HPTC in the 21st Century - Promises and Challenges of Grid and Cluster Computing
Agenda

• Computing Trends

• Clusters and Grids
  – Clusters
    • Cluster Design Parameter
    • Cluster Compute Nodes
    • Provisioning
    • Job scheduling
  – Grids
    • Design Parameter
    • Service Oriented Architectures and new Grid applications

• Bringing it all together
Trends in computing
Computing Trends cont.
Larger servers are nice, but...

- **Vertical scaling**
  - OpenMP/ thread parallel applications
  - Instruction level parallelism
  - Large Memory
  - OS schedules threads
  - Top performance with low latency between CPUs
  - Top Price

- **Horizontal scaling**
  - Throughput / serial applications
  - MPI parallelism
  - Interconnect impacts performance significantly
  - Price/CPU low
...but horizontal scaling comes at a price, too.

- More nodes to administer
- More nodes to connect to Network(s)
- More nodes to connect to Data
- More complexity to provision applications, arrays
- More effort to achieve parallel execution
- Space/Power/Cooling
Trend conclusions

- Clusters are now established as MPP, SMP and Constellations established before.
- Cost per flop/s and intop/s are motivation.
- Clusters create new challenges for administration, interconnects and programming (many of them unsolved).
Clusters and Grids (def)

- **Cluster**
  - Collection of multiple, networked compute entities (nodes) used as a singular resource, housed in physical proximity.

- **Grid**
  - Collection of multiple, networked, autonomous entities (nodes or clusters of computers, instruments or data repositories) used as a singular resource, housed in one or more locations.
Cluster vs. Grid characteristics

- **Cluster**
  - Network Latency between nodes
    - Ethernet, Myrinet, Infiniband
  - Access
    - From within Firewall
    - Unix User-ID
  - File System
    - Performance
    - Scalability to n nodes

- **Grid**
  - Network Bandwidth
    - 10GE, OCx
  - Access
    - across Firewalls
    - Authentication, Authorization
    - “Federations”
  - File System
    - Global File System
    - Ontologies
Cluster and Grid Layers

- Clustergrid
- Campusgrid
- Global Grid
Cluster Design Parameters

- Space/Power/Cooling
- Interconnects
- File-Systems
- SW-Provisioning
- Job Scheduling
## Power consumption and Performance

<table>
<thead>
<tr>
<th></th>
<th>Opteron 140EE</th>
<th>Opteron 146HE</th>
<th>Opteron 250</th>
<th>Xeon 3.6 GHz</th>
<th>EM64T</th>
<th>Itanium 2 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>30 watts</td>
<td>55 watts</td>
<td>89 watts</td>
<td>105 watts</td>
<td>110 watts</td>
<td>140 watts</td>
</tr>
<tr>
<td><strong>Memory Controller</strong></td>
<td>0 watts (integrated)</td>
<td>0 watts (integrated)</td>
<td>0 watts (integrated)</td>
<td>3 watts</td>
<td>3 watts</td>
<td>40 watts</td>
</tr>
<tr>
<td><strong>Power SubTot</strong></td>
<td>30 watts</td>
<td>55 watts</td>
<td>89 watts</td>
<td>108 watts</td>
<td>113 watts</td>
<td>180 watts</td>
</tr>
<tr>
<td><strong>1x2P/year</strong></td>
<td>$105</td>
<td>$193</td>
<td>$312</td>
<td>$378</td>
<td>$396</td>
<td>$631</td>
</tr>
<tr>
<td><strong>500 2P Annual</strong></td>
<td>$52,560</td>
<td>$96,360</td>
<td>$155,928</td>
<td>$189,216</td>
<td>$197,976</td>
<td>$315,360</td>
</tr>
<tr>
<td><strong>Energy Cost</strong></td>
<td>Save 66%</td>
<td>Save 38%</td>
<td>BASELINE</td>
<td>21% More</td>
<td>27% More</td>
<td>102% More</td>
</tr>
<tr>
<td></td>
<td>$103,368</td>
<td>$59,568</td>
<td>BASELINE</td>
<td>$33,288</td>
<td>$42,048</td>
<td>$159,432</td>
</tr>
<tr>
<td><strong>CFP2000</strong></td>
<td>1075/1072</td>
<td>1394/1306</td>
<td>1787/1637</td>
<td>1538/1462</td>
<td>1502/1507</td>
<td>2043/2038</td>
</tr>
<tr>
<td></td>
<td>36.3% less</td>
<td>17.5% less</td>
<td>BASELINE</td>
<td>14% less</td>
<td>16% less</td>
<td>21% more</td>
</tr>
<tr>
<td><strong>CINT2000</strong></td>
<td>990/948</td>
<td>1354/1289</td>
<td>1584/1437</td>
<td>1503/1502</td>
<td>1484/1419</td>
<td>1232/1200</td>
</tr>
<tr>
<td></td>
<td>41.3% less</td>
<td>20% less</td>
<td>BASELINE</td>
<td>5% less</td>
<td>7/-1% less</td>
<td>27% less</td>
</tr>
</tbody>
</table>

Energy includes power input & cooling. Power Utility cost: $0.10/KW-hr. Publicly available processor specifications.
Power consumption, cont.

