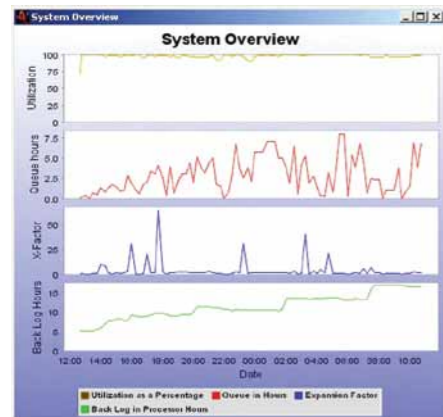


FIGURE 3 IMPROVE PERFORMANCE WITH 90-99% UTILIZATION



MOAB CLUSTER SUITE

Moab Cluster Suite is an intelligent management middleware that provides simple web-based job management, graphical cluster administration and management reporting tools. Organizations will benefit from the ability to provide guaranteed service levels to users and organizations, higher resource utilization rates, and the ability to get more jobs processed with the same resources, resulting in an improved ROI.

By using Moab, sites can dynamically change HPC and data center resource pools on the fly to match specific workload needs. Moab provides flexible management policies to ensure that specific user, group and workload needs are met. Moab also enforces Quality of Service and SLA guarantees and makes certain that high-level company objectives are achieved.

Moab Cluster Suite incorporates the following industry leading applications from Adaptive Computing:

- Moab Workload Manager**
 A policy-based workload management and scheduling engine
- Moab Cluster Manager**
 A powerful graphical cluster administration interface, monitor, and reporting tool
- Moab Access Portal**
 A web-based end-user job submission and management portal

BUSINESS BENEFITS

- Integrate/Unify management across diverse resources and environments in a cluster

- Control/Share resource usage between users, groups and projects
- Simplify use, access and control for both users and administrators
- Track, diagnose and report on cluster workload and status information
- Automate tasks and processes to streamline job turnaround and reduce administrative burden
- Scalable architecture that is grid ready, compatible and extensible

SYSTEM COMPATIBILITY

- Operating system support for Linux (all), UNIX (AIX, IRIX, HP-UX, FreeBSD, OSF/Tru-64, Solaris, etc.), Mac OSX & limited Windows support
- Resource Manager support for LSF, TORQUE, PBSPro, SGE, SLURM, LoadLeveler, OpenPBS, and custom resource managers

KEY CAPABILITIES

Improve Performance with 90-99% Utilization

- Achieve faster job response times as a result of jobs being optimally placed to run based on real-time workload conditions and rules
- Reach higher and more consistent utilization of resource capacity and capabilities through intelligent scheduling, precise policy controls and high availability services
- Guarantee that jobs run at required times with advance reservations
- Ensure the most important work receives the highest priority and quality of service

Gain Control with Automated Tasks, Policies, Reporting

- Automate administrative tasks and response with custom job, node or system-wide triggers based on any event or condition criteria
- Use event, condition and time-based policies in a flexible policy engine to ensure usage matches set levels and priorities
- Identify problems, utilization and ROI easily with precise custom reports that provide centralized and visual status reporting on current and historical jobs
- Use resource consumption reports to effectively share costs of cluster maintenance or to drive self-management across users, groups, and organizations
- Complete tasks quicker with a task-based interface, reusable job templates and group operations to streamline changes across groups of users and resources

Controls for Trusted Multi-Group Sharing

- Share resources fairly with policies that ensure agreed upon service levels are delivered to groups and users based on time, capacity, priority and other settings
- Build trust with resource owners by graphically reporting on individually purchased and shared resource usage and enforcement of resource guarantees through service level policies
- Ensure no one exceeds intended usage by applying soft or hard limits
- Encourage groups and users to self-manage resource usage with built-in accounting capabilities that track credit, time or cost usage against maximums
- Simplify administration of access rights with role-based authorization levels and visual maps of user, group and quality of service relationships and settings

CLUSTER MANAGEMENT MADE EASY

MOAB CLUSTER SUITE

FIGURE 4 INCREASE USER PRODUCTIVITY



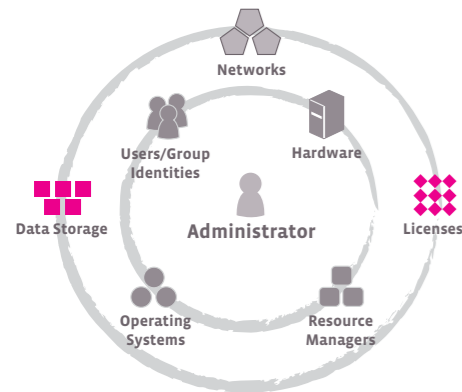
Increase User Productivity

- II Increase productivity by allowing end-users to submit jobs from anywhere via a portal
- II Reduce end-user job management training with an easy-to-use web interface and self-help capabilities such as start time estimates and visual reservation tools
- II Simplify and speed up job submission with basic and advanced job forms, reusable and shared job templates and the ability to browse for needed local and server files
- II Enable users to self-manage usage based on credits left or built-in user reports

Unify Management Across Clusters

- II Unify workload management across existing resource managers, networks and hardware and connect with databases, provisioning systems, portals, allocation managers and other applications for full end-to-end management and integration
- II Eliminate duplicate manual administration work across multiple clusters, freeing valuable staff resources to work on other high priority projects
- II Get grid ready with local area grid support out-of-the-box and support for wide area grids (clusters with non-shared user and data spaces) with Moab Grid Suite
- II Enable future growth with scalability to tens of thousands of diverse cluster nodes

FIGURE 5 UNIFY MANAGEMENT ACROSS CLUSTERS



transtec HPC cluster solutions are designed for maximum flexibility, and ease of management. We not only offer our customers the most powerful and flexible cluster management solution out there, but also provide them with customized setup and site-specific configuration. Whether a customer needs a dynamical Linux-Windows dual-boot solution, unified management of different clusters at different sites, or the fine-tuning of the Moab scheduler for implementing fine-grained policy configuration – transtec not only gives you the framework at hand, but also helps you adapt the system according to your special needs. Needless to say, when customers are in need of special trainings, transtec will be there to provide customers, administrators, or users with specially adjusted Educational Services.

CLUSTER MANAGEMENT MADE EASY

MOAB GRID SUITE

FIGURE 6 PROCESS MORE WORK IN LESS TIME TO MAXIMIZE ROI

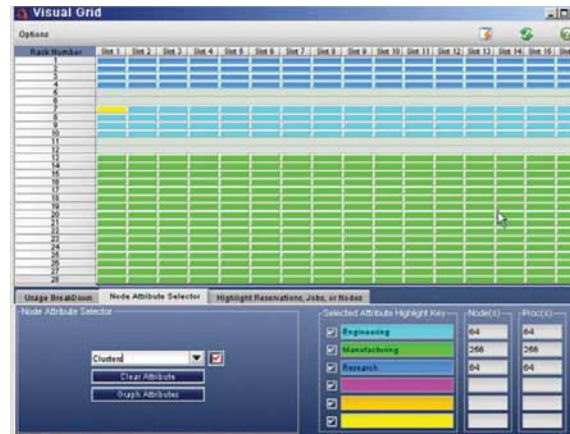
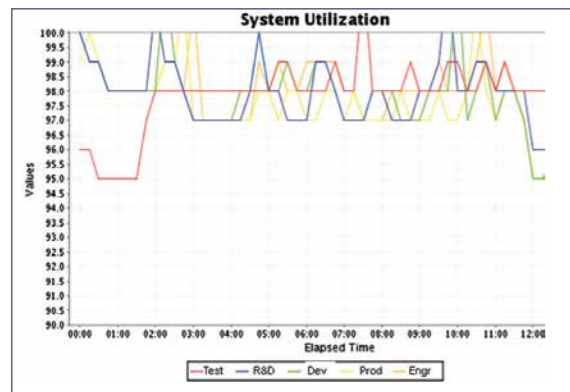


FIGURE 7 AUTOMATING TASKS, POLICIES, REPORTING



MOAB GRID SUITE

Moab Grid Suite is a powerful grid workload management solution that includes scheduling, advanced policy management, and tools to control all components of today's advanced grids. Unlike other "grid" solutions, Moab Grid Suite truly delivers the ability to connect disparate clusters into a logical whole, enabling grid administrators and grid policies to have reign over all systems, while preserving the sovereignty and control at the individual cluster.

Moab Grid Suite is comprised of powerful applications which allow organizations to consolidate reporting, information gathering, and workload, resource, and data management. Moab Grid Suite delivers these services in a near-transparent way: users are not even aware they are using grid resources – they only know that they are getting their work done more easily and faster than ever before.

The Moab Grid Suite incorporates the following industry-leading applications from Adaptive Computing:

- || **Moab Workload Manager**
A policy-based workload management and scheduling engine
- || **Moab Grid Manager**
powerful graphical cluster administration interface, monitor, and reporting tool
- || **Moab Access Portal**
A web-based end-user job submission and management portal

BUSINESS BENEFITS

- || Move from cluster to optimized grid quickly with unified management across heterogeneous clusters
- || Intelligent scheduling that ensures jobs start and run as fast as possible by selecting optimal resources
- || Flexible policy and event engine that adjusts workload processing at both grid and cluster levels
- || Grid-wide interface and reporting tools to view grid resources, status/usage charts, and trends over time for capacity planning, diagnostic, and accounting purposes
- || Control/Allow different business groups to access and view grid resources, regardless of physical or organizational boundaries, or restrict access of resources to specific entities

SYSTEM COMPATIBILITY

- || Operating system support for Linux (all), UNIX (AIX, IRIX, HP-UX, FreeBSD, OSF/Tru-64, Solaris, etc.), Mac OSX & limited Windows support
- || Resource Manager support for LSF, TORQUE, PBSPro, SGE, SLURM, LoadLeveler, OpenPBS, BProc, and custom resource managers

KEY CAPABILITIES

Process More Work in Less Time to Maximize ROI

- || Achieve higher, more consistent utilization of resource capabilities and capacity with intelligent scheduling that matches job requests to best-suited resources
- || Use optimized data staging to ensure remote data availability is synchronized with resource availability to minimize resource blocking
- || Achieve better performance with automatic learning which optimizes scheduling decisions based on historical workload results
- || Allow local cluster level optimization of most grid workload
- || Enable broader resource access with automatic job translation which allows to effectively run on more destination clusters

Grid Control with Automated Tasks, Policies, Reporting

- || Guarantee that the most important work is prioritized first with flexible global grid policies that respect local cluster policies and support grid service level agreements
- || Ensure availability of key resources and specific times with advanced reservations
- || Tune policies prior to roll out with cluster and grid-level simulation
- || Use a global view of all grid operations for grid self-diagnostics, planning, reporting and grid-wide and per-cluster accounting across all resources, jobs and clusters
- || Create Virtual Private Clusters so users only need to see the resources and jobs they have access to without the complexity of seeing or wanting what is off limits

CLUSTER MANAGEMENT MADE EASY

MOAB GRID SUITE

FIGURE 8 CLUSTER SOVEREIGNTY AND TRUSTED SHARING

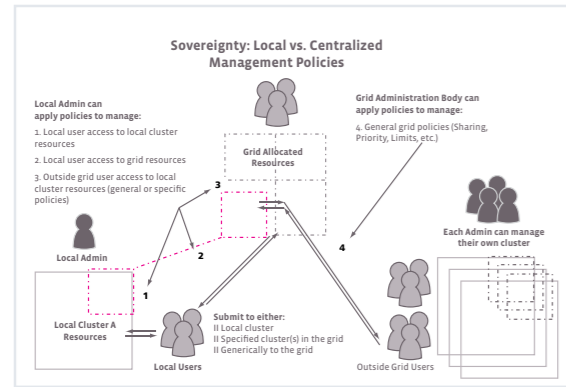


FIGURE 9 INCREASE USER PRODUCTIVITY



Controls for Cluster Sovereignty and Trusted Sharing

- II Guarantee that shared resources are allocated fairly with global policies that fully respect local cluster configuration and needs
- II Establish trust between resource owners through built-in usage controls, accounting and graphical reporting of usage across all shared resources
- II Maintain cluster sovereignty with granular controls over where jobs can originate and where jobs can be processed
- II Establish resource ownership and enforce priority access to resources when they want with owner-based prioritization, preemption, and access guarantees

Increase User Productivity

- II Reduce end user training and job management with easy-to-use graphical interfaces that automate data migration and leverage unified credential mapping
- II Enable end users to easily submit and manage their own jobs through a web browser to minimize the costs of managing a growing base of needy users
- II Collaborate more effectively with multi-cluster co-allocation of resources, allowing compute, network, and data resources to be reserved for key projects
- II Leverage saved job templates allowing users to quickly submit multiple jobs with minimal changes

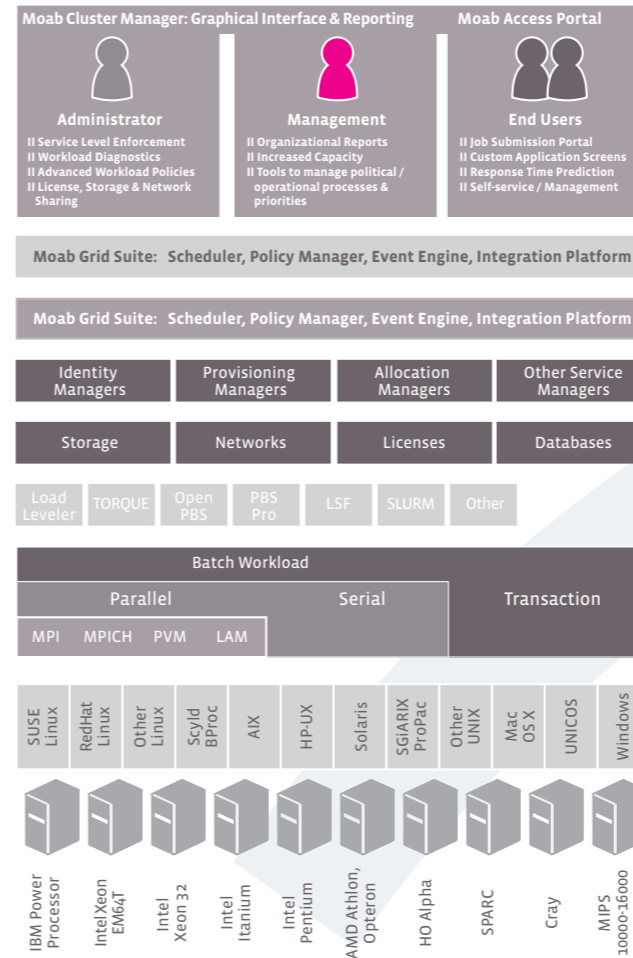
Adaptive Computing, Cluster Resources, Moab, Moab Viewpoint, Moab Workload Manager, Moab Cluster Manager, Moab Cluster Suite, Moab Grid Scheduler, Moab Grid Suite, Moab Access Portal, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo and the Cluster Resources logo are trademarks of Adaptive Computing Enterprises, Inc.



Unify Management Across Diverse Independent Clusters

- II Unify management across existing internal, external and partner clusters and their diverse resource managers, databases, operating systems and hardware
- II Get out-of-the-box local area grid and wide area grid support with scalability to more than 100 clusters and tens of thousands of diverse nodes

FIGURE 10 UNIFIED MANAGEMENT WITH MOAB GRID SUITE



- II Manage secure access to resources with simple credential mapping or seamless integration with Globus's advanced security toolset
- II Leverage existing data migration technologies such as SCP, GASS or GridFTP

CLUSTER MANAGEMENT MADE EASY

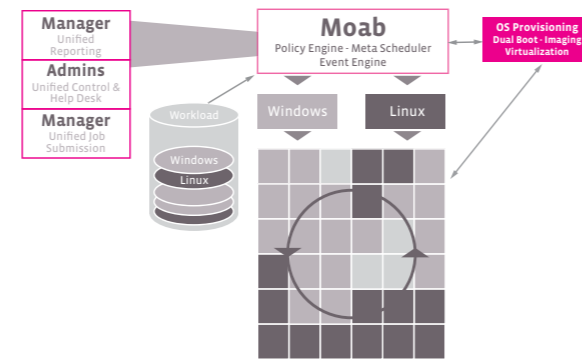
MOAB ADAPTIVE HPC SUITE

MOAB ADAPTIVE HPC SUITE

Moab Adaptive HPC Suite changes a node's operating system on the fly in response to workload needs.

As an intelligent metascheduler, Moab determines when the OS mix should be modified for optimal efficiency, based on defined policies and service-level agreements and on current and projected workload. When specified conditions are met, Moab automatically triggers the OS change using a site's preferred OS-modification technology, such as dual boot, diskful, or stateless (diskfree) provisioning.

FIGURE 11 DYNAMICAL DEPLOYMENT WITH MOAB ADAPTIVE HPC SUITE



Moab Adaptive HPC Suite enables maximum resource utilization and increased ROI for both new and existing clusters. The hybrid-OS cluster model also consolidates administration and centralizes job submission across OS platforms. Administrators can easily manage the policies and workload for multiple OS environments using Moab's unified console. Moab can also make the dual-OS nature of the cluster transparent to end users by applying application and workload information that ensures jobs run on the correct OS without the need for users to specify an OS. Moab can manipulate, grow, and shrink allocated resources to meet service-level targets.

“With Moab solutions, we can meet very demanding customers’ requirements as regards unified management of heterogeneous cluster environments, dual-boot Windows-Linux solutions, and flexible and powerful configuration or reporting options, while at the same time offering an easy-to-use and unified management interface. Our customer esteem that highly.”

