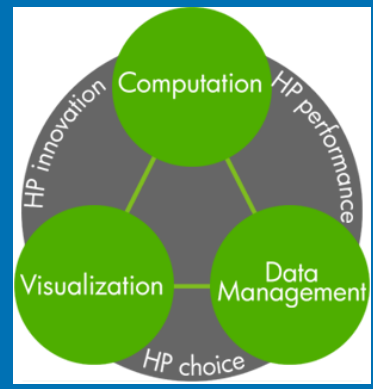
Advanced Cluster Software for Meteorology



Betty van Houten [1]
Frederic Ciesielski / Gavin Brebner / Ghislain de Jacquelot [2]
Michael Riedmann [3]
Henry Strauss [4]

[1] HPC Division Richardson
[2] HPC Competency Centre Grenoble
[3] EPC Boeblingen
[4] HPC PreSales Munich

ECMWF workshop “Use of HPC in Meteorology” -- Reading/UK, November 2nd, 2006

Outline

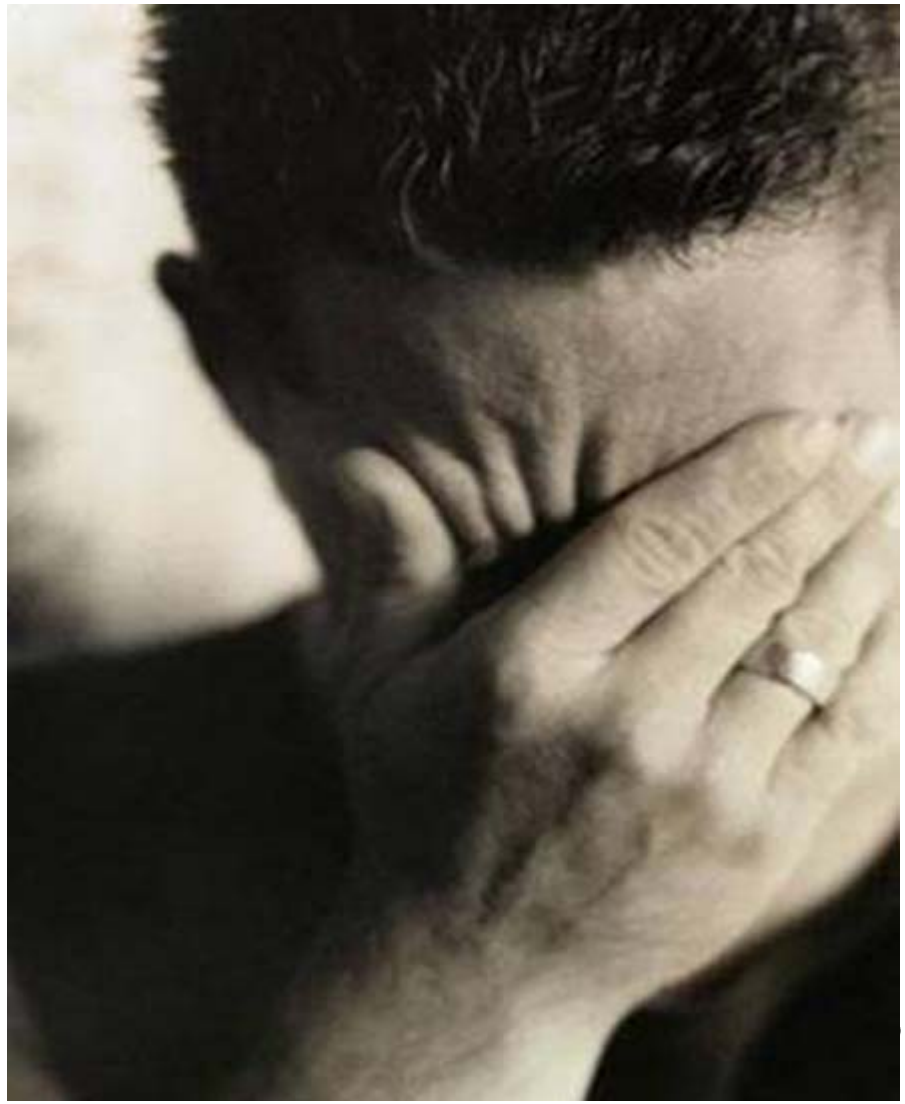
- requirements in met
- HP‘s ways to address those
  - XC – a cluster environment for real-world applications
  - HP-MPI – a universal approach
  - some thoughts on checkpoint/restart and alternatives
  - future work
- LM on HP – status & results
- conclusions

Requirements

- both capability and capacity
- reliability & turn-around times
- data management!
- visualization?