- 1024 CPUs can consume between 30kw (Opteron EE) and 184kw (Itanium)

- 512 dual nodes consume 183kw plus 626kBTU/h and yield ~4TF
  - 16 racks, 11 terminal concentrators, 6 KVM
  - Diskless 3.6 kw less (~7 w per node)

- Dual core CPUs will increase Power- and Space density, efficiency
  - Opteron dual core uses same power as single core
  - Power5 from IBM and SPARC IV from Sun are already there
Multicore CPU's

AMD's dual core Toledo to sample first half 2005

By Mike Magee: Friday 16 July 2004, 10:50
retrieved from http://www.theinquirer.net/?article=17262

Intel president Paul Otellini yesterday opened the autumn Intel Developer Forum (IDF) with an upbeat assessment of the IT market, detailing the company's plans to bring dual-core processing to the mass market.
Future of Cluster and CMT

This patent-pending technology, initially targeted at Java and J2EE platform-based applications, transparently redirects application workload to the compute pool. No changes are required to applications, or the existing infrastructure configuration. The Azul technology works with J2EE platform products including BEA® WebLogic® and IBM® WebSphere® application servers. Each appliance has up to 384 coherent processor cores and 256 gigabytes of memory packed in a purpose-built design that delivers the benefits of symmetric multiprocessing with tremendous economic benefits.

- Multicore Implementations
- “Clusters in a box”
- Multicore and Multi-CPU

Orion DS-96 Deskside Cluster Workstation
* 48-96 node cluster
* 300 Gf (peak) / 150 Gf (sust)
* 7.8Tb of internal disk storage
Cluster Interconnects

- **Ethernet (100/1000)**
  - Virtually free
  - Latency issue (okay for serial, throughput)
  - Bandwidth limited compared to Myrinet, IB
  - Limited scalability

- **Ethernet (10GE)**
  - Can be as low as 10us
  - Bandwidth comparable to Infiniband
  - Expensive (4x NIC over IB HCA)

- **Myrinet/Infiniband**
  - ~10 us latency
  - Significant cost, i.e. $1,500 per $3,000 node
  - Calculate Break even between scalability of code, cost per node and cost of network
  - IB can be used for remote boot, SCSI over IB
Interconnect Impact on cost and performance

- Linpack efficiency $R_{\text{max}}/R_{\text{peak}}$, source TOP500.org
  - TOP500, #60: xSeries Xeon 3.06 GHz – Gig-E, $2403/3916=61\%$
  - TOP500, #58: Pentium4 Xeon 3.06 GHz, Myrinet, $2455/3672=67\%$
  - TOP500, #10: Opteron 2.2 GHz, Myrinet, $8061/11264=71\%$
  - TOP500, #277: Opteron 2.0 GHz, Infiniband, $922/1152=80\%$
Parallel file system options for Clusters

- **Distributed File System** (e.g. NFS/CIFS)
  - Server is bottleneck

- **Symmetric Clustered File System** (e.g. GPFS)
  - Lock management is bottleneck

- **SAN based File Systems** (e.g. SANergy)
  - Server is bottleneck
  - Scale limited

- **Parallel File System** (e.g. PFS)
  - Asymmetric
  - MD Server is bottleneck
Example of an InfiniBand stack

High Performance Block Storage Traffic

Low Latency MPI Applications

High Performance File Storage Traffic, Any Sockets Based Apps

Upper Layer Protocols

InfiniBand Services

Clustering Middleware

HCA Drivers

Access

Data

SCSI RDMA (SRP)
NFS RDMA (RPC RDMA)
iSCSI RDMA (iSER)
MPI
SDP
TCP
uDAPL
kDAPL
Score / LAM
IPoIB

Low Latency Clustering Apps, Clustered File Systems

UDP Based Apps, Management Traffic

High Performance File Storage Traffic, Any Sockets Based Apps
File Systems (1)

- Data over IB (NFS, SCSI or iSCSI)
  - in combination w/ diskless or diskful servers
  - Can bridge to FC/AL SAN
  - Works with fast, reliable File Systems like ZFS or QFS instead of distributed disks, which allows striping (performance) and replication (availability)

- NFS
  - Standardized and part of many OS
  - Works well in Filers, DAS and NAS
  - New RDMA feature will lower CPU utilization and improve performance
  - Not a real “parallel” File System, certainly not for multiple clusters
  - Synchronous procedure calls limit scalability
  - DAFS in combination with NFS-v4 could become interesting
What is NFS/RDMA?