Thomas Gebert HPC Solution Engineer

BENEFITS

- II Leverage your HPC infrastructure more efficiently as a single, unified resource free from restrictive and costly resource silos
- II Simplify the use and management of mixed-cluster environments with a unified interface and automated service-level enforcement
- II Balance resource usage between OS environments

and respond to workload surges and resource failures with dynamic adaptation of the OS mix

- II Schedule automatic reallocation based on workload requirements and established policies
- II Automate full business processes that include both Windows and Linux applications
- II Tune and reserve the future OS environment mix based on scheduled activities and historical usage patterns

Capability	Windows	Linux	Moab	Advantages
Adapt OS Mix to Meet Workload Needs			•	Adapt an OS environment to meet the workload needs and mission objectives of an organization. Handle adaptation of OS mix for resource failures, workload surges, service level guarantees, resource reservations, balancing purposes
Enable Multi-OS Business Processes Workflows			•	Enable more automated business processes through automating event-driven workflows that cross multiple OSes. Example: Process data on Linux and visualize the results on Windows as part of a Moab integrated workflow
Unify Job Submission			•	Submit jobs via a single web submission interface; Moab applies them to the appropriate OS environment
Unify Workload Management & Reporting			•	Manage workload on both OSes through a single cluster management & helpdesk tool & utilize unified reporting
HPC Resource Manager	•		•	Windows HPC Server 2008 R2 includes resource management; Adaptive Computing provides TORQUE Resource Manager for Linux
System Monitoring Tools	•	•	•	Both OSes and TORQUE provide valuable hardware monitoring
Message Passing	•	•		Both OSes include Message Passing Tools
Operating System	•	•		Windows and Linux; any OS can be used



INTEL CLUSTER READY

A QUALITY STANDARD FOR HPC CLUSTERS

Intel Cluster Ready is designed to create predictable expectations for users and providers of HPC clusters, primarily targeting customers in the commercial and industrial sectors. These are not experimental “test-bed” clusters used for computer science and computer engineering research, or high-end “capability” clusters closely targeting their specific computing requirements that power the high-energy physics at the national labs or other specialized research organizations.

Intel Cluster Ready seeks to advance HPC clusters used as computing resources in production environments by providing cluster owners with a high degree of confidence that the clusters they deploy will run the applications their scientific and engineering staff rely upon to do their jobs. It achieves this by providing cluster hardware, software, and system providers with a precisely defined basis for their products to meet their customers’ production cluster requirements.

INTEL CLUSTER READY

A QUALITY STANDARD FOR HPC CLUSTERS

WHAT ARE THE OBJECTIVES OF ICR?

The primary objective of Intel Cluster Ready is to make clusters easier to specify, easier to buy, easier to deploy, and make it easier to develop applications that run on them. A key feature of ICR is the concept of “application mobility”, which is defined as the ability of a registered Intel Cluster Ready application – more correctly, the same binary – to run correctly on any certified Intel Cluster Ready cluster. Clearly, application mobility is important for users, software providers, hardware providers, and system providers.

- II Users want to know the cluster they choose will reliably run the applications they rely on today, and will rely on tomorrow
- II Application providers want to satisfy the needs of their customers by providing applications that reliably run on their customers’ cluster hardware and cluster stacks
- II Cluster stack providers want to satisfy the needs of their customers by providing a cluster stack that supports their customers’ applications and cluster hardware
- II Hardware providers want to satisfy the needs of their customers by providing hardware components that supports their customers’ applications and cluster stacks
- II System providers want to satisfy the needs of their customers by providing complete cluster implementations that reliably run their customers’ applications

Without application mobility, each group above must either try to support all combinations, which they have neither the time nor resources to do, or pick the “winning combination(s)” that best supports their needs, and risk making the wrong choice.

“The Intel Cluster Checker allows us to certify that our transtec HPC clusters are compliant with an independent high quality standard. Our customers can rest assured: their applications run as they expect.”

Thomas Gebert HPC Solution Engineer

The Intel Cluster Ready definition of application portability supports all of these needs by going beyond pure portability, (re-compiling and linking a unique binary for each platform), to application binary mobility, (running the same binary on multiple platforms), by more precisely defining the target system.

A further aspect of application mobility is to ensure that registered Intel Cluster Ready applications do not need special programming or alternate binaries for different message fabrics. Intel Cluster Ready accomplishes this by providing an MPI implementation supporting multiple fabrics at runtime; through this, registered Intel Cluster Ready applications obey the “message layer independence property”. Stepping back, the unifying concept of Intel Cluster Ready is “one-to-many,” that is,

- II One application will run on many clusters
- II One cluster will run many applications

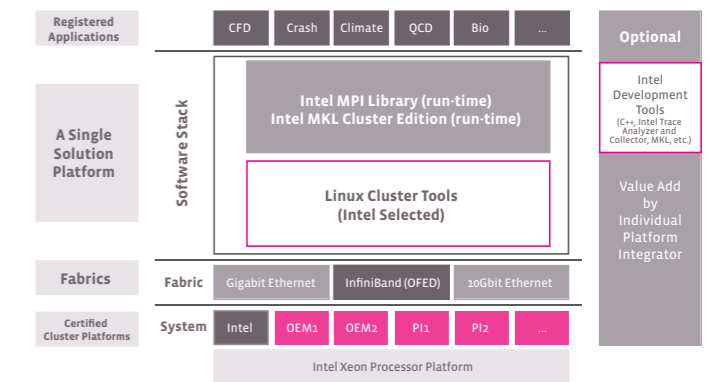
How is one-to-many accomplished? Looking at Figure 1, you see the abstract Intel Cluster Ready “stack” components that always exist in every cluster, i.e., one or more applications, a cluster software stack, one or more fabrics, and finally the underlying cluster hardware. The remainder of that picture (to the right) shows the components in greater detail.

Applications, on the top of the stack, rely upon the various APIs, utilities, and file system structure presented by the underlying software stack. Registered Intel Cluster Ready applications are always able to rely upon the APIs, utilities, and file system structure specified by the Intel Cluster Ready Specification; if

an application requires software outside this “required” set, then Intel Cluster Ready requires the application to provide that software as a part of its installation. To ensure that this additional per-application software doesn’t conflict with the cluster stack or other applications, Intel Cluster Ready also requires the additional software to be installed in application-private trees, so the application knows how to find that software while not interfering with other applications. While this may well cause duplicate software to be installed, the reliability provided by the duplication far outweighs the cost of the duplicated files. A prime example supporting this comparison is the removal of a common file (library, utility, or other) that is unknowingly needed by some other application – such errors can be insidious to repair even when they cause an outright application failure.

Cluster platforms, at the bottom of the stack, provide the APIs, utilities, and file system structure relied upon by registered applications. Certified Intel Cluster Ready platforms ensure

FIGURE 1 ICR STACK



the APIs, utilities, and file system structure are complete per the Intel Cluster Ready Specification; certified clusters are able to provide them by various means as they deem appropriate. Because of the clearly defined responsibilities ensuring the presence of all software required by registered applications, system providers have a high confidence that the certified clusters they build are able to run any certified applications their customers rely on. In addition to meeting the Intel Cluster Ready requirements, certified clusters can also provide their added value, that is, other features and capabilities that increase the value of their products.

HOW DOES INTEL CLUSTER READY ACCOMPLISH ITS OBJECTIVES?

At its heart, Intel Cluster Ready is a definition of the cluster as a parallel application platform, as well as a tool to certify an actual cluster to the definition. Let's look at each of these in more detail, to understand their motivations and benefits.

A definition of the cluster as parallel application platform

The Intel Cluster Ready Specification is very much written as the requirements for, not the implementation of, a platform upon which parallel applications, more specifically MPI applications, can be built and run. As such, the specification doesn't care whether the cluster is diskful or diskless, fully distributed or single system image (SSI), built from "Enterprise" distributions or community distributions, fully open source or not. Perhaps more importantly, with one exception, the specification doesn't have any requirements on how the cluster is built; that one exception is that compute nodes must be built with automated tools, so that new, repaired, or replaced nodes can be rebuilt identically to the existing nodes without any manual interaction, other than possibly initiating the build process.

Some items the specification does care about include:

- || the ability to run both 32- and 64-bit applications, including MPI applications and X-clients, on any of the compute nodes
- || consistency among the compute nodes' configuration, capability, and performance
- || the identical accessibility of libraries and tools across the cluster
- || the identical access by each compute node to permanent and temporary storage, as well as users' data
- || the identical access to each compute node from the head node
- || the MPI implementation provides fabric independence
- || all nodes support network booting and provide a remotely accessible console

The specification also requires that the runtimes for specific Intel software products are installed on every certified cluster:

- || Intel Math Kernel Library
- || Intel MPI Library Runtime Environment
- || Intel Threading Building Blocks

This requirement does two things. First and foremost, main-line Linux distributions do not necessarily provide a sufficient software stack to build an HPC cluster – such specialization is beyond their mission. Secondly, the requirement ensures that programs built with this software will always work on certified clusters and enjoy simpler installations. As these runtimes are directly available from the web, the requirement does not cause additional costs to certified clusters. It is also very important to note that this does not require certified applications to use these libraries nor does it preclude alternate libraries, e.g., other MPI implementations, from being present on certified clusters. Quite clearly, an

application that requires, e.g., an alternate MPI, must also provide the runtimes for that MPI as a part of its installation.

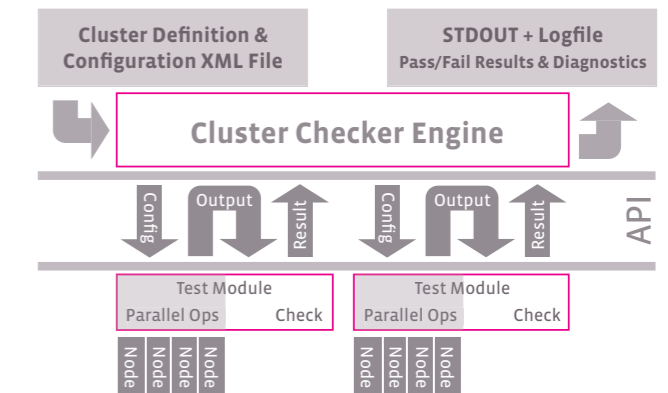
A tool to certify an actual cluster to the definition

The Intel Cluster Checker, included with every certified Intel Cluster Ready implementation, is used in four modes in the life of a cluster:

- || to certify a system provider's prototype cluster as a valid implementation of the specification
- || to verify to the owner that the just-delivered cluster is a "true copy" of the certified prototype
- || to ensure the cluster remains fully functional, reducing service calls not related to the applications or the hardware
- || to help software and system providers diagnose and correct actual problems to their code or their hardware.

While these are critical capabilities, in all fairness, this greatly understates the capabilities of Intel Cluster Checker. The tool will not only verify the cluster is performing as expected. To do this, per-node and cluster-wide static and dynamic tests are made of the hardware and software.

FIGURE 2 INTEL CLUSTER CHECKER



INTEL CLUSTER READY

INTEL CLUSTER READY BUILDS HPC MOMENTUM

The static checks ensure the systems are configured consistently and appropriately. As one example, the tool will ensure the systems are all running the same BIOS versions as well as having identical configurations among key BIOS settings. This type of problem – differing BIOS versions or settings – can be the root cause of subtle problems such as differing memory configurations that manifest themselves as differing memory bandwidths only to be seen at the application level as slower than expected overall performance. As is well-known, parallel program performance can be very much governed by the performance of the slowest components, not the fastest. In another static check, the Intel Cluster Checker will ensure that the expected tools, libraries, and files are present on each node, identically located on all nodes, as well as identically implemented on all nodes. This ensures that each node has the minimal software stack specified by the specification, as well as identical software stack among the compute nodes.

A typical dynamic check ensures consistent system performance, e.g., via the STREAM benchmark. This particular test ensures processor and memory performance is consistent across compute nodes, which, like the BIOS setting example above, can be the root cause of overall slower application performance. An additional check with STREAM can be made if the user configures an expectation of benchmark performance; this check will ensure that performance is not only consistent across the cluster, but also meets expectations. Going beyond processor performance, the Intel MPI Benchmarks are used to ensure the network fabric(s) are performing properly and, with a configuration that describes expected performance levels, up to the cluster



INTEL CLUSTER READY BUILDS HPC MOMENTUM

With the Intel Cluster Ready (ICR) program, Intel Corporation set out to create a win-win scenario for the major constituencies in the high-performance computing (HPC) cluster market. Hardware vendors and independent software vendors (ISVs) stand to win by being able to ensure both buyers and users that their products will work well together straight out of the box. System administrators stand to win by being able to meet corporate demands to push HPC competitive advantages deeper into their organizations while satisfying end users' demands for reliable HPC cycles, all without increasing IT staff. End users stand to win by being able to get their work done faster, with less downtime, on certified cluster platforms. Last but not least, with ICR, Intel has positioned itself to win by expanding the total addressable market (TAM) and reducing time to market for the company's microprocessors, chip sets, and platforms.

The Worst of Times

For a number of years, clusters were largely confined to government and academic sites, where contingents of graduate students and midlevel employees were available

to help program and maintain the unwieldy early systems. Commercial firms lacked this low-cost labor supply and mistrusted the favored cluster operating system, open source Linux, on the grounds that no single party could be held accountable if something went wrong with it. Today, cluster penetration in the HPC market is deep and wide, extending from systems with a handful of processors to some of the world's largest supercomputers, and from under \$25,000 to tens or hundreds of millions of dollars in price. Clusters increasingly pervade every HPC vertical market: biosciences, computer-aided engineering, chemical engineering, digital content creation, economic/financial services, electronic design automation, geosciences/geo-engineering, mechanical design, defense, government labs, academia, and weather/climate.

But IDC studies have consistently shown that clusters remain difficult to specify, deploy, and manage, especially for new and less experienced HPC users. This should come as no surprise, given that a cluster is a set of independent computers linked together by software and networking technologies from multiple vendors.

Clusters originated as do-it-yourself HPC systems. In the late 1990s users began employing inexpensive hardware to cobble together scientific computing systems based on the "Beowulf cluster" concept first developed by Thomas Sterling and Donald Becker at NASA. From their Beowulf origins, clusters have evolved and matured substantially, but the system management issues that plagued their early years remain in force today.

INTEL CLUSTER READY

INTEL CLUSTER READY BUILDS HPC MOMENTUM

The Need for Standard Cluster Solutions

The escalating complexity of HPC clusters poses a dilemma for many large IT departments that cannot afford to scale up their HPC-knowledgeable staff to meet the fast-growing end-user demand for technical computing resources. Cluster management is even more problematic for smaller organizations and business units that often have no dedicated, HPC-knowledgeable staff to begin with.

The ICR program aims to address burgeoning cluster complexity by making available a standard solution (aka reference architecture) for Intel-based systems that hardware vendors can use to certify their configurations and that ISVs and other software vendors can use to test and register their applications, system software, and HPC management software. The chief goal of this voluntary compliance program is to ensure fundamental hardware-software integration and interoperability so that system administrators and end users can confidently purchase and deploy HPC clusters, and get their work done, even in cases where no HPC-knowledgeable staff are available to help.

The ICR program wants to prevent end users from having to become, in effect, their own systems integrators. In smaller organizations, the ICR program is designed to allow overworked IT departments with limited or no HPC expertise to support HPC user requirements more readily. For larger organizations with dedicated HPC staff, ICR creates confidence that required user applications will work, eases the problem of system administration, and allows HPC cluster

systems to be scaled up in size without scaling support staff. ICR can help drive HPC cluster resources deeper into larger organizations and free up IT staff to focus on mainstream enterprise applications (e.g., payroll, sales, HR, and CRM).

The program is a three-way collaboration among hardware vendors, software vendors, and Intel. In this triple alliance, Intel provides the specification for the cluster architecture implementation, and then vendors certify the hardware configurations and register software applications as compliant with the specification. The ICR program's promise to system administrators and end users is that registered applications will run out of the box on certified hardware configurations.

ICR solutions are compliant with the standard platform architecture, which starts with 64-bit Intel Xeon processors in an Intel-certified cluster hardware platform. Layered on top of this foundation are the interconnect fabric (Gigabit Ethernet, InfiniBand) and the software stack: Intel-selected Linux cluster tools, an Intel MPI runtime library, and the Intel Math Kernel Library. Intel runtime components are available and verified as part of the certification (e.g., Intel tool runtimes) but are not required to be used by applications. The inclusion of these Intel runtime components does not exclude any other components a systems vendor or ISV might want to use. At the top of the stack are Intel-registered ISV applications.

At the heart of the program is the Intel Cluster Checker, a validation tool that verifies that a cluster is specification compliant and operational before ISV applications are

ever loaded. After the cluster is up and running, the Cluster Checker can function as a fault isolation tool in wellness mode. Certification needs to happen only once for each distinct hardware platform, while verification – which determines whether a valid copy of the specification is operating – can be performed by the Cluster Checker at any time.

Cluster Checker is an evolving tool that is designed to accept new test modules. It is a productized tool that ICR members ship with their systems. Cluster Checker originally was designed for homogeneous clusters but can now also be applied to clusters with specialized nodes, such as all-storage sub-clusters. Cluster Checker can isolate a wide range of problems, including network or communication problems.



INTEL CLUSTER READY

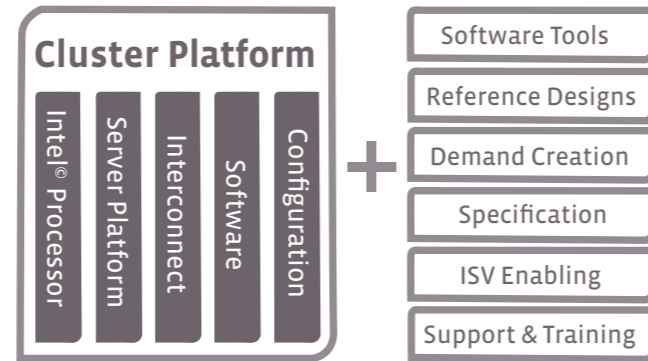
THE TRANSTEC BENCHMARKING CENTER

Intel, XEON and certain other trademarks and logos appearing in this brochure, are trademarks or registered trademarks of Intel Corporation.



provider's performance expectations. Network inconsistencies due to poorly performing Ethernet NICs, InfiniBand HBAs, faulty switches, and loose or faulty cables can be identified. Finally, the Intel Cluster Checker is extensible, enabling additional tests to be added supporting additional features and capabilities. This enables the Intel Cluster Checker to not only support the minimal requirements of the Intel Cluster Ready Specification, but the full cluster as delivered to the customer.