Cluster Implementation Challenges

- Manageability
- Scalability
- Integration of Data Mgmt & Visualization
- Interconnect/Network Complexity
- Version Control
- Application Availability



XC Cluster: HP's Linux-Based Production Cluster for HPC

- A production computing environment for HPC built on Linux/Industry standard clusters
  - Industrial-hardened, scalable, supported
  - Integrates leading technologies from open source and partners
- Simple and complete product for real world usage
  - Turn-key, with single system ease of deployment and management
  - Smart provisioning takes the guess work and experimentation out of deployment
  - Allows customer focus on computation rather than infrastructure
- General purpose technical computing flexibility
  - Supports throughput 'farms' as well as MPI parallel jobs
  - Plugs into 'grid' environments

XC System Software

- A complete HP-supported Linux cluster stack for operation and management of HP clusters
- An integrated and tested collection of best-in-class software technologies
  - open source where appropriate
  - easily portable across servers and interconnects
- HP-developed installation and configuration procedures and defaults, management commands, scripts and plug-ins

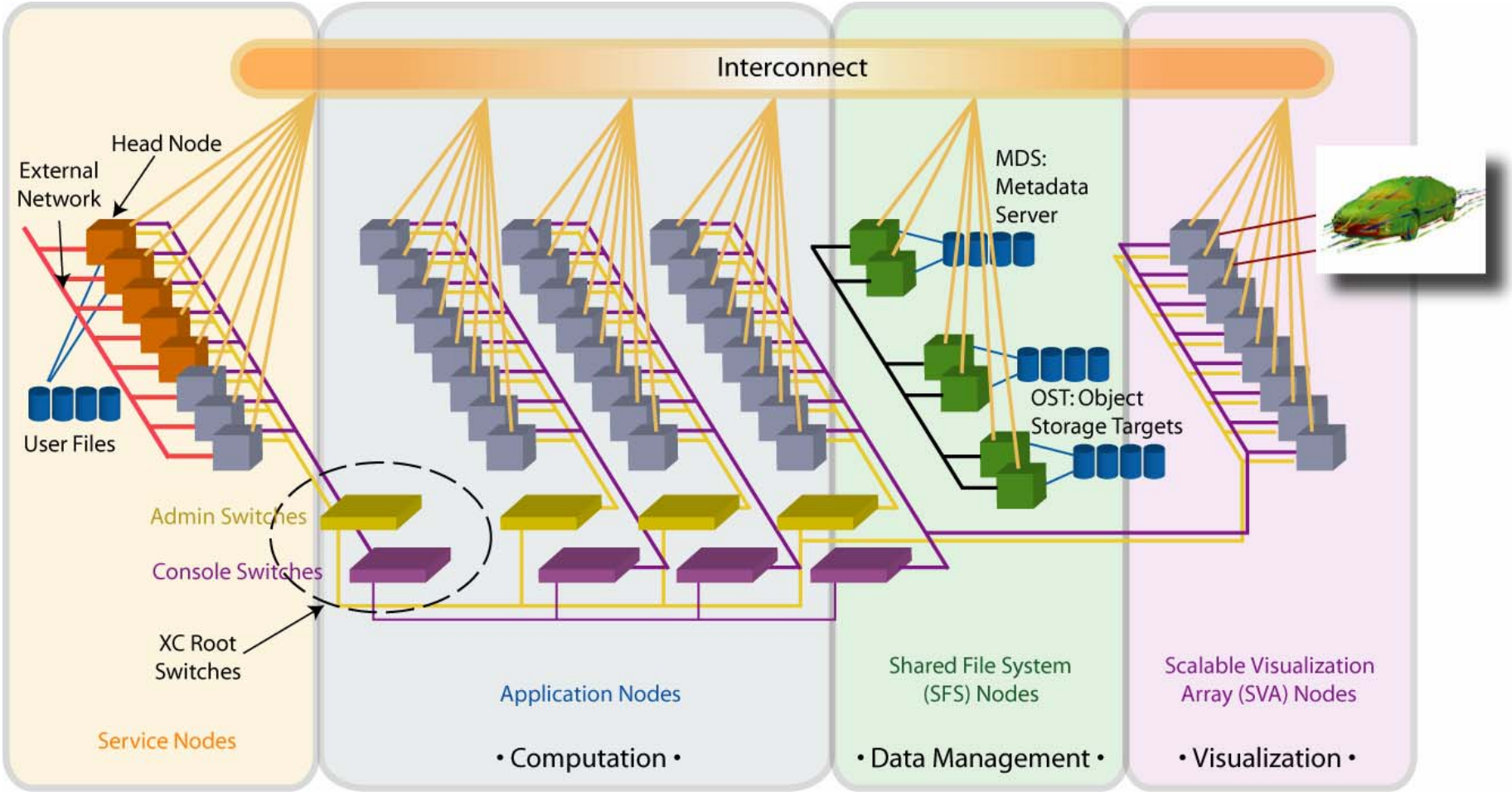
XC System Software V3.0

- Available for Xeon, Opteron and Itanium HP servers and optimized for CP
- Worldwide HP support
- Designed to be compatible with Red Hat EL 4.0 U2
- Includes:
  - Suite of system mgt and monitoring utilities
  - LSF workload mgt (choice of LSF HPC or LSF standard)
  - Interconnect libraries
  - HP MPI (MPI-2 compliant)
  - Modules for version selection of tools
  - Lustre client support
  - Preconfigured firewall
  - Preconfigured NAT router services
- LVS for single system login
- Some key availability features
- **Designed for scalability**
- Can be integrated with high performance viz
- Suite of qualified ISV apps

XC V3: Key Software Technologies

Function	Technology	Features and Benefits
Distribution and Kernel	RHEL 4	**Red Hat** Current shipping product, Posix enhancements, support for latest Opteron and Core ("Woodcrest"), ISV support