- **Goals:**
  - Eliminate data copies
  - Reduce CPU and NFS overhead (OS bypass)
  - Reduce Latency

- **Protocol components**
  - RDMA fabric
    - eg. iWARP and/or IB w/ kDAPL
  - RPC RDMA
  - NFS V3/V4/V4.1 w/ sessions
File Systems (2)

- **Lustre**
  - Object oriented
  - Scales well in larger clusters
  - Transparent to UFS
  - Open source; http://www.lustre.org/

- **GPFS for Linux**
  - Data striping to increase performance
  - True parallel file system
  - Lock Manager bottleneck

- **SAM-FS**
  - Transparent multi-tiered Archiving
  - Linux clients early '05
SW-Provisioning

- **PXE-boot and diskless servers**
  - One or more boot-images are linked at boot time
  - Very flexible
  - Local disk can be substituted by SCSI-over-IB

- **ROCKS**
  - Easy to use, ready made “rolls”
  - Limitations to OS, Software stack
  - Monitoring with ganglia, not integrated
  - www.rocksclusters.org

- **XCAT**
  - Versatile, open source
  - Includes many monitoring functions
Job Scheduling

• Use Resources of a cluster more efficiently
  – Resources can be CPU, memory, SW-License
  – Detect idle resource and schedule to appropriate resource

• Features to look for
  – Granularity (job run time) & Throughput (jobs per minute or h)
  – Policy based scheduling (share-tree, urgency, override)
  – Reservation w/ backfilling
  – Reporting (i.e. License utilization, Queue Times, Use per group, project or user)

• Standardization under way
  – Necessary to optimize applications (API, wrappers, scripts)
  – e.g. DRMAA part of Globus, supported by Intel, Altair (PBSPro), IBM, United Devices, Entropia, GridIron, Condor
Example: Sun™ Cluster Grid Model Solution

- Maximize resources for single projects, teams, departments
- Prioritize jobs
- Manage jobs from start to finish
Solution: N1GE6 Policies and Monitoring

- Owners negotiate policies
- Automated tools enforce policies
- Exceptions for specific needs/events provide flexibility
- Monitoring ensures policies are enforced
Job Resource Reservation

Jobs, with resource requirements, ordered by priority
length represents duration of requirement
### Grid Resources Representation

<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>Host 2</th>
<th>Host 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>CPU</td>
<td>CPU</td>
</tr>
</tbody>
</table>

- **Time**

```markdown
<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>Host 2</th>
<th>Host 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>CPU</td>
<td>CPU</td>
</tr>
</tbody>
</table>
```
Simple, priority-based scheduling

Goes last!

Global

Host 2

Job 3

Wasted resources

CPU

Mem.

Job 3

Job 3

Job 2

Job 2

Host 1

CPU

Mem.

Job 1

Job 5

Job 2

Job 2

Job 4

Job 6

Job 2

Job 2

Job 4

Job 6

Job 2

Job 2

Job 1

Job 5

Job 2

Job 2

CPU

Mem.

Time

Lic.
Scheduling with Resource Reservation

- **Global**: Job 2, Job 4, Job 6
- **Host 2**: Job 2, Job 3, Job 3
- **Host 1**: Job 2, Job 2, Job 4, Job 6

Time

CPU

Mem.
Resource Reservation with backfilling

<table>
<thead>
<tr>
<th>Global</th>
<th>Lic.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Job 6 Job 2 Job 4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host 2</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Job 6 Job 2 Job 3</td>
</tr>
<tr>
<td></td>
<td>Job 6 Job 2 Job 3</td>
</tr>
<tr>
<td></td>
<td>Job 2</td>
</tr>
<tr>
<td></td>
<td>Job 6 Job 2 Job 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host 1</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Job 5 Job 2</td>
</tr>
<tr>
<td></td>
<td>Job 1 Job 2</td>
</tr>
<tr>
<td></td>
<td>Job 1 Job 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 6</td>
</tr>
<tr>
<td>Job 2</td>
</tr>
<tr>
<td>Job 2</td>
</tr>
<tr>
<td>Job 5</td>
</tr>
<tr>
<td>Job 2</td>
</tr>
<tr>
<td>Job 4</td>
</tr>
</tbody>
</table>

Time
Cluster node selector

Existing application?