FIGURE 3 INTEL CLUSTER READY PROGRAM



Conforming hardware and software

The preceding was primarily related to the builders of certified clusters and the developers of registered applications. For end users that want to purchase a certified cluster to run registered applications, the ability to identify registered applications and certified clusters is most important, as that will reduce their effort to evaluate, acquire, and deploy the clusters that run their applications, and then keep that computing resource operating properly, with full performance, directly increasing their productivity.



transtec offers their customers a new and fascinating way to evaluate transtec's HPC solutions in real world scenarios. With the transtec Benchmarking Center solutions can be explored in detail with the actual applications the customers will later run on them. Intel Cluster Ready makes this feasible by simplifying the maintenance of the systems and set-up of clean systems very easily, and as often as needed. As High-Performance Computing (HPC) systems are utilized for numerical simulations, more and more advanced clustering technologies are being deployed. Because of its performance, price/performance and energy efficiency advantages, clusters now dominate all segments of the HPC market and continue to gain acceptance. HPC computer systems have become far more widespread and pervasive in government, industry, and academia. However, rarely does the client have the possibility to test their actual application on the system they are planning to acquire.

THE TRANSTEC BENCHMARKING CENTER

transtec HPC solutions get used by a wide variety of clients. Among those are most of the large users of compute power at German and other European universities and research centers as well as governmental users like the German army's compute center and clients from the high tech, the automotive and other sectors. transtec HPC solutions have demonstrated their value in more than 400 installations. Most of transtec's cluster systems are based on SUSE Linux Enterprise Server, Red Hat Enterprise Linux, CentOS, or Scientific Linux. With xCAT for efficient cluster deployment, and Moab Cluster Suite by Adaptive

Computing for high-level cluster management, transtec is able to efficiently deploy and ship easy-to-use HPC cluster solutions with enterprise-class management features. Moab has proven to provide easy-to-use cluster and job management for small systems as well as the largest cluster installations worldwide. However, when selling clusters to governmental customers as well as other large enterprises, it is often required that the client can choose from a range of competing offers. Many times there is a fixed budget available and competing solutions are compared based on their performance towards certain custom benchmark codes.

So, in 2007 transtec decided to add another layer to their already wide array of competence in HPC – ranging from cluster deployment and management, the latest CPU, board and network technology to HPC storage systems. In transtec's HPC Competence Center the systems are being assembled. transtec is using Intel Cluster Ready to facilitate testing, verification, documentation, and final testing throughout the actual build process. At the benchmarking center transtec can now offer a set of small clusters with the "newest and hottest technology" through Intel Cluster Ready. A standard installation infrastructure gives transtec a quick and easy way to set systems up according to their customers' choice of operating system, compilers, workload management suite, and so on. With Intel Cluster Ready there are prepared standard set-ups available with verified performance at standard benchmarks while the system stability is guaranteed by our own test suite and the Intel Cluster Checker.

The Intel Cluster Ready program is designed to provide a common standard for HPC clusters, helping organizations design and build seamless, compatible and consistent cluster configurations. Integrating the standards and tools provided by this program can help significantly simplify the deployment and management of HPC clusters.



WINDOWS HPC SERVER 2008 R2 MICROSOFT HPC SOLUTION

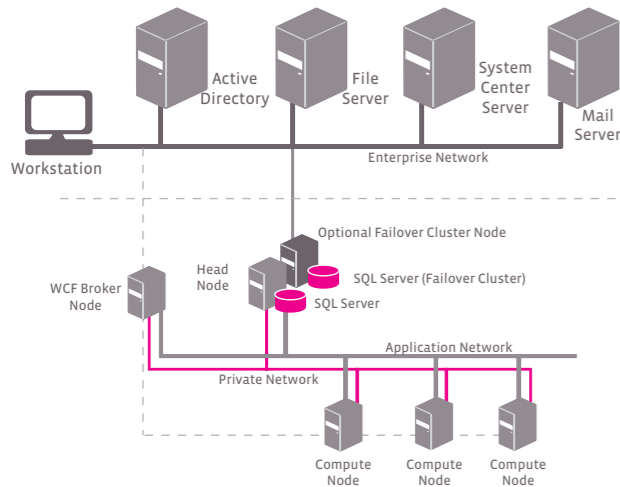
Windows HPC Server 2008 R2 is the third version of the Microsoft solution for high performance computing (HPC). Built on Windows Server 2008 R2 64-bit technology, Windows HPC Server 2008 R2 efficiently scales to thousands of nodes and integrates seamlessly with Windows-based IT infrastructures, providing a powerful combination of ease-of-use, low ownership costs, and performance.

Compared to the previous version, Windows HPC Server 2008 R2 delivers significant improvements in several areas. Through these enhancements, Windows HPC Server 2008 R2 makes it easier than ever for companies to benefit from high-performance computing. System administrators can more easily deploy and manage powerful HPC solutions, developers can more easily build applications, and end users can more easily access those solutions from their Windows-based desktops.

WINDOWS HPC SERVER 2008 R2

ELEMENTS OF THE MICROSOFT HPC SOLUTION

FIGURE 1 WINDOWS HPC CLUSTER SETUP



ELEMENTS OF THE MICROSOFT HPC SOLUTION

Windows HPC Server 2008 R2 combines the underlying stability and security of Windows Server 2008 R2 with the features of Microsoft HPC Pack 2008 R2 to provide a robust, scalable, cost-effective, and easy-to-use HPC solution. A basic Windows HPC Server 2008 R2 solution is composed of a cluster of servers, with a single head node (or a primary and backup head node in a highly available configuration) and one or more compute nodes (see Figure 1). The head node controls and mediates all access to the cluster resources and is the single point of management, deployment, and job scheduling for the cluster. Windows HPC Server 2008 R2 can integrate with an existing Active Directory directory service infrastructure for security and account management, and can use Microsoft System Center Operations Manager for data center monitoring. Windows HPC Server 2008 R2 uses Microsoft SQL Server 2008 as a data repository for the head node. Windows HPC Server 2008 R2 can take advantage of the failover clustering capabilities provided in Windows Server 2008 R2 Enterprise and some editions of Microsoft SQL Server to provide high-availability failover clustering for the head node. With clustering, in the event of a head node failure, the Job Scheduler will automatically – or manually, if desired – fail over to a second server. Job Scheduler clients see no change in the head node during the failover and fail-back processes, helping to ensure uninterrupted cluster operation. Windows HPC Server 2008 R2 adds support for a remote head node database, enabling organizations to take advantage of an existing enterprise database.

Feature	Implementation	Benefits
Operating system	Windows Server 2008 and/or Windows Server 2008 R2 (Head node is R2 only, compute nodes can be both)	Inherits security and stability features from Windows Server 2008 and Windows Server 2008 R2.
Processor type	x64 (AMD64 or Intel EM64T)	Large memory model and processor efficiencies of x64 architecture.
Node deployment	Windows Deployment Services	Image-based deployment, with full support for multicasting and diskless boot.
Head node redundancy	Windows Failover Clustering and SQL Server Failover Clustering	Provides a fully redundant head node and scheduler (requires Windows Server 2008 R2 Enterprise and SQL Server Standard Edition).
Management	Integrated Administration Console	Provides a single user interface for all aspects of node and job management, grouping, monitoring, diagnostics, and reporting.
Network topology	Network Configuration Wizard	Fully automated Network Configuration Wizard for configuring the desired network topology.
Application network	MS-MPI	High-speed application network stack using NetworkDirect. Shared memory implementation for multicore processors. Highly compatible with existing MPICH2 implementations.
Scheduler	Job Manager Console	GUI is integrated into the Administration Console or can be used standalone. Command line interface supports Windows PowerShell scripting and legacy command-line scripts from Windows Compute Cluster Server. Greatly improved speed and scalability. Support for SOA applications.
Monitoring	Integrated into Administration Console	New heat map provides at-a-glance view of cluster performance and status for up to 1,000 nodes.
Reporting	Integrated into Administration Console	Standard, prebuilt reports and historical performance charts. Additional reports can be created using SQL Server Analysis Services.
Diagnostics	Integrated into Administration Console	Out-of-the-box verification and performance tests, with the ability to store, filter, and view test results and history. An extensible diagnostic framework for creating custom diagnostics and reports.
Parallel runtime	Enterprise-ready SOA infrastructure	Windows HPC Server 2008 R2 provides enhanced support for SOA workloads, helping organizations more easily build interactive HPC applications, make them more resilient to failure, and more easily manage those applications.

WINDOWS HPC SERVER 2008 R2 DEPLOYMENT, SYSTEM MANAGEMENT, AND MONITORING

DEPLOYMENT

One challenge to the adoption of HPC solutions lies in the deployment of large clusters. With a design goal of supporting the deployment of 1,000 nodes in less than an hour, Windows HPC Server 2008 R2 builds on the capabilities provided by the Windows Deployment Services transport to simplify and streamline the deployment and updating of cluster nodes, using Windows Imaging Format (WIM) files and multiband multicast to rapidly deploy compute nodes in parallel. Graphical deployment tools are integrated into the Administration Console, including Node Templates for easily defining the configuration of compute nodes. New features in Windows HPC Server 2008 R2 – such as support for Windows Server 2008-based, mixed version clusters, and diskless boot – provide additional flexibility, enabling organizations to easily deploy solutions that are optimized to meet their needs.

Node Templates in Windows HPC Server 2008 R2 provide an easy way to define the desired configuration of compute nodes, with each Node Template including the base operating system image, drivers, configuration parameters, and, if desired, additional software. A Node Template Generation Wizard guides the administrator through the process of creating Node Templates, including support for injecting drivers into images. An improved Template Editor provides advanced configuration capabilities, including configuring Node Templates for automatic application deployment. Windows HPC Server 2008 R2 supports the deployment of compute nodes and broker nodes based on Windows Server 2008 or Windows Server 2008 R2, including mixed-version clusters.

“The performance of transtec HPC systems combined with the usability of Windows HPC Server 2008 R2 provides our customers with HPC solutions that are unrivalled in power as well as ease of management.”

Dr. Oliver Tennert Director Marketing & HPC Solutions

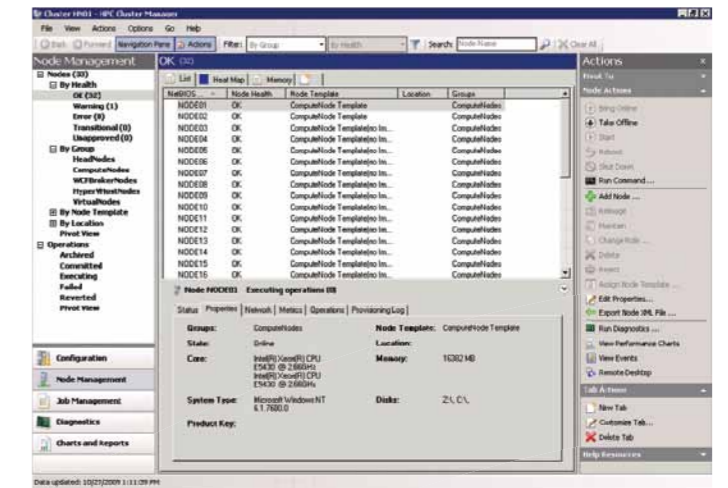
Diskless booting of compute nodes, a new feature in Windows HPC Server 2008 R2, is provided through support for iSCSI boot from a storage array. This mechanism uses DHCP reservations for mapping to disk and leverages the storage vendor’s mechanism for creating differencing disks for compute nodes. The Administration Console includes diagnostic tests that can be used post-deployment to detect common problems, monitor node loading, and view job status across the cluster. In addition, the new “Lizard” (LINPACK Wizard) in Windows HPC Server 2008 R2 enables administrators to heavily load the cluster – thereby providing an efficient mechanism for detecting issues related to configuration and deployment, networking, power, cooling, and so on.

SYSTEM MANAGEMENT

Another major challenge that organizations can face is the management and administration of HPC clusters. This has traditionally been a departmental or organizational-level challenge, requiring one or more dedicated IT professionals to manage and deploy nodes. At the same time, users submitting batch jobs are competing for limited HPC resources. Windows HPC Server 2008 R2 is designed to facilitate ease-of-management. It provides a graphical Administration Console that puts all the tools required for system management at an administrator’s fingertips, including the ability to easily drill down on node details such as metrics, logs, and configuration status. Support for the Windows PowerShell scripting language facilitates the automation of system administration tasks. An enhanced “heat map” view provides system administrators with an “at a glance” view of the cluster status, including the ability to define tabs with different views of system health and resource usage. Other new

management-related features in Windows HPC Server 2008 R2 include additional criteria for filtering views, support for location-based node grouping, a richer reporting database for building custom reports, and an extensible diagnostic framework.

FIGURE 2 THE ADMINISTRATION CONSOLE



MONITORING, REPORTING, AND DIAGNOSTICS

The Node Management pane within the Administration Console is used to monitor node status and initiate node-specific actions. New node management-related features in Windows HPC Server 2008 R2 include an enhanced heat map with overlay view, additional filtering criteria, customizable tabs, and location-based node grouping. In Windows HPC Server 2008 R2, the heat map has been enhanced to provide an at-a-glance view of system health and performance for clusters upwards of 1,000 nodes. System administrators can define and prioritize up to three metrics (as

WINDOWS HPC SERVER 2008 R2

JOB SCHEDULING

FIGURE 3 THE HEAT MAP VIEW GIVES INSTANT FEEDBACK ON THE HEALTH OF THE CLUSTER

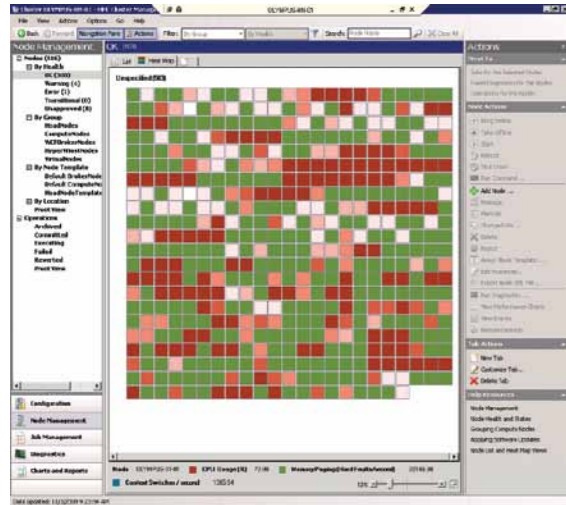
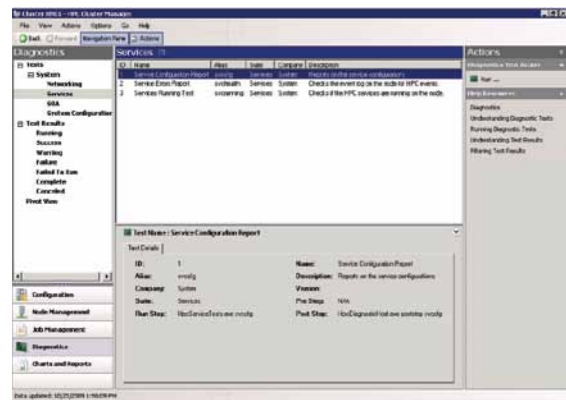


FIGURE 4 DIAGNOSTIC PANE



well as minimum and maximum thresholds for each metric) to build customized views of cluster health and status.

Windows HPC Server 2008 R2 provides a set of prebuilt reports and charts to help system administrators understand system status, usage, and performance. Accessed through the Reports and Charts tab on the Administrator Console, these prebuilt reports span four main categories: node availability, job resource usage, job throughput, and job turnaround. Windows HPC Server 2008 R2 also provides a set of prebuilt diagnostic reports to help system administrators verify that their clusters are working properly, along with a systematic way of running the tests and storing and viewing results. This significantly improves an administrator's experience in verifying deployment, troubleshooting failures, and detecting performance degradation. Cluster administrators can view a list of these diagnostic tests, run them, change diagnostic parameters at runtime, and view the results using the Diagnostics tab in the Administration Console or by using Windows PowerShell commands.

JOB SCHEDULING

The Job Scheduler queues jobs and their associated tasks, allocates resources to the jobs, initiates the tasks on the compute nodes, and monitors the status of jobs and tasks. In Windows HPC Server 2008 R2, the Job Scheduler has been enhanced to support larger clusters, more jobs, and larger jobs – including improved scheduling and task throughput at scale. It includes new policies for greater flexibility and resource utilization, and is built to address both traditional batch jobs as well as newer service-oriented applications.

The Job Scheduler supports both command-line and graphical interfaces. The graphical interface is provided through the Job Scheduling tab of the Administration Console or through the HPC Job Manager, a graphical interface for use by end-users submitting and managing jobs. Other supported interfaces include:

- Command line (cmd.exe)
- Windows PowerShell 2.0
- COM and .NET application programming interfaces to support a variety of languages, including VBScript, Perl, Fortran, C/C++, C#, and Java.
- The Open Grid Forum's HPC Basic Profile Web Services Interface, which supports job submission and monitoring from many platforms and languages.

The Windows HPC Server 2008 R2 interfaces are fully backwards compatible, allowing job submission and management from Microsoft Compute Cluster Server and Windows HPC Server 2008 interfaces.