Inbound Network / Cluster Alias	LVS	**Linux Virtual Server** High availability virtual server project for managing incoming requests, with load balancing
Batch	LSF 6.x	**Platform LSF HPC** Premier scheduler, policy driven, allocation controls, MAUI support. Provides migration for AlphaserverSC customers
Resource Management	SLURM	**Simple Linux Utility for Resource Management** Fault tolerant, highly scalable, uses standard kernel
MPI	HP-MPI 2.x	**HP's Message Passing Interface** Provides standard interface for multiple interconnects, MPICH compatible, support for **MPI-2** functionality
System Files Management	SystemImager Configuration tools Cluster database	**SystemImager** Automates Linux installs, software distribution, and production deployment. Supports complete, bootable image; can use multicast; used at PNNL and Sandia
Console	HPLS_PowerD [power mgmt] Telnet based console commands	**Power control** Adaptable for HP integrated management processors – no need for terminal servers, reduced wiring **IPMI, ILO** server interfaces to low level console controls **CMF** content management framework
Monitoring	Nagios SuperMON	**Nagios** Browser based, robust host, service and network monitor from open source. **SuperMon** supports high speed, high sample rates, low perturbation monitoring for clusters.
High Perf I/O	Lustre 1.2.x	**Lustre Parallel File System** High performance parallel file system – efficient, robust, scalable

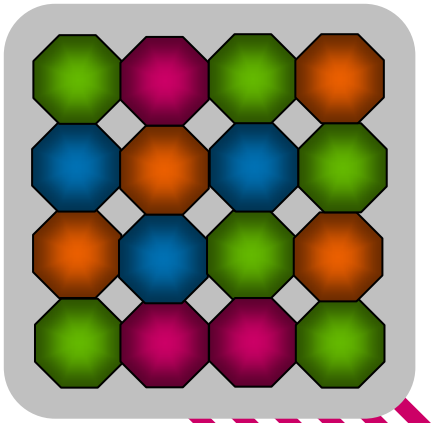
XC System Architecture



HP StorageWorks Scalable File Share

Scalable High-Bandwidth Storage Appliance for Linux Clusters

Higher Cluster Throughput:
Solves the I/O Bottleneck for Linux Clusters



A Cluster of Data Servers Forming a Single Virtual Scalable File Server

Scalable Bandwidth

Linux Cluster

- **Scalable bandwidth**
  - 200 MB/s to 35 GB/s (more by request)
  - Excellent price/performance
- **Scalable capacity**
  - 2 TB to 1024 TB (more by request)
- **Scalable connectivity**
  - Tens to thousands of compute clients
- **Scalable resiliency**
  - No single points of failure
  - Flexible resiliency choices
- **Scalable simplicity**
  - Easy-to-use standard, simple-to-use shared file system