Scaling serial, MPI or OpenMP?

More nodes or faster nodes?

I/O to compute ratio?

Small (<2GB)
- Serial jobs, limited scalability
- Standard Network

Medium (2-16 GB)
- Moderate I/O intensity
- Benefits from larger memory
- Low latency Network (LLN)

Large (>16GB)
- OpenMP/MPI Hybrids
- Memory & CPU intensity
- Backplane & LLN

Choose appropriate development tools

- a. lower clocked, high density nodes for “light weight” threads
- b. fast CPUs for ILP, “heavy threads”

- a. Choose network interconnect
- b. Reduce interconnects with multi CPU nodes
- c. Optimal memory per node
Grid design parameters

• Access to grid resources
  – Within campus, LDAP becomes critical foundation
  – If users are dispersed, identity management becomes an issue
    • Certificate based authentication (OGSA)
    • Shibboleth (Internet2)
    • Federations

• Offer variety of resources
  – Sharing unique resources allows optimized use

• Negotiate usage policies
  – Who can use what, when, and for what compensation?

• Provide Network Bandwidth
  – Internet2, Teragrid, NLBR, Canet4, regional Grid initiatives

• Enable SOA for new breed of grid aware and optimized applications
Bandwidth is being addressed....

National LambdaRail Architecture

For more information regarding NLR see http://www.nlr.net or contact info@nlr.net
Beyond resource sharing....

Untapped resources are available for everyone.
Putting it all together

- Different clusters, designed with different parameters, can form a powerful grid
  - Run application where they fit best
- Augument with SMP, Data- and Visualization resources
- Access and Identity Management
  - Often underestimated, critical issue to address
- Platform for collaboration and exchange
SOA Definitions

• Service oriented architecture (SOA)
  – “a set of components which can be invoked, and whose interface descriptions can be published and discovered” W3C
  – “an Application Architecture that is designed to achieve loose coupling among interacting software applications. SOA provides greater flexibility in developing, integrating, and managing Enterprise Applications” Stanev
“A service in SOA is an exposed piece of functionality with three major properties:

- The interface contract to the service is platform-independent.
- The service can be dynamically located and invoked.
- The service is self-contained. That is, the service maintains its own state. “ Stanev
Orbitz SOA

Hoffmann et al., 2004
SORCER
(Texas Tech, Sobolewski)

• Service Oriented Computer Environment
  – Research in network, service, and object-centric programming:
    • Service-Oriented Programming,
    • Service-Oriented Computing Environment,
    • Service-Oriented Programming Development Tools,
    • Service-Federated Assurance and Security,
    • Self-Aware Service Federations,
    • Autonomic Service Federations, and
    • Service Federated Grids.

• Java, JINI component architecture, lookup service, discover & register

Source: http://sorcer.cs.ttu.edu/about/about.html
The computer is the service grid that exposes services to clients AWAT

Federated S2S environment to ...

- Build new services
- Convert legacy apps to dynamic SORCER services (J2EE™ technology)
- Assemble SORCER services together (RMI, Jini, Rio, JXTA, WS technologies)
- Create modern clients accessing services
1. Update combustor PCS

2. Request for nozzle validation

3. Check for nozzle insertion

4. Perform modal analysis

5. Perform CFD blow analysis
GridLab Architecture

GridLab Services

- GAS
- Viz Service
- GRMS
- Replica Catalog
- Data Movement
- Mobile Services
- Mercury

Mobile clients

Gridsphere Portal

Application Layer

ASTROPHYSICS
BIOINFORMATICS
OTHER LIFE SCIENCES

GAT Layer

GAT API

GSI-enabled Web Service API

Third Party Services and Libraries

Service Layer

C and Java APIs to Globus 2.X/3.X Pre-WS and other Core Services

Core Layer

Capability Space

User Space

GridLab Testbed and Infrastructure
Visualization as a Grid Service

- Tightly coupled simulation and visualization
- Viz output can be displayed anywhere on the grid
- In production in the Teragrid as of October 1st 2004
Traditional (Globus) Grids

• 3 sites
• 4 major systems
  – Portal access
  – SGE EE
  – Globus
  – Production Global Grid
• DAME (Rolls Royce)
White Rose Grid, Yorkshire, UK

- SGE, SGEE, Globus, Portal
- Access to compute and data resources
- Operational
Heinz J. Schwarz
Sr. Program Manager,
External Research Office
Sun Microsystems Laboratories
joerg.schwarz@sun.com