Windows HPC Server 2008 R2 also provides a new user interface for showing job progress, and an enhanced API that enables developers to report more detailed job progress status to their HPC applications.

Scheduling policies determine how resources are allocated to jobs. Windows HPC Server 2008 R2 provides the ability to switch between traditional first-come, first-serve scheduling and a new service-balanced scheduling policy designed for SOA/dynamic (grid) workloads, with support for preemption, heterogeneous matchmaking (targeting of jobs to specific types

of nodes), growing and shrinking of jobs, backfill, exclusive scheduling, and task dependencies for creating workflows.

Windows HPC Server 2008 R2 also introduces prep and release tasks – tasks that are run before and after a job. Prep tasks are guaranteed to run once on each node in a job before any other tasks, as may be required to support setup or validation of node before the job is run. Release tasks are guaranteed to run once on each node in a job after all other tasks, as may be required to clean up or transfer files after the job.



SERVICE-ORIENTED ARCHITECTURE

With the number and size of problems being tackled on ever-larger clusters continuing to grow, organizations face increased challenges in developing HPC applications. Not only must these applications be built quickly, but they must run efficiently and be managed in a way that optimizes application performance, reliability, and resource utilization.

One approach to meeting these challenges is a service-oriented architecture (SOA) – an approach to building distributed, loosely coupled applications in which functions are separated into distinct services that can be distributed over a network, combined, and reused. Windows HPC Server 2008 R2 provides enhanced support for SOA workloads, helping organizations more easily build interactive HPC applications, make them more resilient to failure, and more easily manage those applications – capabilities that open the door to new application scenarios in areas such as financial trading and risk management.

When SOA Can Be Useful – and How it Works on a Cluster

HPC applications submitted to compute clusters are typically classified as either message intensive or embarrassingly parallel. While message-intensive applications comprise sequential

tasks, embarrassingly parallel problems can be easily divided into large numbers of parallel tasks, with no dependency or communication between them. To solve these embarrassingly parallel problems without having to write low-level code, developers need to encapsulate core calculations as a software modules. An SOA approach to development makes this encapsulation not only possible but easy, effectively hiding the details of data serialization and distributed computing.

With Windows HPC Server 2008 R2, tasks can run interactively as SOA applications. For interactive SOA applications, in addition to a head node and one or more compute nodes, the cluster also includes one or more Windows Communication Foundation broker nodes. The broker nodes act as intermediaries between the client application and the Windows Communication Foundation hosts running on compute nodes, load-balancing the client application's requests and returning the results to it.

Building SOA-Based HPC Applications

One attractive aspect of SOA applications is the ability to develop them quickly, without having to write a lot of low-level code. To achieve this, developers need to be able to easily encapsulate core calculations as software modules that can be deployed and run on the cluster. These software modules identify and marshal the data required for each calculation and optimize performance by minimizing the data movement and communication overhead.

Microsoft Visual Studio provides easy-to-use Windows Communication Foundation service templates and service referencing utilities to help software developers quickly prototype, debug, and unit-test SOA applications, with Win-

dows Communication Foundation effectively hiding the complexity of data serialization and distributed computing.

- **Fire-and Recollect Programming Model:**
A fire-and-recollect programming model – sometimes called fire-and-forget – is a common approach to building long-running SOA applications. The SOA runtime in Windows HPC Server 2008 R2 adds support for fire-and-recollect programming, enabling developers to implement reattachable sessions by decoupling requests and responses.
- **Durable Sessions:**
Another new feature in the Windows HPC Server 2008 R2 is the ability to implement durable sessions, where the SOA runtime persists requests and their corresponding responses on behalf of the client.
- **Finalization Hooks:**
The SOA runtime in Windows HPC Server 2008 R2 also adds support for finalization hooks, enabling developers to add logic to perform cleanup before a service exits.
- **Improved Java Interoperability:**
With Java sample code provided in the Windows HPC Server 2008 R2 Software Development Kit (SDK), developers can more easily write Java-based client applications that communicate with .NET services – and enjoy the same level of functionality provided with clients based on the .NET Framework and Windows Communication Foundation.

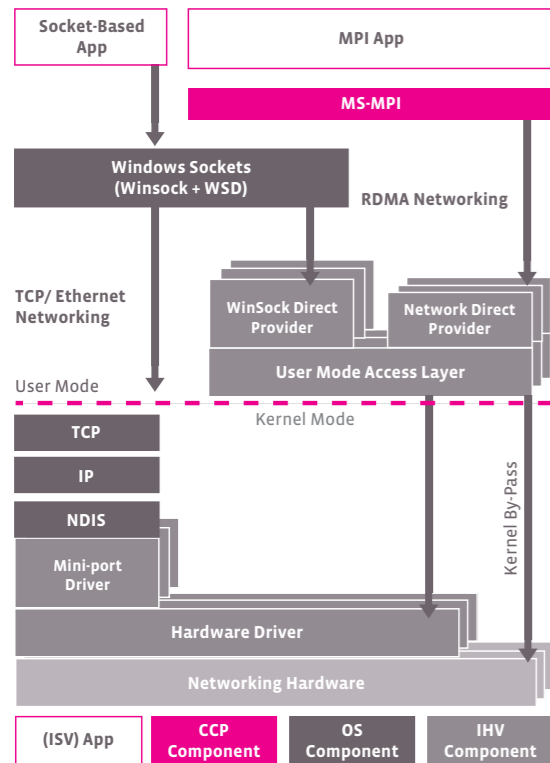
Running SOA-Based HPC Applications

In addition to developing SOA applications quickly, organizations must be able to run those applications efficiently, securely, and reliably. The SOA runtime in Windows HPC Server 2008 R2 helps organizations meet those needs through features such as low-latency round-trips for efficiently dis-

WINDOWS HPC SERVER 2008 R2

NETWORKING AND MPI

FIGURE 5 NETWORKDIRECT ARCHITECTURE



tributing short calculation requests, end-to-end Kerberos authentication with Windows Communication Foundation transport-level security, and dynamic allocation of resources to service instances. Windows HPC Server 2008 R2 also provides several new features to help organizations more reliably run their SOA applications, including support for broker restart/failover and message persistence.

II Message Resilience:

In the case of a temporary broker node failure or a catastrophic failure of the cluster, the SOA broker nodes will persist calculation requests and results. The session can continue without lost requests or results after the cluster recovers and the broker nodes are restarted.

II High-Availability Broker Nodes (Broker Restart/Failover):

Furthermore, the SOA runtime in Windows HPC Server 2008 R2 adds support for automated broker failover, enabling organizations to preserve computation results in the event of a failure – an essential requirement for nonstop processing of mission-critical applications. Configured using Microsoft Message Queuing (MSMQ) on remote storage and failover broker nodes, the cluster will migrate active sessions on failed broker nodes to healthy ones, thereby enabling nonstop processing.

NETWORKING AND MPI

Windows HPC Server 2008 R2 uses the Microsoft Message Passing Interface (MS-MPI), a portable, flexible, interconnect-independent API for messaging within and between HPC nodes. MS-MPI is based on the Argonne National Laboratory open-source MPICH2 implementation, and is compatible with the MPI2 standard.

MS-MPI can run over Gigabit Ethernet, 10 Gigabit Ethernet, and high-performance networking hardware such as InfiniBand, iWARP Ethernet, and Myrinet – or any other type of interconnect that provides a Winsock Direct, NetworkDirect, or TCP/IP interface. MS-MPI includes application support (bindings) for the C, Fortran77, and Fortran90 programming languages. With Windows HPC Server 2008 R2, organizations also can take advantage of new interconnect options, such as support for RDMA over Ethernet (iWARP) from Intel and new RDMA over Infiniband QDR (40 Gbps) hardware.

MS-MPI is optimized for shared memory communication to benefit the multicore systems prevalent in today's HPC clusters. MS-MPI in Windows HPC Server 2008 R2 introduces optimization of shared memory implementations for new Intel "Nehalem"-based processors, with internal testing by Microsoft showing up to a 20 to 30 percent performance improvement on typical commercial HPC.

NetworkDirect

MS-MPI can take advantage of NetworkDirect – a remote direct memory access (RDMA)-based interface – for superior networking performance and CPU efficiency. As shown in Figure 5, NetworkDirect uses a more direct path from MPI applications to networking hardware, resulting in very fast and efficient networking. Speeds and latencies are similar to those of custom, hardware-native interfaces from hardware providers.

Easier Troubleshooting of MPI Applications

MS-MPI integrates with Event Tracing for Windows to facilitate performance-tuning, providing a time-synchronized log for debugging MPI system and application events across multiple computers running in parallel. In addition, Microsoft Visual Studio 2008 includes a MPI Cluster Debugger that works with MS-MPI. Developers can launch their MPI applications on multiple compute nodes from within the Visual Studio environment, and Visual Studio will automatically connect the processes on each node, enabling developers individually pause and examine program variables on each node.

Tuning Wizard for LINPACK ("Lizard")

A new feature for Windows HPC Server 2008 R2, the Tuning Wizard for LINPACK ("Lizard") is a pushbutton, standalone executable that enables administrators to easily measure computational performance and efficiency for an HPC cluster. Furthermore,



MICROSOFT OFFICE EXCEL SUPPORT

Microsoft Office Excel is a critical business application across a broad range of industries. With its wealth of statistical analysis functions, support for constructing complex analyses, and virtually unlimited extensibility, Excel is clearly a tool of choice for analyzing business data. However, as calculations and modeling performed in Excel become more and more complex, Excel workbooks can take longer and longer to calculate, thereby reducing the business value provided.

Windows HPC Server 2008 R2 enables organizations to take advantage of HPC clusters to reduce calculation times for Excel workbooks by one or more orders of magnitude, scaling close to linearly as nodes or cores are added. Faster calculation times give business users and decision makers more information in less time, enabling more thorough analysis, faster access to important information, and better informed decisions. In addition, running Excel workbooks on an HPC cluster provides unique benefits in terms of reliability, resource utilization, and accounting and auditing support.

SPEEDING UP EXCEL WORKBOOKS

Windows HPC Server 2008 R2 supports three different approaches to calculating Excel workbooks on an HPC cluster: using Excel as a cluster SOA client, running Excel user defined functions (UDFs) on a cluster, and running Excel workbooks on a cluster. Using Excel as a cluster SOA client was possible with earlier versions of Windows HPC Server. Running Excel UDFs and Excel workbooks on a cluster are new capabilities, both of which require a combination of Windows HPC Server 2008 R2 and Office Excel 2010.

Using Excel as a Cluster SOA Client

Visual Studio Tools for Office provides a programming environment that is integrated with Excel and other Office products. Using Visual Studio Tools for Office, developers can write custom code to run Excel calculations on an HPC cluster utilizing SOA calls. Visual Studio Tools for Office supports the client libraries for Windows HPC Server 2008 R2, enabling the integration of Excel with any service or application that runs on the cluster.

Running Excel User Defined Functions on an HPC Cluster

User-defined functions (UDFs) are a well-established mechanism for extending Excel, enabling functions that are contained in Excel extension libraries (XLLs) to be called from spreadsheet cells like any standard Excel function. Excel 2010 extends this model to the HPC cluster by enabling UDFs to be calculated on an HPC cluster by one or more compute nodes. If a long-running workbook includes multiple independent calls to defined functions and these functions contribute to the overall processing time, then moving those calculations to the cluster can result in significant overall performance improvement. As far as users are con-

cerned, there is no difference between a desktop function and a function running on the cluster – except for better performance.

Running Excel Workbooks on an HPC Cluster

Many complex, long-running workbooks run iteratively – that is, they perform a single calculation over and over, using different sets of input data. Such workbooks might include complex mathematical calculations contained in multiple worksheets, or they might contain complex VBA applications. When a workbook runs iteratively, the best option for parallelizing the calculation can be to run the entire workbook on the cluster.

Windows HPC Server 2008 R2 supports running Office Excel 2010 instances on the compute nodes of an HPC cluster, so that multiple long-running and iterative workbooks can be calculated in parallel to achieve better performance. Many workbooks that run on the desktop can run on the cluster – including workbooks that use Visual Basic for Applications, macros, and third-party add-ins. Support for running Excel workbooks on a cluster also includes features designed to run workbooks without user interaction, providing a robust platform for calculating Excel models without requiring constant oversight. Although this approach can be used to calculate many workbooks on a cluster, some development is required. When workbooks run on the desktop, calculation results are inserted into spreadsheet cells. Because running Excel workbooks on a cluster uses Excel processes running on cluster nodes, the user or developer must define what data is to be calculated and how to retrieve the results. A macro framework is provided that can handle much of this work, and developers can customize the framework or write their own code to manage calculations and results, providing for virtually unlimited flexibility.

because it heavily loads the cluster, the Lizard can be a valuable tool for break-in and detecting issues related to configuration and deployment, networking, power, cooling, and so on.

The Lizard calculates the performance and efficiency of an HPC cluster by automatically running the LINPACK Benchmark several times, analyzing the results of each run and automatically adjusting the parameters used for the subsequent LINPACK run. Eventually, the Lizard determines the parameters that provide optimal LINPACK performance, which is measured in terms of billions of floating-point operations per second (GFLOPS) and percentage efficiency that was achieved at peak performance. After running the Lizard, administrators can review the LINPACK results and save both the results and the parameters that were used to achieve them to a file.

Administrators can run the Lizard either in express tuning mode or in advanced tuning mode. In express tuning mode, the Lizard starts the tuning process immediately, using default values for LINPACK parameters. In advanced tuning mode, administrators can provide specific values to use when the tuning process starts, and can also configure how the tuning process is run.

Microsoft, Windows, Windows Vista, Windows Server, Visual Studio, Excel, Office, Visual Basic, DirectX, Direct3D, Windows PowerShell and certain other trademarks and logos appearing in this brochure, are trademarks or registered trademarks of Microsoft Corporation.



transtec HPC expertise encompasses the Windows world as well. transtec is able to provide customers with Windows HPC systems that integrate seamlessly into their environment. Be it diskful or diskless deployment via WDM, integration into an existing AD environment, or setup and configuration of a WSUS server for centralized update provisioning, transtec gives customers any Windows solution at hand that is needed for High Productivity Computing.

To meet more advanced requirements, by means of Moab Adaptive HPC Suite, transtec as a provider of HPC Professional Services will also set up dynamical deployment solutions for mixed Linux-Windows systems, either by dual-boot or by virtualization techniques.



PARALLEL NFS

THE NEW STANDARD FOR HPC STORAGE

HPC computation results in the terabyte range are not uncommon. The problem in this context is not so much storing the data at rest, but the performance of the necessary copying back and forth in the course of the computation job flow and the dependent job turn-around time. For interim results during a job runtime or for fast storage of input and results data, parallel file systems have established themselves as the standard to meet the ever-increasing performance requirements of HPC storage systems. Parallel NFS is about to become the new standard framework for a parallel file system.

PARALLEL NFS

THE NEW STANDARD FOR HPC STORAGE

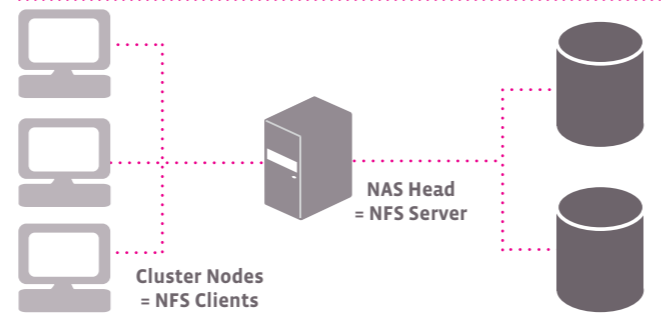
YESTERDAY'S SOLUTION: NFS FOR HPC STORAGE

The original Network File System (NFS) developed by Sun Microsystems at the end of the eighties – now available in version 4.1 – has been established for a long time as a de-facto standard for the provisioning of a global namespace in networked computing.

A very widespread HPC cluster solution includes a central master node acting simultaneously as an NFS server, with its local file system storing input, interim and results data and exporting them to all other cluster nodes.

There is of course an immediate bottleneck in this method: When the load of the network is high, or where there are large numbers of nodes, the NFS server can no longer keep up delivering or receiving the data. In high-performance comput-

FIGURE 1.A CLASSICAL NFS SERVER IS A BOTTLENECK



ing especially, the nodes are interconnected at least once via Gigabit Ethernet, so the sum total throughput is well above what an NFS server with a Gigabit interface can achieve. Even a powerful network connection of the NFS server to the cluster, for example with 10-Gigabit Ethernet, is only a temporary

solution to this problem until the next cluster upgrade. The fundamental problem remains – this solution is not scalable; in addition, NFS is a difficult protocol to cluster in terms of load balancing: either you have to ensure that multiple NFS servers accessing the same data are constantly synchronised, the disadvantage being a noticeable drop in performance or you manually partition the global namespace which is also time-consuming. NFS is not suitable for dynamic load balancing as on paper it appears to be stateless but in reality is, in fact, stateful.

TODAY'S SOLUTION: PARALLEL FILE SYSTEMS

For some time, powerful commercial products have been available to meet the high demands on an HPC storage system. The open-source solution Lustre is widely used in the Linux HPC world, and also several other free as well as commercial parallel file system solutions exist.

What is new is that the time-honoured NFS is to be upgraded, including a parallel version, into an Internet Standard with the aim of interoperability between all operating systems. The original problem statement for parallel NFS access was written by Garth Gibson, a professor at Carnegie Mellon University and founder and CTO of Panasas. Gibson was already a renowned figure being one of the authors contributing to the original paper on RAID architecture from 1988. The original statement from Gibson and Panasas is clearly noticeable in the design of pNFS. The powerful HPC file system developed by Gibson and Panasas, ActiveScale PanFS, with object-based storage devices functioning as central components, is basically the commercial continuation of the “Network-Attached Secure Disk (NASD)” project also developed by Garth Gibson at the Carnegie Mellon University.