HP-MPI – a universal approach

- **The Universal MPI for Linux, HP-UX, Tru64 and MS Windows**
  - Same interface and run-time environment across all platforms
    - Same launch, header files and API for different OSs
  - Supports HP 9000 servers, HP Integrity servers, and HP ProLiant servers
  - Transparent support for multiple interconnects
    - TCP/IP; InfiniBand (all vendors); Myrinet; Quadrics, Level 5 Networks
  - Enables single executable for each OS (HP-UX, Linux, Tru64, Windows)
  - HP-UX version used by major ISVs for over a decade; now also a leader in Linux

- **Greatly reduces the complexity to develop and deploy MPI applications**
  - HP-MPI is the only MPI you need to support a wide choice of industry-standard processors, interconnects, and OSs
  - single qualification for each platform (CPU/OS), independent of interconnect

- **MPICH Object Compatibility**
  - HP-MPI is object compatible with MPICH V1.2.5 and later
  - build as shared library, dynamically link

HP-MPI Value Propositions

Value Propositions	ISV & End User Benefits
Portability	Application independence from switch, OS, CPU type ...
Robustness	Bulletproof run time execution; backward compatibility
Performance	Parity or performance gain over alternatives
Support by HP	Superior support to public domain or other commercial MPI libraries
Applications	Broad adoption ensures application availability on widest choice of platforms

HP-MPI Portability for developers

- Debug and Profiling Tools
  - Supports TotalView and Intel Trace Collector/Analyzer (Vampir)
  - Unique built-in diagnostic library
  - Advanced run-time error checking and analysis
    - Message signature analysis to detect type mismatches in MPI calls
    - MPI object-space corruption detection
    - Multiple buffer writes detect whether the data type specified in a receive or gather operation causes MPI to write to a user buffer more than once
- Full MPI-2 functionality
  - Dynamic processes
  - One-sided communications
  - Extended collectives
  - Thread safe
  - Current ROMIO (a portable implementation of MPI I/O)

HP-MPI Robustness for production deployment

- HP-MPI supports important reliability features
  - resource cleanup
    - HP-MPI implements a mechanism to automatically clean-up resource when your MPI job exits (successfully or not)
  - signal propagation
    - signals sent to mpirun are propagated to all ranks
  - stdio processing
    - HP-MPI provides optional stdio processing features
      - stdin can be targeted to a particular process, or can be broadcast to every process.
      - stdout processing includes buffer control, pre-pending MPI rank numbers, and combining repeated output

HP-MPI Performance for scalability

- Performance optimization
  - shared memory optimization for intra-host communication
    - take advantage of shared memory for lower latency and higher bandwidth
  - native implementations on high performance interconnects
    - low level APIs are used to take full advantage of high performance interconnects
      - Quadrics Elan3/4
      - InfiniBand VAPI, uDAPL, OpenFabrics, IT-API for HP-UX
      - Myrinet GM/GM2, MX
  - collective routines are optimized

HP-MPI V2.2.5 10/06

- Enhanced usability of interconnect environment variables
- MPI-2 name publishing functions
- IB partitioning
- QLogic's InfiniPath™ support
- Myrinet MX support
- Expansion of Signal Propagation
- New mpirun option for intra-host performance tuning
- Fast One-sided lock/unlock under InfiniBand VAPI
- OpenFabric support

www.hp.com/go/mpi

Challenge

Several times a day, stop any job running on my large cluster, run a large but short simulation on all the CPUs, and then resume everything

- Common problem in weather forecasting

Possible answers

- checkpointing?
  - not at o/s level! (might take forever on very large clusters, not supported by Linux)
- Fixed reservations
  - But can't always rely on jobs finishing on time.
- Dedicated resources
  - Wasteful
- Pre-emption
  - kill existing jobs ?
  - suspend existing jobs ?
    - How to restart ?
    - How to free resources used by suspended jobs ?