PARALLEL NFS

Parallel NFS (pNFS) is gradually emerging as the future standard to meet requirements in the HPC environment. From the industry's as well as the user's perspective, the benefits of utilising standard solutions are indisputable: besides protecting end user investment, standards also ensure a defined level of interoperability without restricting the choice of products available. As a result, less user and administrator training is required which leads to simpler deployment and at the same time, a greater acceptance.

As part of the planned NFS 4.1 Internet Standard, pNFS will not only adopt the semantics of NFS in terms of cache consistency or security, it also represents an easy and flexible extension of the NFS 4 protocol. pNFS is optional, in other words, NFS 4.1 implementations do not have to include pNFS as a feature. The scheduled Internet Standard NFS 4.1 is today (May 2010) presented as IETF RFC 5661.

The pNFS protocol supports a separation of metadata and data: a pNFS cluster comprises so-called storage devices which store the data from the shared file system and a metadata server (MDS), called Director Blade with Panasas – the actual NFS 4.1 server. The metadata server keeps track of which data is stored on which storage devices and how to access the files, the so-called layout. Besides these “striping parameters”, the MDS also manages other metadata including access rights or similar, which is usually stored in a file's inode.

The layout types define which Storage Access Protocol is used by the clients to access the storage devices. Up until now,

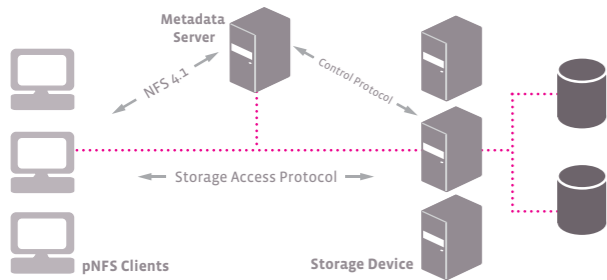
“Outstanding in the HPC world, the ActiveStor solutions provided by Panasas are undoubtedly the only HPC storage solutions that combine highest scalability and performance with a convincing ease of management.”

Thomas Gebert HPC Solution Engineer

PARALLEL NFS

THE NEW STANDARD FOR HPC STORAGE

FIGURE 2 PARALLEL NFS



three potential storage access protocols have been defined for pNFS: file, block and object-based layouts. Last but not least, a Control Protocol is also used by the MDS and storage devices to synchronise status data. This protocol is deliberately unspecified in the standard to give manufacturers certain flexibility. The NFS 4.1 standard does however specify certain conditions which a control protocol has to fulfil, for example, how to deal with the change/modify time attributes of files.

pNFS supports backwards compatibility with non-pNFS compatible NFS 4 clients. In this case, the MDS itself gathers data from the storage devices on behalf of the NFS client and presents the data to the NFS client via NFS 4. The MDS acts as a kind of proxy server – which is e.g. what the Director Blades from Panasas do.

PNFS LAYOUT TYPES

If storage devices act simply as NFS 4 file servers, the file layout is used. It is the only storage access protocol directly specified in the NFS 4.1 standard. Besides the stripe sizes and stripe locations (storage devices), it also includes the NFS file handles which the client needs to use to access the separate file areas. The file layout is compact and static, the striping information does not change even if changes are made to the file enabling multiple pNFS clients to simultaneously cache the layout and avoid synchronisation overhead between clients and the MDS or the MDS and storage devices.

File system authorisation and client authentication can be well implemented with the file layout. When using NFS 4 as the storage access protocol, client authentication merely depends on the security flavor used – when using the RPC-SEC_GSS security flavor, client access is kerberized,



Having a many years' experience in deploying parallel file systems from smaller scales up to hundreds of terabytes capacity and throughputs of several gigabytes per second, transtec chose Panasas, the leader in HPC storage solutions, as the partner for providing highest performance and scalability on the one hand, and ease of management on the other. Therefore, with Panasas as the technology leader, and transtec's overall experience and customer-oriented approach, customers can be assured to get the best possible HPC storage solution available.

PARALLEL NFS

PANASAS HPC STORAGE

for example and the server controls access authorization using specified ACLs and cryptographic processes.

In contrast, the block/volume layout uses volume identifiers and block offsets and extents to specify a file layout. SCSI block commands are used to access storage devices. As the block distribution can change with each write access, the layout must be updated more frequently than with the file layout.

Block-based access to storage devices does not offer any secure authentication option for the accessing SCSI initiator. Secure SAN authorisation is possible with host granularity only, based on World Wide Names (WWNs) with Fibre Channel or Initiator Node Names (IQNs) with iSCSI. The server cannot enforce access control governed by the file system. On the contrary, a pNFS client basically voluntarily abides with the access rights, the storage device has to trust the pNFS client – a fundamental access control problem that is a recurrent issue in the NFS protocol history.

The object layout is syntactically similar to the file layout, but it uses the SCSI object command set for data access to so-called Object-based Storage Devices (OSDs) and is heavily based on the DirectFLOW protocol of the ActiveScale PanFS from Panasas. From the very start, Object-Based Storage Devices were designed for secure authentication and access. So-called capabilities are used for object access which involves the MDS issuing so-called capabilities to the pNFS clients. The ownership of these capabilities represents the authoritative access right to an object.

pNFS can be upgraded to integrate other storage access protocols and operating systems, and storage manufacturers also have the option to ship additional layout drivers for their pNFS implementations.



The foundation of Panasas ActiveStor parallel storage solutions is the Panasas ActiveScale operating environment which includes the PanFS parallel file system, and the DirectFLOW Protocol. The ActiveScale operating environment delivers orders of magnitude performance improvements over traditional file storage architectures and dramatically lowers the cost of managing data storage by supporting Terabytes (TBs) to Petabytes (PBs) of data capacity growth all within a single, easily managed namespace.

OBJECT-BASED ARCHITECTURE

The core principle of the ActiveScale object-based architecture is distributing block management to the storage device in contrast to traditional storage systems that manage blocks on the file server and create a storage bottleneck. Data objects are written to smart object storage devices on a massively parallel scale to remove the performance bottlenecks of traditional designs. Also, data objects can be resized without limitation and independently from other storage system activity, allowing the management of all data within a single seamless namespace and providing independent and parallel growth properties.

PANFS PARALLEL FILE SYSTEM

Designed from the ground up for efficient partitioning of

workload and minimal interdependence between clustered elements, the Panasas PanFS parallel file system turns files into smart data objects and then dynamically distributes data transfer operations across the StorageBlade modules. Panasas DirectorBlade modules provide metadata services and are clustered together to create a highly scalable, fault-tolerant storage architecture enhanced with fine-grained, dynamic load balancing. The DirectorBlade modules coordinate access to the StorageBlade modules while maintaining cache coherency among the clients. To remove potential I/O bottlenecks, active “hot spots” are identified quickly and object migration routines are invoked to move existing data objects to less utilized StorageBlade modules.

DIRECTFLOW PROTOCOL

The performance advantage of Panasas storage is attributed to the out-of-band DirectFLOW protocol that is the foundation of the pNFS standard. It enables direct communication between clients and StorageBlade modules.

This concurrency eliminates the bottleneck of traditional, monolithic storage systems and delivers massive throughput performance improvement. ActiveStor storage clusters also support NFS and CIFS protocols enabling UNIX and Windows applications to take full advantage of the storage cluster with no client agent required.

PREDICTIVE SELF-MANAGEMENT TECHNOLOGIES

Panasas Predictive Self-Management technologies work together to deliver health-monitoring and self-healing capabilities, ensuring the ActiveStor storage cluster always delivers maximum performance and continuous access to data.

PARALLEL NFS

PANASAS HPC STORAGE

©2009 Panasas Incorporated. All rights reserved. Panasas, the Panasas logo, Accelerating Time to Results, ActiveScale, DirectFLOW, DirectorBlade, StorageBlade, PanFS, PanActive and MyPanasas are trademarks or registered trademarks of Panasas, Inc. in the United States and other countries. All other trademarks are the property of their respective owners. Information supplied by Panasas, Inc. is believed to be accurate and reliable at the time of publication, but Panasas, Inc. assumes no responsibility for any errors that may appear in this document. Panasas, Inc. reserves the right, without notice, to make changes in product design, specifications and prices. Information is subject to change without notice.



ActiveScan Monitoring ensures continuous performance and data availability by monitoring data objects, parity, disk media and individual disk drive attributes. If a potential problem is detected, the data objects or the parity can be moved to empty space on the disk or to other StorageBlade modules, eliminating reconstruction. If a reconstruction is required, all DirectorBlade modules cooperate in parallel to speed reconstruction up to 10x the rate of conventional RAID controllers, completing rebuild of an 2 TB blade in approximately 30 minutes. Real time monitoring of client load generation automatically identifies user performance bottlenecks and notifies administrators.

Tiered Parity Data Protection is a comprehensive architecture that executes appropriate families of error detection and correction codes. Three tiers of protection are independent, yet complementary to each other, and collectively provide the most comprehensive and scalable reliability architecture available today for high performance storage. Vertical Parity isolates and repairs media errors at the disk sector level. Horizontal Parity enables faster and more efficient RAID reconstructions. Network Parity identifies silent data corruption.

ActiveGuard Failover provides protection from network and metadata service failures. In the event that a DirectorBlade module encounters a problem, the metadata services it is running are automatically transferred to another DirectorBlade module within the storage cluster. Storage clusters may be configured with redundant network switches allowing ActiveGuard failover to the spare switch in the event of network failure in the data path of the primary switch.

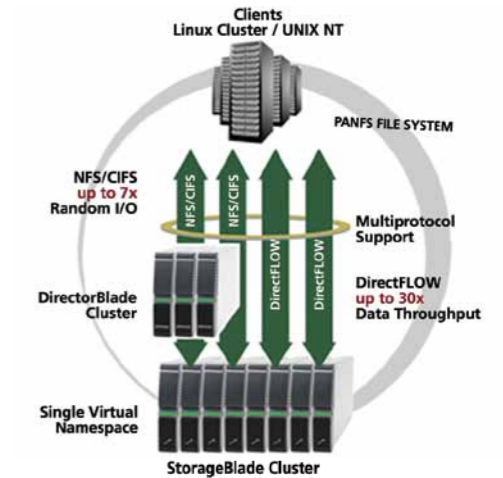
ActiveImage Snapshots creates and recovers snapshots in seconds. Panasas ActiveImage snapshots provide enterprise class data protection through point-in-time copies that can be easily retrieved at any time by a user or system administrator. Using intelligent copy-on-write technology, only the changes made to data are recorded limiting the overhead required and enabling maximum flexibility in snapshot scheduling and administration.

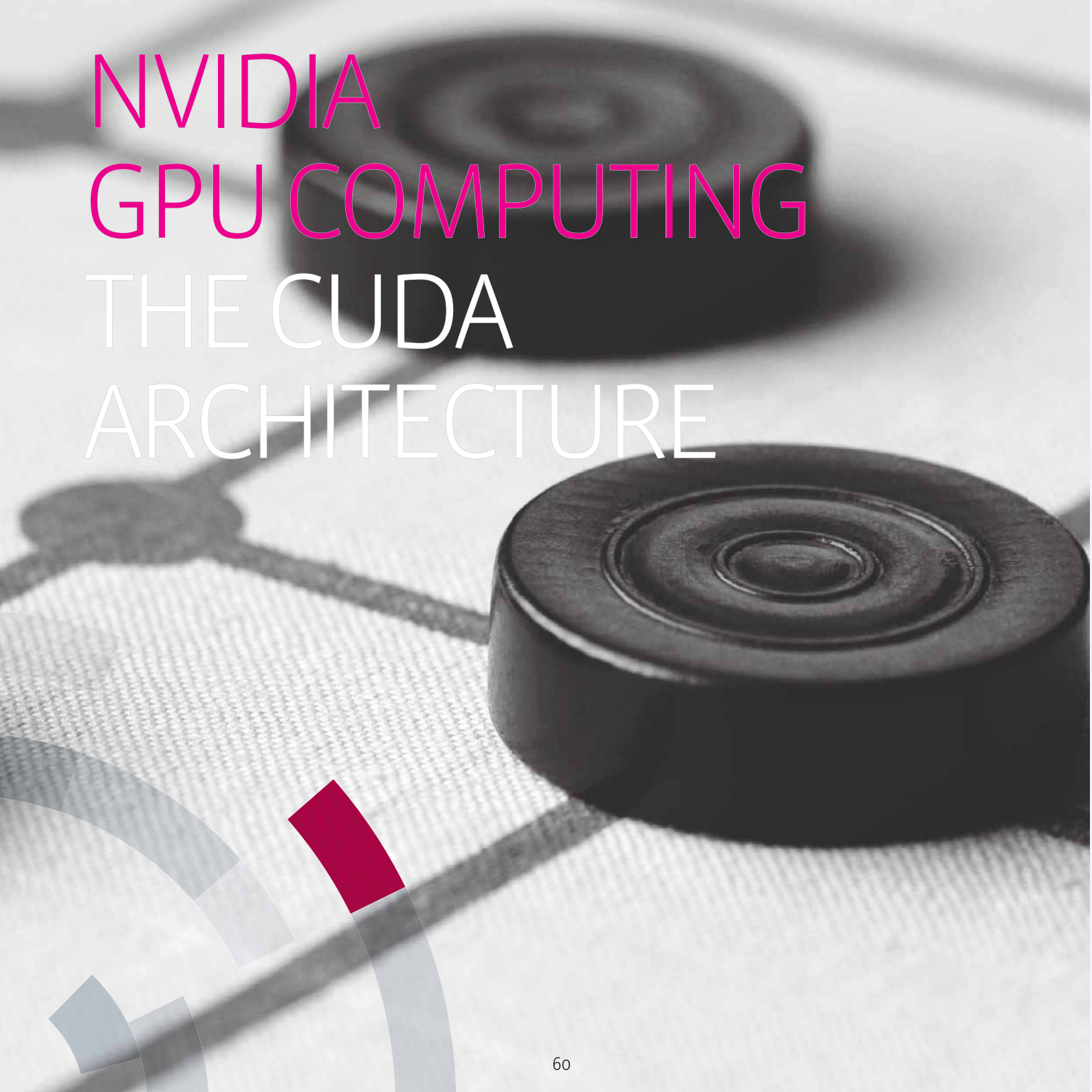
NDMP Backup and Restore Software provides a standards-compliant NDMP instruction set communication interface between the PanFS parallel file system and third-party backup and recovery management applications. The NDMP software provides integration with ActiveImage snapshots, resulting in a granular, flexible and reliable backup and restore capability. Combining industry-leading backup and recovery applications from vendors such as Symantec, Veritas and EMC Legato with the Panasas NDMP functionality provides a cost effective and reliable solution for backup and recovery that greatly enhances data protection.

THE PANASAS PROMISE: ACCELERATE TIME TO RESULTS AND MAXIMIZE ROI

Panasas offers comprehensive storage solutions that deliver the highest bandwidth and I/O performance, scalable performance and capacity, and simple unified management. Panasas integrated hardware and software solutions have been deployed in hundreds of installations around the globe, helping organizations accelerate time to results and maximizing ROI from data intensive and high performance computing environments.

The DirectFLOW protocol moves the metadata manager out of the data transfer path enabling multiple parallel I/O paths to be created between the storage cluster and client nodes. DirectorBlade modules provide simultaneous metadata services for DirectFLOW clients and support clustered NFS and CIFS access to the storage cluster. The DirectFLOW protocol enables direct communication between clients and storage.





NVIDIA GPU COMPUTING THE CUDA ARCHITECTURE

Graphics chips started as fixed function graphics pipelines. Over the years, these graphics chips became increasingly programmable, which led NVIDIA to introduce the first GPU or Graphics Processing Unit. In the 1999-2000 timeframe, computer scientists in particular, along with researchers in fields such as medical imaging and electromagnetics started using GPUs for running general purpose computational applications. They found the excellent floating point performance in GPUs led to a huge performance boost for a range of scientific applications. This was the advent of the movement called GPGPU or General Purpose computing on GPUs. The problem was that GPGPU required using graphics programming languages like OpenGL and Cg to program the GPU. Developers had to make their scientific applications look like graphics applications and map them into problems that drew triangles and polygons. This limited the accessibility of tremendous performance of GPUs for science.

NVIDIA realized the potential to bring this performance to the larger scientific community and decided to invest in modifying the GPU to make it fully programmable for scientific applications and added support for high-level languages like C and C++. This led to the CUDA architecture for the GPU.

NVIDIA GPU COMPUTING

THE CUDA ARCHITECTURE

WHAT IS GPU COMPUTING?

GPU computing is the use of a GPU (graphics processing unit) to do general purpose scientific and engineering computing. The model for GPU computing is to use a CPU and GPU together in a heterogeneous computing model. The sequential part of the application runs on the CPU and the computationally intensive part runs on the GPU. From the user's perspective, the application just runs faster because it is using the high-performance of the GPU to boost performance.

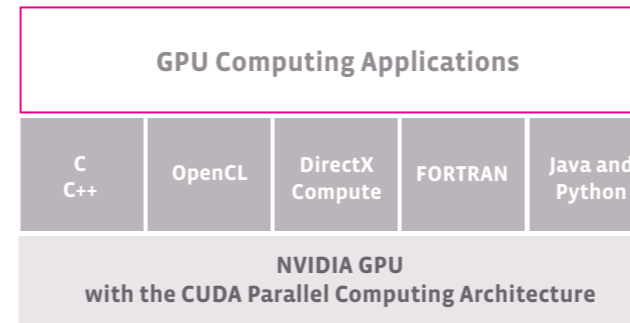
The application developer has to modify the application to take the compute-intensive kernels and map them to the GPU. The rest of the application remains on the CPU. Mapping a function to the GPU involves rewriting the function to expose the parallelism in the function and adding "C" keywords to move data to and from the GPU.

GPU computing is enabled by the massively parallel architecture of NVIDIA's GPUs called the CUDA architecture. The CUDA architecture consists of 100s of processor cores that operate together to crunch through the data set in the application.

CUDA PARALLEL ARCHITECTURE AND PROGRAMMING MODEL

The CUDA parallel hardware architecture is accompanied by the CUDA parallel programming model that provides a set of abstractions that enable expressing fine-grained and coarse-grain data and task parallelism. The programmer can choose to express the parallelism in high-level languages such as C, C++, Fortran or driver APIs such as OpenCL and DirectX-11 Compute.

FIGURE 1 THE CUDA PARALLEL ARCHITECTURE



The CUDA parallel programming model guides programmers to partition the problem into coarse sub-problems that can be solved independently in parallel. Fine grain parallelism in the sub-problems is then expressed such that each sub-problem can be solved cooperatively in parallel. The CUDA GPU architecture and the corresponding CUDA parallel computing model are now widely deployed with 100s of applications and nearly a 1000 published research papers.

GPU COMPUTING WITH CUDA

NVIDIA CUDA technology leverages the massively parallel processing power of NVIDIA GPUs. The CUDA architecture is a revolutionary parallel computing architecture that delivers the performance of NVIDIA's world-renowned graphics processor technology to general-purpose GPU Computing. Applications that run on the CUDA architecture can take advantage of an installed base of over one hundred million CUDA-enabled GPUs in desktop and notebook computers, professional workstations, and supercomputer clusters.

With the CUDA architecture and tools, developers are achieving dramatic speedups in fields such as medical imaging and natural resource exploration, and creating breakthrough applications in areas such as image recognition and real-time HD video playback and encoding. CUDA enables this unprecedented performance via standard APIs such as the soon to be released OpenCL and DirectX Compute, and high level programming languages such as C/C++, Fortran, Java, Python, and the Microsoft .NET Framework.

CUDA: THE DEVELOPER'S VIEW

The CUDA package includes three important components: the CUDA Driver API (also known as "Low-Level API"), the CUDA toolkit (the actual development environment including runtime libraries) and a Software Development Kit (CUDA SDK) with code examples.

The CUDA toolkit is in principle a C development environment and includes the actual compiler (nvcc), an update of the PathScale C compiler, optimized FFT and BLAS libraries as well as a visual profiler (cudaprof), a gdb-based debugger (cudagdb), shared libraries for the runtime environment for CUDA programs (the "Runtime API") and last but not least, comprehensive documentation including a developer's manual.

The CUDA Developer SDK includes examples with source codes for matrix calculation, pseudo random number generators, image convolution, wavelet calculations and a lot more besides.

"We are very proud to be one of the leading providers of Tesla systems who are able to combine the overwhelming power of NVIDIA Tesla systems with the fully engineered and thoroughly tested transtec hardware to a total Tesla-based solution."

Norbert Zeidler Senior System Engineer/ HPC Solution Engineer

NVIDIA GPU COMPUTING

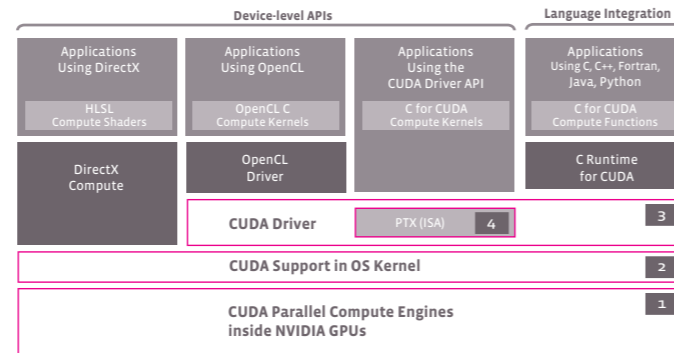
THE CUDA ARCHITECTURE

THE CUDA ARCHITECTURE

The CUDA Architecture consists of several components, in the boxes below:

- 1 Parallel compute engines inside NVIDIA GPUs
- 2 OS kernel-level support for hardware initialization, configuration, etc.
- 3 User-mode driver, which provides a device-level API for developers
- 4 PTX instruction set architecture (ISA) for parallel computing kernels and functions

FIGURE 2 THE CUDA PROGRAMMING MODEL



The CUDA Software Development Environment supports two different programming interfaces:

- II A device-level programming interface, in which the application uses DirectX Compute, OpenCL or the CUDA Driver API directly to configure the GPU, launch compute kernels, and read back results
- II A language integration programming interface, in which an application uses the C Runtime for CUDA and developers use a small set of extensions to indicate which compute functions should be performed on the GPU instead of the CPU.

When using the device-level programming interface, developers write compute kernels in separate files using the kernel language supported by their API of choice. DirectX Compute kernels (aka “compute shaders”) are written in HLSL. OpenCL kernels are written in a C-like language called “OpenCL C”. The CUDA Driver API accepts kernels written in C or PTX assembler.

When using the language integration programming interface, developers write compute functions in C and the C Runtime for CUDA automatically handles setting up the GPU and executing the compute functions. This programming interface enables developers to take advantage of native support for high-level languages such as C, C++, Fortran, Java, Python, and more, reducing code complexity and development costs through type integration and code integration:

- II Type integration allows standard types as well as vector types and user-defined types (including structs) to be used seamlessly across functions that are executed on the CPU and functions that are executed on the GPU.
- II Code integration allows the same function to be called from functions that will be executed on the CPU and functions that will be executed on the GPU
- II When necessary to distinguish functions that will be executed on the CPU from those that will be executed on the GPU, the term C for CUDA is used to describe the small set of extensions that allow developers to specify which functions will be executed on the GPU, how GPU memory will be used, and how the parallel processing capabilities of the GPU will be used by the application

THE G80 ARCHITECTURE

NVIDIA’s GeForce 8800 was the product that gave birth to the new GPU Computing model. Introduced in November 2006, the G80 based GeForce 8800 brought several key innovations to GPU Computing:

- II G80 was the first GPU to support C, allowing programmers to use the power of the GPU without having to learn a new programming language
- II G80 was the first GPU to replace the separate vertex and pixel pipelines with a single, unified processor that executed vertex, geometry, pixel, and computing programs
- II G80 was the first GPU to utilize a scalar thread processor, eliminating the need for programmers to manually manage vector registers
- II G80 introduced the single-instruction multiple-thread (SIMT) execution model where multiple independent threads execute concurrently using a single instruction
- II G80 introduced shared memory and barrier synchronization for inter-thread communication

In June 2008, NVIDIA introduced a major revision to the G80 architecture. The second generation unified architecture – GT200 (first introduced in the GeForce GTX 280, Quadro FX 5800, and Tesla T10 GPUs) – increased the number of streaming processor cores (subsequently referred to as CUDA cores) from 128 to 240. Each processor register file was doubled in

NVIDIA GPU COMPUTING

CODENAME "FERMI"

size, allowing a greater number of threads to execute on-chip at any given time. Hardware memory access coalescing was added to improve memory access efficiency. Double precision floating point support was also added to address the needs of scientific and high-performance computing (HPC) applications.

When designing each new generation GPU, it has always been the philosophy at NVIDIA to improve both existing application performance and GPU programmability; while faster application performance brings immediate benefits, it is the GPU's relentless advancement in programmability that has allowed it to evolve into the most versatile parallel processor of our time.

CODENAME "FERMI"

The Fermi architecture is the most significant leap forward in GPU architecture since the original G80. G80 was the initial vision of what a unified graphics and computing parallel processor should look like. GT200 extended the performance and functionality of G80. With Fermi, NVIDIA has taken everything learned from the two prior processors and all the applications that were written for them, and employed a completely new approach to design to create the world's first computational GPU. When they started laying the groundwork for Fermi, they gathered extensive user feedback on GPU computing since the introduction of G80 and GT200, and focused on the following key areas for improvement:

- II Improved double precision performance: while single precision floating point performance was on the order of ten times the performance of desktop CPUs, some GPU computing applications desired more double precision performance as well

- II ECC support: ECC allows GPU computing users to safely deploy large numbers of GPUs in datacenter installations, and also ensure data-sensitive applications like medical imaging and financial options pricing are protected from memory errors
- II True cache hierarchy: some parallel algorithms were unable to use the GPU's shared memory, and users requested a true cache architecture to aid them
- II More shared memory: many CUDA programmers requested more than 16 KB of SM shared memory to speed up their applications
- II Faster context switching: users requested faster context switches between application programs and faster graphics and compute interoperation
- II Faster atomic operations: users requested faster read-modify-write atomic operations for their parallel algorithms

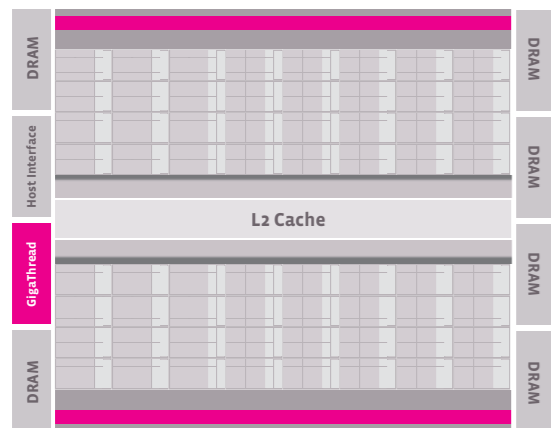
With these requests in mind, the Fermi team designed a processor that greatly increases raw compute horsepower, and through architectural innovations, also offers dramatically increased programmability and compute efficiency. The key architectural highlights of Fermi are:

- II Third Generation Streaming Multiprocessor (SM):
 - 32 CUDA cores per SM, 4x over GT200
 - 8x the peak double precision floating point performance over GT200
 - Dual Warp Scheduler simultaneously schedules and dispatches instructions from two independent warps
 - 64 KB of RAM with a configurable partitioning of

shared memory and L1 cache

- II Second Generation Parallel Thread Execution ISA:
 - Unified Address Space with Full C++ Support
 - Optimized for OpenCL and DirectCompute
 - Full IEEE 754-2008 32-bit and 64-bit precision
 - Full 32-bit integer path with 64-bit extensions
 - Memory access instructions to support Transition to 64-bit addressing
 - Improved performance through predication
- II Improved Memory Subsystem:
 - NVIDIA Parallel DataCache hierarchy with configurable L1 and Unified L2 Caches
 - First GPU with ECC memory support
 - Greatly improved atomic memory operation performance
- II NVIDIA GigaThread Engine:
 - 10x faster application context switching
 - Concurrent kernel execution
 - Out of Order thread block execution
 - Dual overlapped memory transfer engines

FIGURE 3 IMPROVED MEMORY SUBSYSTEM



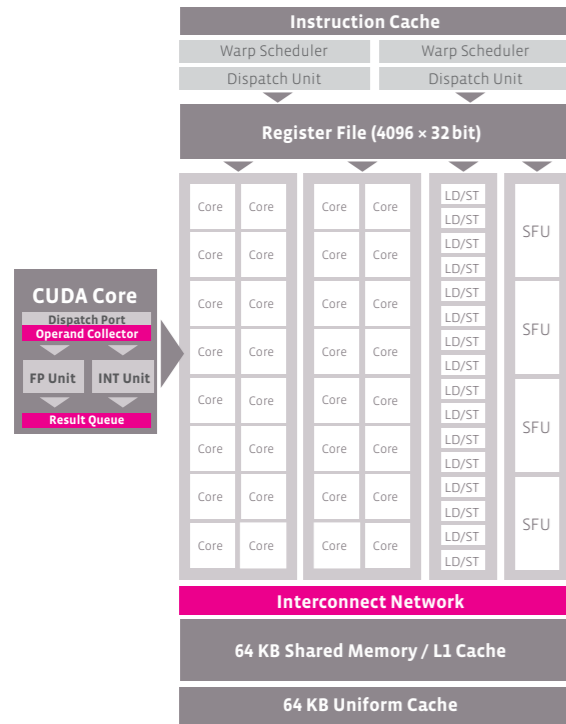
AN OVERVIEW OF THE FERMI ARCHITECTURE

The first Fermi based GPU, implemented with 3.0 billion transistors, features up to 512 CUDA cores. A CUDA core executes a floating point or integer instruction per clock for a thread. The 512 CUDA cores are organized in 16 SMs of 32 cores each. The GPU has six 64-bit memory partitions, for a 384-bit memory interface, supporting up to a total of 6 GB of GDDR5 DRAM memory. A host interface connects the GPU to the CPU via PCI-Express. The GigaThread global scheduler distributes thread blocks to SM thread schedulers.

NVIDIA GPU COMPUTING

CODENAME “FERMI”

FIGURE 4 THIRD GENERATION STREAMING MULTIPROCESSOR



THIRD GENERATION STREAMING MULTIPROCESSOR

The third generation SM introduces several architectural innovations that make it not only the most powerful SM yet built, but also the most programmable and efficient.

512 High Performance CUDA Cores

Each SM features 32 CUDA processors – a fourfold increase over prior SM designs. Each CUDA processor has a fully pipelined integer arithmetic logic unit (ALU) and floating point unit (FPU). Prior GPUs used IEEE 754-1985 floating point arithmetic. The Fermi architecture implements the new IEEE 754-2008 floating-point standard, providing the fused multiply-add (FMA) instruction for both single and double precision arithmetic. FMA improves over a multiply-add (MAD) instruction by doing the multiplication and addition with a single final rounding step, with no loss of precision in the addition. FMA is more accurate than performing the operations separately. GT200 implemented double precision FMA.

In GT200, the integer ALU was limited to 24-bit precision for multiply operations; as a result, multi-instruction emulation sequences were required for integer arithmetic. In Fermi, the newly designed integer ALU supports full 32-bit precision for all instructions, consistent with standard programming language requirements. The integer ALU is also optimized to efficiently support 64-bit and extended precision operations. Various instructions are supported, including Boolean, shift, move, compare, convert, bit-field extract, bit-reverse insert, and population count.

16 Load/Store Units

Each SM has 16 load/store units, allowing source and destination addresses to be calculated for sixteen threads per clock. Supporting units load and store the data at each address to cache or DRAM.

Four Special Function Units

Special Function Units (SFUs) execute transcendental instructions such as sine, cosine, reciprocal, and square root. Each SFU executes one instruction per thread, per clock; a warp executes over eight clocks. The SFU pipeline is decoupled from the dispatch unit, allowing the dispatch unit to issue to other execution units while the SFU is occupied.

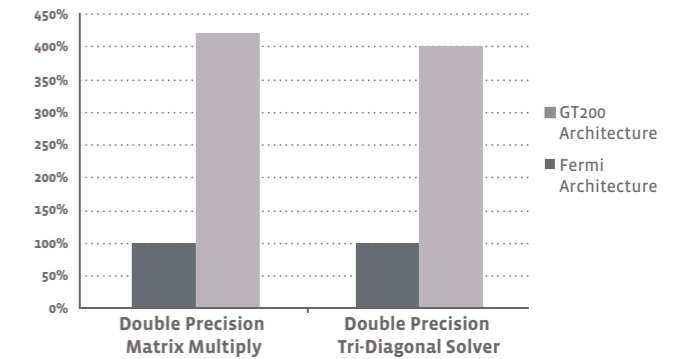
Designed for Double Precision

Double precision arithmetic is at the heart of HPC applications such as linear algebra, numerical simulation, and quantum chemistry. The Fermi architecture has been specifically designed to offer unprecedented performance in double precision; up to 16 double precision fused multiply-add operations can be performed per SM, per clock, a dramatic improvement over the GT200 architecture.

DUAL WARP SCHEDULER

The SM schedules threads in groups of 32 parallel threads called warps. Each SM features two warp schedulers and two instruction dispatch units, allowing two warps to be issued and executed concurrently. Fermi’s dual warp scheduler selects two warps, and issues one instruction from each warp to a group of sixteen cores, sixteen load/store units, or four SFUs. Because warps execute independently, Fermi’s scheduler does not need to check for dependencies from within the instruction stream. Using this elegant model of dual-issue, Fermi achieves near peak hardware performance.

FIGURE 5 DOUBLE PRECISION APPLICATION PERFORMANCE

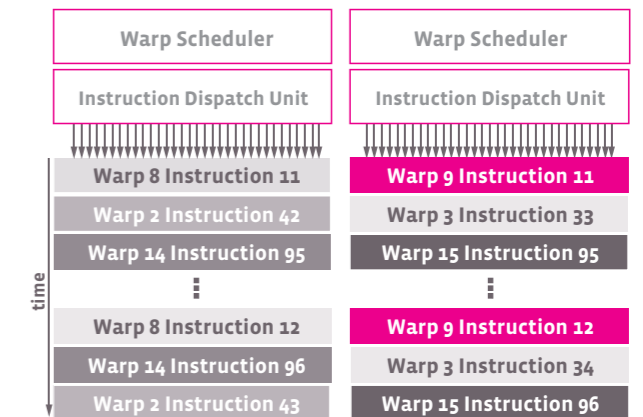


Most instructions can be dual issued; two integer instructions, two floating instructions, or a mix of integer, floating point, load, store, and SFU instructions can be issued concurrently. Double precision instructions do not support dual dispatch with any other operation.

64 KB Configurable Shared Memory and L1 Cache

One of the key architectural innovations that greatly improved both the programmability and performance of GPU

FIGURE 6 DUAL WARP SCHEDULER



NVIDIA GPU COMPUTING

CODENAME "FERMI"

NVIDIA, GeForce, Tesla, CUDA, PhysX, GigaThread, NVIDIA Parallel DataCache and certain other trademarks and logos appearing in this brochure, are trademarks or registered trademarks of NVIDIA Corporation.



applications is on-chip shared memory. Shared memory enables threads within the same thread block to cooperate, facilitates extensive reuse of on-chip data, and greatly reduces off-chip traffic. Shared memory is a key enabler for many high-performance CUDA applications.