- preemption could be better and faster
  - supported by LSF
  - requires nevertheless signals to be forwarded to all processes
    - feature implemented in HP MPI 2.2+
  - implies overcommitment

Proposed solution, using LSF + advanced customization: 'suspend-restart'

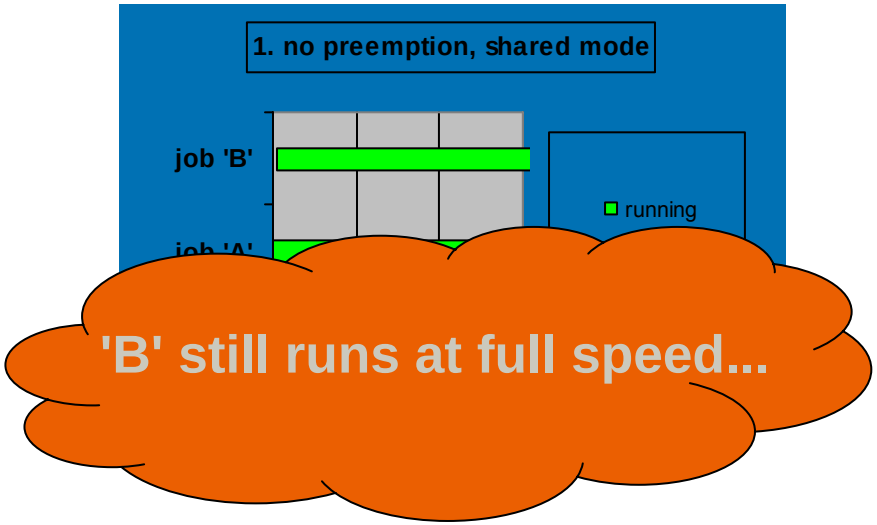
- preempted jobs are suspended
  - HP MPI and sequential jobs will stop properly
- preempted processes are swapped out to disk
  - memory is released
- preemptive job starts and completes
- preempted processes are swapped in
- preempted jobs are resumed and run again at full speed

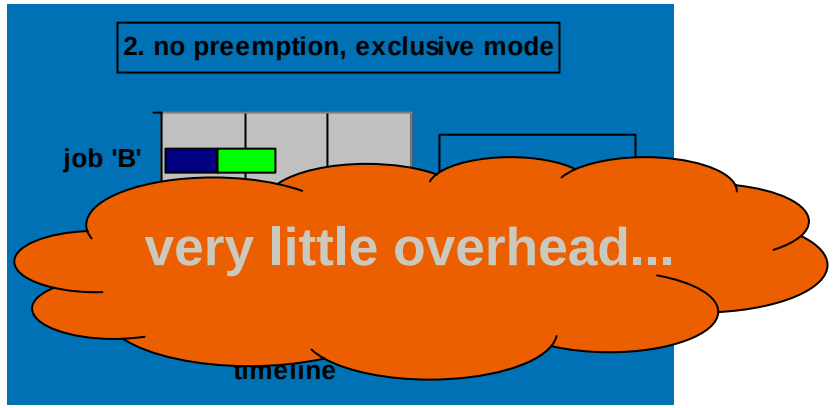
Known limitations

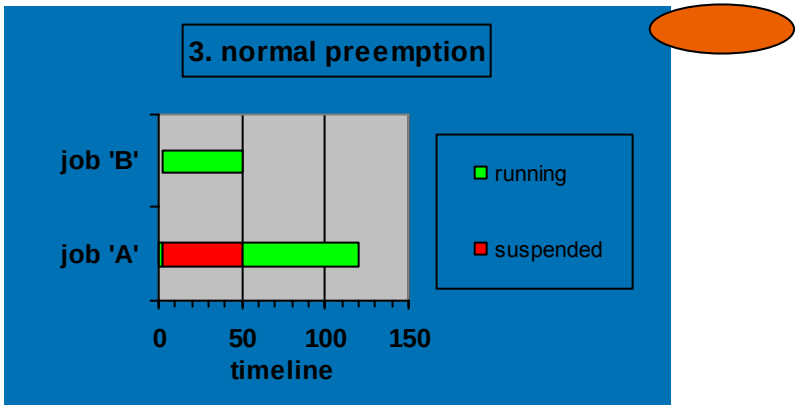
- overcommitment of GM ports
  - not possible with DL585 DC and two 8-process MPI jobs using Myrinet *(removed – Myricom added more ports to their driver)*