G80 and GT200 have 16 KB of shared memory per SM. In the Fermi architecture, each SM has 64 KB of on-chip memory that can be configured as 48 KB of shared memory with 16 KB of L1 cache or as 16 KB of shared memory with 48 KB of L1 cache.

For existing applications that make extensive use of shared memory, tripling the amount of shared memory yields significant performance improvements, especially for problems that are bandwidth constrained. For existing applications that use shared memory as software managed cache, code can be streamlined to take advantage of the hardware caching system, while still having access to at least 16 KB of shared memory for explicit thread cooperation. Best of all, applications that do not use shared memory automatically benefit from the L1 cache, allowing high performance CUDA programs to be built with minimum time and effort.

FIGURE 7 RADIX SORT USING SHARED MEMORY

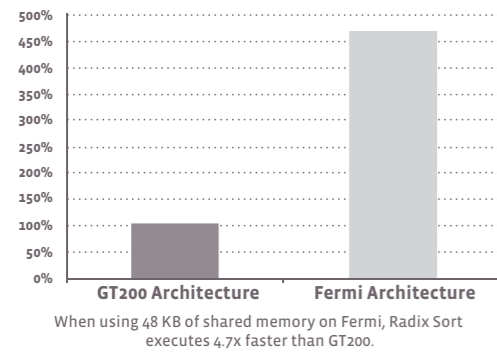
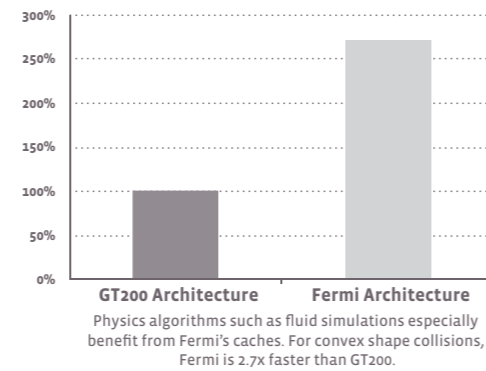


FIGURE 8 PHYSX FLUID COLLISION FOR CONVEX SHAPES



GPU	G80	GT200	Fermi
Transistors	681 million	1.4 billion	3.0 billion
CUDA Cores	128	240	512
Double Precision Floating Point Capability	None	30 FMA ops/clock	256 FMA ops/clock
Single Precision Floating Point Capability	128 MAD ops/clock	240 MAD ops/clock	512 FMA ops/clock
Special Function Units (SFUs) / SM	2	2	4
Warp schedulers (per SM)	1	1	2
Shared Memory (per SM)	16 KB	16 KB	Configurable 48 KB or 16 KB
L1 Cache (per SM)	None	None	Configurable 16 KB or 48 KB
L2 Cache	None	None	768 KB
ECC Memory Support	No	No	Yes
Concurrent Kernels	No	No	Up to 16
Load/Store Address Width	32-bit	32-bit	64-bit

GIGATHREAD THREAD SCHEDULER

One of the most important technologies of the Fermi architecture is its two-level, distributed thread scheduler. At the chip level, a global work distribution engine schedules thread blocks to various SMs, while at the SM level, each warp scheduler distributes warps of 32 threads to its execution units. The first generation GigaThread engine introduced in G80 managed up to 12,288 threads in realtime. The Fermi architecture improves on this foundation by providing not only greater thread throughput, but dramatically faster context switching, concurrent kernel execution, and improved thread block scheduling.

10x Faster Application Context Switching

Like CPUs, GPUs support multitasking through the use of context switching, where each program receives a time slice of the processor's resources. The Fermi pipeline is optimized to reduce the cost of an application context switch to below 25 microseconds, a significant improvement over last generation GPUs. Besides improved performance, this allows developers to create applications that take greater advantage of frequent kernel-to-kernel communication, such as fine-grained interoperation between graphics and PhysX applications.

NVIDIA GPU COMPUTING

INTRODUCING NVIDIA NEXUS

FIGURE 9 SERIAL KERNEL EXECUTION

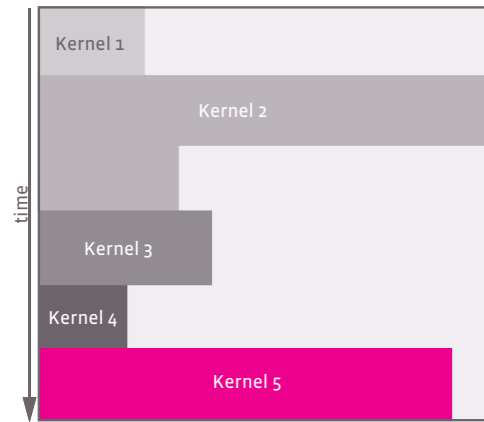
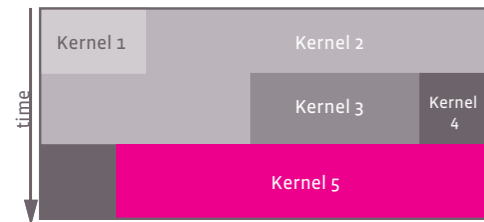


FIGURE 10 CONCURRENT KERNEL EXECUTION



Concurrent Kernel Execution

Fermi supports concurrent kernel execution, where different kernels of the same application context can execute on the GPU at the same time. Concurrent kernel execution allows programs that execute a number of small kernels to utilize the whole GPU. For example, a PhysX program may invoke a fluids solver and a rigid body solver which, if executed sequentially, would use only half of the available thread processors. On the Fermi architecture, different kernels of the same CUDA context can execute concurrently, allowing maximum utilization of GPU resources. Kernels from different application contexts can still run sequentially with great efficiency thanks to the improved context switching performance.

INTRODUCING NVIDIA NEXUS

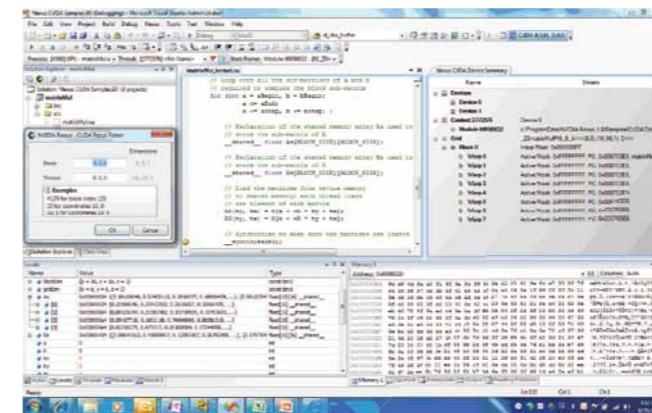
NVIDIA Nexus is the first development environment designed specifically to support massively parallel CUDA C, OpenCL, and DirectCompute applications. It bridges the productivity gap between CPU and GPU code by bringing parallel-aware hardware source code debugging and performance analysis directly into Microsoft Visual Studio, the most widely used integrated application development environment under Microsoft Windows.

Nexus allows Visual Studio developers to write and debug GPU source code using exactly the same tools and interfaces that are used when writing and debugging CPU code, including source and data breakpoints, and memory inspection. Further-

more, Nexus extends Visual Studio functionality by offering tools to manage massive parallelism, such as the ability to focus and debug on a single thread out of the thousands of threads running parallel, and the ability to simply and efficiently visualize the results computed by all parallel threads.

Nexus is the perfect environment to develop co-processing applications that take advantage of both the CPU and GPU. It captures performance events and information across both processors, and presents the information to the developer on a single correlated timeline. This allows developers to see how their application behaves and performs on the entire system, rather than through a narrow view that is focused on a particular subsystem or processor.

FIGURE 11 NVIDIA NEXUS



transtec has strived for developing well-engineered GPU Computing solutions from the very beginning of the Tesla era. From High-Performance GPU Workstations to rack-mounted Tesla server solutions, transtec has a broad range of specially designed systems available. As an NVIDIA Tesla Preferred Provider (TPP), transtec is able to provide customers with the latest NVIDIA GPU technology as well as fully-engineered hybrid systems and Tesla Preconfigured Clusters. Thus, customers can be assured that transtec's large experience in HPC cluster solutions is seamlessly brought into the GPU computing world. Performance Engineering made by transtec.

BLAS (“Basic Linear Algebra Subprograms”)

Routines that provide standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. Because the BLAS are efficient, portable, and widely available, they are commonly used in the development of high quality linear algebra software, e.g. → LAPACK . Although a model Fortran implementation of the BLAS is available from netlib in the BLAS library, it is not expected to perform as well as a specially tuned implementation on most high-performance computers – on some machines it may give much worse performance – but it allows users to run → LAPACK software on machines that do not offer any other implementation of the BLAS.

CISC (“complex instruction-set computer”)

A computer instruction set architecture (ISA) in which each instruction can execute several low-level operations, such as a load from memory, an arithmetic operation, and a memory store, all in a single instruction. The term was retroactively coined in contrast to reduced instruction set computer (RISC). The terms RISC and CISC have become less meaningful with the continued evolution of both CISC and RISC designs and implementations, with modern processors also decoding and splitting more complex instructions

into a series of smaller internal micro-operations that can thereby be executed in a pipelined fashion, thus achieving high performance on a much larger subset of instructions.

cluster

Aggregation of several, mostly identical or similar systems to a group, working in parallel on a problem. Previously known as Beowulf Clusters, HPC clusters are composed of commodity hardware, and are scalable in design. The more machines are added to the cluster, the more performance can in principle be achieved.

control protocol

Part of the → parallel NFS standard

CUDA driver API

Part of the → CUDA architecture

CUDA SDK

Part of the → CUDA architecture

CUDA toolkit

Part of the → CUDA architecture

CUDA (“Compute Uniform Device Architecture”)

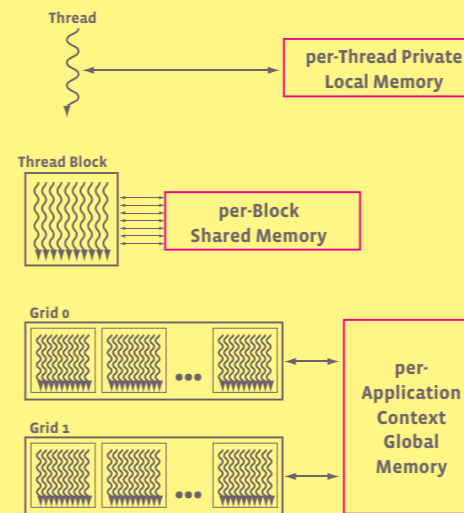
A parallel computing architecture developed by NVIDIA.

CUDA is the computing engine in NVIDIA graphics processing units or GPUs that is accessible to software developers through industry standard programming languages. Programmers use ‘C for CUDA’ (C with NVIDIA extensions), compiled through a PathScale Open64 C compiler, to code algorithms for execution on the GPU. CUDA architecture supports a range of computational interfaces including → OpenCL and → DirectCompute. Third party wrappers are also available for Python, Fortran, Java and Matlab. CUDA works with all NVIDIA GPUs from the G8X series onwards, including GeForce, Quadro and the Tesla line. CUDA provides both a low level API and a higher level API. The initial CUDA SDK was

made public on 15 February 2007, for Microsoft Windows and Linux. Mac OS X support was later added in version 2.0, which supersedes the beta released February 14, 2008.

CUDA is the hardware and software architecture that enables NVIDIA GPUs to execute programs written with C, C++, Fortran, → OpenCL, → DirectCompute, and other languages. A CUDA program calls parallel kernels. A kernel executes in parallel across a set of parallel threads. The programmer or compiler organizes these threads in thread blocks and grids of thread blocks. The GPU instantiates a kernel program on a grid of parallel thread blocks. Each thread within a thread block executes an instance of the kernel, and has a thread ID within its thread block, program counter, registers, per-thread private memory, inputs, and output results.

A thread block is a set of concurrently executing threads that can cooperate among themselves through barrier synchronization and shared memory. A thread block has a block ID within its grid. A grid is an array of thread blocks that execute the same kernel, read inputs from global memory, write results to global memory, and synchronize between dependent kernel calls. In the CUDA parallel programming model, each thread has a per-thread private memory space used for register spills, function calls, and C automatic array variables. Each thread block has a per-block shared memory space used for inter-thread com-



munication, data sharing, and result sharing in parallel algorithms. Grids of thread blocks share results in global memory space after kernel-wide global synchronization.

CUDA's hierarchy of threads maps to a hierarchy of processors on the GPU; a GPU executes one or more kernel grids; a streaming multiprocessor (SM) executes one or more thread blocks; and CUDA cores and other execution units in the SM execute threads. The SM executes threads in groups of 32 threads called a warp. While programmers can generally ignore warp execution for functional correctness and think of programming one thread, they can greatly improve performance by having threads in a warp execute the same code path and access memory in nearby addresses. See the main article "GPU Computing" for further details.

DirectCompute

An application programming interface (API) that supports general-purpose computing on graphics processing units (GPUs) on Microsoft Windows Vista or Windows 7. DirectCompute is part of the Microsoft DirectX collection of APIs. DirectCompute will initially be released with the DirectX 11 API but runs on both DirectX 10 and DirectX 11 GPUs.

floating point standard (IEEE 754)

The most widely-used standard for floating-point computation,

and is followed by many hardware (CPU and FPU) and software implementations. Many computer languages allow or require that some or all arithmetic be carried out using IEEE 754 formats and operations. The current version is IEEE 754-2008, which was published in August 2008; the original IEEE 754-1985 was published in 1985. The standard defines arithmetic formats, interchange formats, rounding algorithms, operations, and exception handling. The standard also includes extensive recommendations for advanced exception handling, additional operations (such as trigonometric functions), expression evaluation, and for achieving reproducible results. The standard defines single-precision, double-precision, as well as 128-byte quadruple-precision floating point numbers. In the proposed 754r version, the standard also defines the 2-byte half-precision number format.

grid (in CUDA architecture)

Part of the → CUDA programming model

HLSL ("High Level Shader Language")

The High Level Shader Language or High Level Shading Language (HLSL) is a proprietary shading language developed by Microsoft for use with the Microsoft DirectX 3D API. It is analogous to the GLSL shading language used with the OpenGL standard. It is very similar to the NVIDIA Cg shading language, as it was developed alongside it.

HLSL programs come in three forms, vertex shaders, geometry shaders, and pixel (or fragment) shaders. A vertex shader is executed for each vertex that is submitted by the application, and is primarily responsible for transforming the vertex from object space to view space, generating texture coordinates, and calculating lighting coefficients such as the vertex's tangent, binormal and normal vectors. When a group of vertices (normally 3, to form a triangle) come through the vertex shader, their output position is interpolated to form pixels within its area; this process is known as rasterisation. Each of these pixels comes through the pixel shader, whereby the resultant screen colour is calculated.

Optionally, an application using a DirectX 10 interface and DirectX 10 hardware may also specify a geometry shader. This shader takes as its input the three vertices of a triangle and uses this data to generate (or tessellate) additional triangles, which are each then sent to the rasterizer.

InfiniBand

Switched fabric communications link primarily used in HPC.

	SDR	DDR	QDR
1X	2 Gbit/s	4 Gbit/s	8 Gbit/s
4X	8 Gbit/s	16 Gbit/s	32 Gbit/s
12X	24 Gbit/s	48 Gbit/s	96 Gbit/s

Its features include quality of service and failover, and it is designed to be scalable. The InfiniBand architecture specification defines a connection between processor nodes and high performance I/O nodes such as storage devices. The serial connection's signalling rate is 2.5 Gbit/s in each direction per connection. Links use 8B/10B encoding – every 10 bits sent carry 8 bits of data – so that the useful data transmission rate is four-fifths the raw rate. Thus single, double, and quad data rates carry 2, 4, or 8 Gbit/s respectively. Links can be aggregated in units of 4 or 12, called 4X or 12X.

iSER ("iSCSI Extensions for RDMA")

A protocol that maps the iSCSI protocol over a network that provides RDMA services (like → iWARP or → InfiniBand). This permits data to be transferred directly into SCSI I/O buffers without intermediate data copies. The Datamover Architecture (DA) defines an abstract model in which the movement of data between iSCSI end nodes is logically separated from the rest of the iSCSI protocol. iSER is one Datamover protocol. The interface between the iSCSI and a Datamover protocol, iSER in this case, is called Datamover Interface (DI).

iWARP ("Internet Wide Area RDMA Protocol")

An Internet Engineering Task Force (IETF) update of the RDMA Consortium's → RDMA over TCP standard. This later standard is zero-copy transmission over legacy TCP. Because

a kernel implementation of the TCP stack is a tremendous bottleneck, a few vendors now implement TCP in hardware. This additional hardware is known as the TCP offload engine (TOE). TOE itself does not prevent copying on the receive side, and must be combined with RDMA hardware for zero-copy results. The main component is the Data Direct Protocol (DDP), which permits the actual zero-copy transmission. The transmission itself is not performed by DDP, but by TCP.

kernel (in CUDA architecture)

Part of the → CUDA programming model

LAM/MPI

A high-quality open-source implementation of the → MPI specification, including all of MPI-1.2 and much of MPI-2. Superseded by the → OpenMPI implementation

LAPACK (“linear algebra package”)

Routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The original goal of the LAPACK project was to make the widely used EISPACK and → LINPACK libraries run efficiently on shared-memory vector and parallel processors. LAPACK routines are

written so that as much as possible of the computation is performed by calls to the → BLAS library. While → LINPACK and EISPACK are based on the vector operation kernels of the Level 1 BLAS, LAPACK was designed at the outset to exploit the Level 3 BLAS. Highly efficient machine-specific implementations of the BLAS are available for many modern high-performance computers. The BLAS enable LAPACK routines to achieve high performance with portable software.

layout

Part of the → parallel NFS standard. Currently three types of layout exist: file-based, block/volume-based, and object-based, the latter making use of → object-based storage devices

LINPACK

A collection of Fortran subroutines that analyze and solve linear equations and linear least-squares problems. LINPACK was designed for supercomputers in use in the 1970s and early 1980s. LINPACK has been largely superseded by → LAPACK, which has been designed to run efficiently on shared-memory, vector supercomputers. LINPACK makes use of the → BLAS libraries for performing basic vector and matrix operations. The LINPACK benchmarks are a measure of a system’s floating point computing power and measure how fast a computer solves a dense N by N system of linear equations $Ax=b$, which is a common task in engineering. The solution is obtained by Gaussian elimination

with partial pivoting, with $\frac{2}{3} \cdot N^3 + 2 \cdot N^2$ floating point operations. The result is reported in millions of floating point operations per second (MFLOP/s, sometimes simply called MFLOPS).

LNET

Communication protocol in → Lustre

logical object volume (LOV)

A logical entity in → Lustre

Lustre

An object-based → parallel file system.

management server (MGS)

A functional component in → Lustre

metadata server (MDS)

A functional component in → Lustre

metadata target (MDT)

A logical entity in → Lustre

MPI, MPI-2 (“message-passing interface”)

A language-independent communications protocol used to program parallel computers. Both point-to-point and collec-

tive communication are supported. MPI remains the dominant model used in high-performance computing today. There are two versions of the standard that are currently popular: version 1.2 (shortly called MPI-1), which emphasizes message passing and has a static runtime environment, and MPI-2.1 (MPI-2), which includes new features such as parallel I/O, dynamic process management and remote memory operations. MPI-2 specifies over 500 functions and provides language bindings for ANSI C, ANSI Fortran (Fortran90), and ANSI C++. Interoperability of objects defined in MPI was also added to allow for easier mixed-language message passing programming. A side effect of MPI-2 standardization (completed in 1996) was clarification of the MPI-1 standard, creating the MPI-1.2 level. MPI-2 is mostly a superset of MPI-1, although some functions have been deprecated. Thus MPI-1.2 programs still work under MPI implementations compliant with the MPI-2 standard. The MPI Forum reconvened in 2007, to clarify some MPI-2 issues and explore developments for a possible MPI-3.

MPICH2

A freely available, portable → MPI 2.0 implementation, maintained by Argonne National Laboratory

MPP (“massively parallel processing”)

So-called MPP jobs are computer programs with several parts

running on several machines in parallel, often calculating simulation problems. The communication between these parts can e.g. be realized by the → MPI software interface.

MS-MPI

Microsoft → MPI 2.0 implementation shipped with Microsoft HPC Pack 2008 SDK, based on and designed for maximum compatibility with the → MPICH2 reference implementation.

MVAPICH2

An → MPI 2.0 implementation based on → MPICH2 and developed by the Department of Computer Science and Engineering at Ohio State University. It is available under BSD licensing and supports MPI over InfiniBand, 10GigE/iWARP and RDMAoE.

NetworkDirect

A remote direct memory access (RDMA)-based network interface implemented in Windows Server 2008 and later. NetworkDirect uses a more direct path from → MPI applications to networking hardware, resulting in very fast and efficient networking. See the main article “Windows HPC Server 2008 R2” for further details.

NFS (Network File System)

A network file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to

access files over a network in a manner similar to how local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The Network File System is an open standard defined in RFCs, allowing anyone to implement the protocol.

Sun used version 1 only for in-house experimental purposes. When the development team added substantial changes to NFS version 1 and released it outside of Sun, they decided to release the new version as V2, so that version interoperability and RPC version fallback could be tested. Version 2 of the protocol (defined in RFC 1094, March 1989) originally operated entirely over UDP. Its designers meant to keep the protocol stateless, with locking (for example) implemented outside of the core protocol. Version 3 (RFC 1813, June 1995) added:

- support for 64-bit file sizes and offsets, to handle files larger than 2 gigabytes (GB)
- support for asynchronous writes on the server, to improve write performance
- additional file attributes in many replies, to avoid the need to re-fetch them
- a REaddirPLUS operation, to get file handles and attributes along with file names when scanning a directory
- assorted other improvements

At the time of introduction of Version 3, vendor support for TCP as a transport-layer protocol began increasing. While several vendors had already added support for NFS Version 2 with TCP as a transport, Sun Microsystems added support for TCP as a transport for NFS at the same time it added support for Version 3. Using TCP as a transport made using NFS over a WAN more feasible.

Version 4 (RFC 3010, December 2000; revised in RFC 3530, April 2003), influenced by AFS and CIFS, includes performance improvements, mandates strong security, and introduces a stateful protocol. Version 4 became the first version developed with the Internet Engineering Task Force (IETF) after Sun Microsystems handed over the development of the NFS protocols.

NFS version 4 minor version 1 (NFSv 4.1) has been approved by the IESG and received an RFC number since Jan 2010. The NFSv 4.1 specification aims: to provide protocol support to take advantage of clustered server deployments including the ability to provide scalable parallel access to files distributed among multiple servers. NFSv 4.1 adds the parallel NFS (pNFS) capability, which enables data access parallelism. The NFSv 4.1 protocol defines a method of separating the filesystem meta-data from the location of the file data; it goes beyond the simple name/data separation by strip-

ing the data amongst a set of data servers. This is different from the traditional NFS server which holds the names of files and their data under the single umbrella of the server.

In addition to pNFS, NFSv 4.1 provides sessions, directory delegation and notifications, multi-server namespace, access control lists (ACL/SACL/DACL), retention attributions, and SECINFO_NO_NAME. See the main article “Parallel Filesystems” for further details.

NUMA (“non-uniform memory access”)

A computer memory design used in multiprocessors, where the memory access time depends on the memory location relative to a processor. Under NUMA, a processor can access its own local memory faster than non-local memory, that is, memory local to another processor or memory shared between processors.

object storage server (OSS)

A functional component in → Lustre

object storage target (OST)

A logical entity in → Lustre

object-based storage device (OSD)

An intelligent evolution of disk drives that can store and serve objects rather than simply place data on tracks and sectors.

This task is accomplished by moving low-level storage functions into the storage device and accessing the device through an object interface. Unlike a traditional block-oriented device providing access to data organized as an array of unrelated blocks, an object store allows access to data by means of storage objects. A storage object is a virtual entity that groups data together that has been determined by the user to be logically related. Space for a storage object is allocated internally by the OSD itself instead of by a host-based file system. OSDs manage all necessary low-level storage, space management, and security functions. Because there is no host-based metadata for an object (such as inode information), the only way for an application to retrieve an object is by using its object identifier (OID). The SCSI interface was modified and extended by the OSD Technical Work Group of the Storage Networking Industry Association (SNIA) with varied industry and academic contributors, resulting in a draft standard to T10 in 2004. This standard was ratified in September 2004 and became the ANSI T10 SCSI OSD V1 command set, released as INCITS 400-2004. The SNIA group continues to work on further extensions to the interface, such as the ANSI T10 SCSI OSD V2 command set.

OpenCL (“Open Computing Language”)

A framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, and other processors. OpenCL includes a language (based on C99) for

writing kernels (functions that execute on OpenCL devices), plus APIs that are used to define and then control the platforms. OpenCL provides parallel computing using task-based and data-based parallelism. OpenCL is analogous to the open industry standards OpenGL and OpenAL, for 3D graphics and computer audio, respectively. Originally developed by Apple Inc., which holds trademark rights, OpenCL is now managed by the non-profit technology consortium Khronos Group.

OpenMP (“Open Multi-Processing”)

An application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++ and Fortran on many architectures, including Unix and Microsoft Windows platforms. It consists of a set of compiler directives, library routines, and environment variables that influence run-time behavior.

Jointly defined by a group of major computer hardware and software vendors, OpenMP is a portable, scalable model that gives programmers a simple and flexible interface for developing parallel applications for platforms ranging from the desktop to the supercomputer.

An application built with the hybrid model of parallel programming can run on a computer cluster using both OpenMP and Message Passing Interface (MPI), or more transparently through the use of OpenMP extensions for non-shared memory systems.

OpenMPI

An open source → MPI-2 implementation that is developed and maintained by a consortium of academic, research, and industry partners.

A distributed filesystem like → Lustre or → parallel NFS, where a single storage namespace is spread over several storage devices across the network, and the data is accessed in a striped way over the storage access paths in order to increase performance. See the main article “Parallel Filesystems” for further details.

parallel NFS (pNFS)

A → parallel file system standard, optional part of the current → NFS standard 4.1. See the main article “Parallel Filesystems” for further details.

PCI Express (PCIe)

A computer expansion card standard designed to replace the older PCI, PCI-X, and AGP standards. Introduced by Intel in 2004, PCIe (or PCI-E, as it is commonly called) is the latest standard for expansion cards that is available on mainstream computers. PCIe, unlike previous PC expansion standards, is structured around point-to-point serial links, a pair of which (one in each direction) make up lanes; rather than a shared parallel bus. These lanes are routed by a hub on the main-board acting as a crossbar switch. This dynamic point-to-point behavior allows

more than one pair of devices to communicate with each other at the same time. In contrast, older PC interfaces had all devices permanently wired to the same bus; therefore, only one device could send information at a time. This format also allows “channel grouping”, where multiple lanes are bonded to a single device pair in order to provide higher bandwidth. The number of lanes is “negotiated” during power-up or explicitly during operation. By making the lane count flexible a single standard can provide for the needs of high-bandwidth cards (e.g. graphics cards, 10 Gigabit Ethernet cards and multiport Gigabit Ethernet cards) while also being economical for less demanding cards.

Unlike preceding PC expansion interface standards, PCIe is a network of point-to-point connections. This removes the need for “arbitrating” the bus or waiting for the bus to be free and allows for full duplex communications. This means that while

	PCIe 1.x	PCIe 2.x	PCIe 3.0
x1	256 MB/s	512 MB/s	1 GB/s
x2	512 MB/s	1 GB/s	2 GB/s
x4	1 GB/s	2 GB/s	4 GB/s
x8	2 GB/s	4 GB/s	8 GB/s
x16	4 GB/s	8 GB/s	16 GB/s
x32	8 GB/s	16 GB/s	32 GB/s

standard PCI-X (133 MHz 64 bit) and PCIe x4 have roughly the same data transfer rate, PCIe x4 will give better performance if multiple device pairs are communicating simultaneously or if communication within a single device pair is bidirectional. Specifications of the format are maintained and developed by a group of more than 900 industry-leading companies called the PCI-SIG (PCI Special Interest Group). In PCIe 1.x, each lane carries approximately 250 MB/s. PCIe 2.0, released in late 2007, adds a Gen2-signalling mode, doubling the rate to about 500 MB/s. PCIe 3.0, currently in development (for release around 2010), will add a Gen3-signalling mode, at 1 GB/s.

process
→ thread

PTX (“parallel thread execution”)

Parallel Thread Execution (PTX) is a pseudo-assembly language used in NVIDIA’s CUDA programming environment. The ‘nvcc’ compiler translates code written in CUDA, a C-like language, into PTX, and the graphics driver contains a compiler which translates the PTX into something which can be run on the processing cores.

RDMA (“remote direct memory access”)

Allows data to move directly from the memory of one computer

into that of another without involving either one’s operating system. This permits high-throughput, low-latency networking, which is especially useful in massively parallel computer clusters. RDMA relies on a special philosophy in using DMA. RDMA supports zero-copy networking by enabling the network adapter to transfer data directly to or from application memory, eliminating the need to copy data between application memory and the data buffers in the operating system. Such transfers require no work to be done by CPUs, caches, or context switches, and transfers continue in parallel with other system operations. When an application performs an RDMA Read or Write request, the application data is delivered directly to the network, reducing latency and enabling fast message transfer. Common RDMA implementations include → InfiniBand, → iSER, and → iWARP.

RISC (“reduced instruction-set computer”)

A CPU design strategy emphasizing the insight that simplified instructions that “do less“ may still provide for higher performance if this simplicity can be utilized to make instructions execute very quickly → CISC.

ScaLAPACK (“scalable LAPACK”)

Library including a subset of → LAPACK routines redesigned for distributed memory MIMD (multiple instruction, multiple data) parallel computers. It is currently written in a Single-Program-Multiple-Data style using explicit message passing

for interprocessor communication. ScaLAPACK is designed for heterogeneous computing and is portable on any computer that supports → MPI. The fundamental building blocks of the ScaLAPACK library are distributed memory versions (PBLAS) of the Level 1, 2 and 3 → BLAS, and a set of Basic Linear Algebra Communication Subprograms (BLACS) for communication tasks that arise frequently in parallel linear algebra computations. In the ScaLAPACK routines, all interprocessor communication occurs within the PBLAS and the BLACS. One of the design goals of ScaLAPACK was to have the ScaLAPACK routines resemble their → LAPACK equivalents as much as possible.

service-oriented architecture (SOA)

An approach to building distributed, loosely coupled applications in which functions are separated into distinct services that can be distributed over a network, combined, and reused. See the main article “Windows HPC Server 2008 R2” for further details.

single precision / double precision

→ floating point standard

SMP (“shared memory processing”)

So-called SMP jobs are computer programs with several parts running on the same system and accessing a shared memory region. A usual implementation of SMP jobs is →

multi-threaded programs. The communication between the single threads can e.g. be realized by the → OpenMP software interface standard, but also in a non-standard way by means of native UNIX interprocess communication mechanisms.

SMP (“symmetric multiprocessing”)

A multiprocessor or multicore computer architecture where two or more identical processors or cores can connect to a single shared main memory in a completely symmetric way, i.e. each part of the main memory has the same distance to each of the cores. Opposite: → NUMA

storage access protocol

Part of the → parallel NFS standard

STREAM

A simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels.

streaming multiprocessor (SM)

Hardware component within the → Tesla GPU series

superscalar processors

A superscalar CPU architecture implements a form of parallelism called instruction-level parallelism within a single proces-

sor. It thereby allows faster CPU throughput than would otherwise be possible at the same clock rate. A superscalar processor executes more than one instruction during a clock cycle by simultaneously dispatching multiple instructions to redundant functional units on the processor. Each functional unit is not a separate CPU core but an execution resource within a single CPU such as an arithmetic logic unit, a bit shifter, or a multiplier.

Tesla

NVIDIA's third brand of GPUs, based on high-end GPUs from the G80 and on. Tesla is NVIDIA's first dedicated General Purpose GPU. Because of the very high computational power (measured in floating point operations per second or FLOPS) compared to recent microprocessors, the Tesla products are intended for the HPC market. The primary function of Tesla products are to aid in simulations, large scale calculations (especially floating-point calculations), and image generation for professional and scientific fields, with the use of → CUDA. See the main article "NVIDIA GPU Computing" for further details.

thread

A thread of execution is a fork of a computer program into two or more concurrently running tasks. The implementation of threads and processes differs from one operating system

to another, but in most cases, a thread is contained inside a process. On a single processor, multithreading generally occurs by multitasking: the processor switches between different threads. On a multiprocessor or multi-core system, the threads or tasks will generally run at the same time, with each processor or core running a particular thread or task. Threads are distinguished from processes in that processes are typically independent, while threads exist as subsets of a process. Whereas processes have separate address spaces, threads share their address space, which makes inter-thread communication much easier than classical inter-process communication (IPC).

thread (in CUDA architecture)

Part of the → CUDA programming model

thread block (in CUDA architecture)

Part of the → CUDA programming model

thread processor array (TPA)

Hardware component within the → Tesla GPU series

10 Gigabit Ethernet

The fastest of the Ethernet standards, first published in 2002 as IEEE Std 802.3ae-2002. It defines a version of Ethernet with

a nominal data rate of 10 Gbit/s, ten times as fast as Gigabit Ethernet. Over the years several 802.3 standards relating to 10GbE have been published, which later were consolidated into the IEEE 802.3-2005 standard. IEEE 802.3-2005 and the other amendments have been consolidated into IEEE Std 802.3-2008. 10 Gigabit Ethernet supports only full duplex links which can be connected by switches. Half Duplex operation and CSMA/CD (carrier sense multiple access with collision detect) are not supported in 10GbE. The 10 Gigabit Ethernet standard encompasses a number of different physical layer (PHY) standards. As of 2008 10 Gigabit Ethernet is still an emerging technology with only 1 million ports shipped in 2007, and it remains to be seen which of the PHYs will gain widespread commercial acceptance.

warp (in CUDA architecture)

Part of the → CUDA programming model



transtec Germany

Tel +49 (0) 7071/703-400
transtec@transtec.de
www.transtec.de

transtec Switzerland

Tel +41 (0) 44/818 47 00
transtec.ch@transtec.ch
www.transtec.ch

transtec Austria

Tel +43 (0) 1/726 60 90 11
transtec.at@transtec.at
www.transtec.at

transtec United Kingdom

Tel +44 (0) 1295/756 100
transtec.uk@transtec.co.uk
www.transtec.co.uk

ttec Netherlands

Tel +31 (0) 24 34 34 210
ttec@ttec.nl
www.ttec.nl

transtec France

Tel +33 (0) 3.88.55.16.00
transtec.fr@transtec.fr
www.transtec.fr

Texts and conception:

Layout:

Dr. Oliver Tennert, Director Marketing & HPC Solutions | Oliver.Tennert@transtec.de

Karolin Kraut, Senior Graphics & Design / Art Direction | Karolin.Kraut@transtec.de

© transtec AG, April 2010

The graphics, diagrams and tables found herein are the intellectual property of transtec AG and may be reproduced or published only with its express permission. No responsibility will be assumed for inaccuracies or omissions. Other names or logos may be trademarks of their respective owners.