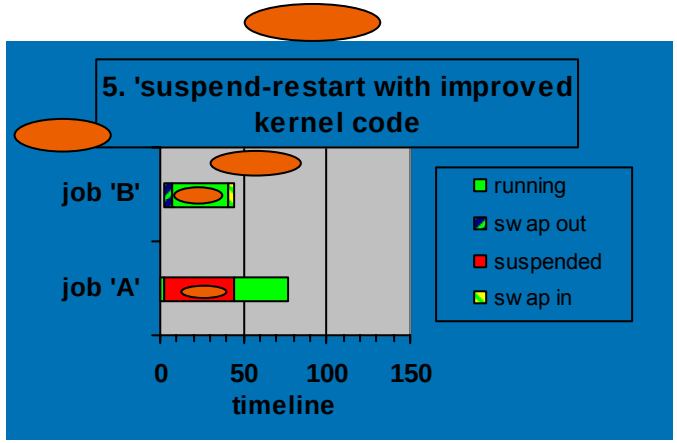
    but also: need to have MPI licenses for 2x your cores
- makes rogue processes automatic cleanup more difficult to achieve
  - but easily solved if specific user account used exclusively for preemptive queue
  - more advanced configuration can provide 99% of this functionality in other cases

Kernel improvements make it more beautiful !









Conclusion

- It’s not necessary to have full checkpointing to be able to free up resources for pre-emption.
- Managed swap is better than leaving the kernel to swap as required.
- Enhancements in the kernel can speed this up –
  - as might having fast file system handy.

future work

- test and certify s/r
- integrate with non-LSF job scheduling (OpenPBS)
- add fault-tolerant features to HP-MPI (OpenMPI?)
- …

LM on HP

The following slides contain updated results of a joint HP-DWD project to evaluate usage and performance of the LM on clusters made of Industry Standard Servers.

1. List of tested platforms and porting experiences
2. Single Node Performance
3. Parallel Performance

What's new ?

- Improved Itanium / Linux performance
- More data on HP SFS combined with LM Async I/O
- Large clusters with 512+ cores
- New chips - Woodcrest and Montecito

Target Platforms

LM-RAPS 3.0 was ported and tested on the following platforms

- RHAS 4.0 and SLES9 on Xeon and Opteron, Intel Compiler 9.1, 64 bits
- RHAS 4.0 and SLES9 on Xeon and Opteron, PGI Compiler 6.0, 64 bits
- RHAS 4.0 and SLES9 on Xeon and Opteron, PathScale Compiler 2.2, 64 bits
- RHAS 4.0 on Itanium2, Intel Compiler 9.1, 64 bits
- HP-UX 11.23 on Itanium2, HP Fortran90 v2.9, 32 and 64 bits

Comments

- No major porting problems. Very few source changes required.
- Moderate FP optimisation switches were used for all platforms. Reordering was allowed, Loss of precision was not allowed. Quality of results was not thoroughly checked.
- Few specific modules had to be compiled at lower optimisation to get rid of SIGSEGV and wrong answers. Automatic search tool had to be used to identify such modules, especially with PathScale‘s compiler.
- HP MPI 2.2 was used for all platforms.
- Some EM64T executables don‘t work with MPI structured data types.

Comments on Single Node Performance

- Itanium is the clear performance leader, despite its low clock rate.
  - Comparing the different Itanium platforms, all running at 1.6 GHz, it is obvious that cache size does matter for LM.
  - SX2000 has more headroom than ZX1 chipset.
  - Dual-Core does deliver double performance with SX2000.
  - HP-UX and Intel compilers are now equally efficient on Itanium.
    Intel v9.1 has substantially improved robustness as well as performance.
- Woodcrest DC easily outperforms Opteron DC, however some bandwidth contention can be observed when going from 2 to 4 cores.
- Opteron Dual-Core chips provide roughly a 1.5x speedup over faster clocked Single-Core chip. License cost per core is not relevant for LM.
  Use of Linux NUMA support is mandatory to get optimal performance with Opteron systems.
- Compilers optimize differently for EM64T and AMD64 targets.
  EM64T code always works on AMD but not vice versa.

Some good reasons to use HP MPI

- No need to build it yourself. Install RPM and go.
- Available for all common Linux platforms like IA32, EM64T, AMD64, IPF on Linux flavours RHAS 3,4 and SLES 9,10
- No dependency with specific compilers

- Supported interconnects
  for IPF: GigaBit TCP, Myrinet, InfiniBand, Quadrics
  for EM64T: GigaBit TCP and RDMA, Myrinet, InfiniBand, Myri10G

- Supports easy processor and memory binding on NUMA platforms
- Optimized shared memory support
- Supports full MPI-2 standard
- Lightweight instrumentation for messaging statistics and profiles
- Supported by HP

http://www.hp.com/go/mpi

Comments on parallel tests

- MPI overhead was reported by HP MPI lightweight instrumentation.
  In this context MPI overhead includes
  1. Message passing activity
  2. Busy waiting due to load imbalance
  3. Message packing and unpacking if **`ldatatypes=TRUE`**

- PathScale executable fails with case LMK.
  PGI was chosen as next best and most robust replacement.

Comments on parallel efficiency

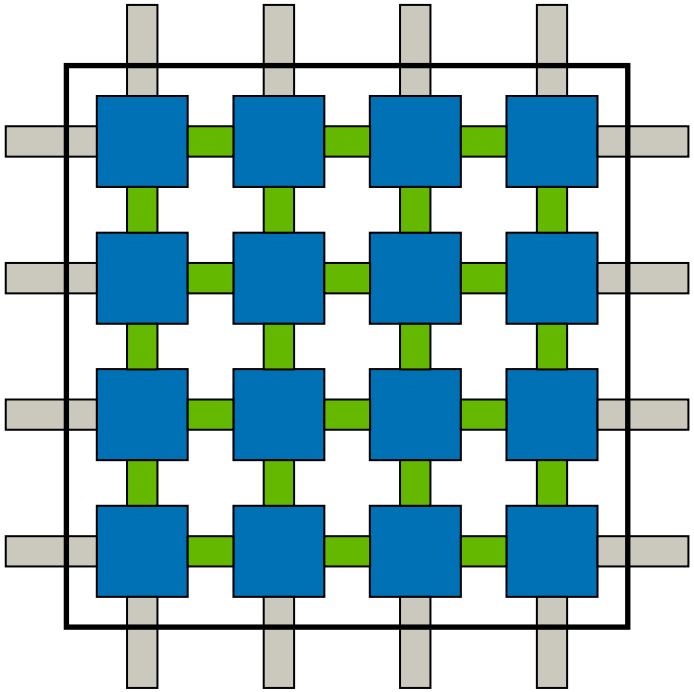
- MPI bandwidth requirements are moderate as the majority of messages are local between domain neighbors.
- Volume of global operations is small. Still there is substantial elapsed time spent in these operations. However these times are caused by synchronisation due to slight load imbalances.

- ldatatypes=.TRUE. works perfectly fine on IPF with both Linux and HP-UX. On EM64T platforms it had to be set to .FALSE. for both stability and performance reasons.
- ltime_barrier was left .FALSE. which appears to be the best setting.
- ncomm_type=3 (SendRecv) is roughly 5% slower than mode 1 (Isend, Recv, Wait) so mode 1 was chosen.

- Domain decomposition should choose NPX smaller than NPY to keep the loop count of inner loops (mostly working in X-direction) fairly high. This improves efficiency of compiler optimisations like data prefetching, unrolling and vectorisation.
  This approach is constrained by the increase of boundary perimeters causing more traffic.

Process placement in a multilevel topology, e.g. Blade clusters with IB

- MPI traffic by LM is mostly local e.g. data exchange along domain boundaries. LM domains are rectangular and the exchange traffic corresponds to the boundary length.
- This allows to optimize process placement for locality.
- Example: A blade enclosure carries 16 blades. The built-in IB switch connects the 16 blades internally and provides 8 IB uplinks. Is the number of uplinks sufficient ?



- InfiniBand switch capacity
  Total number of ports = 24
  Internal ports = 16 = 66%
  External ports = 8 = 33%
- LM bandwidth requirement
  Total number of MPI connections = 40
  Internal connections = 24 = 60%
  External connections = 16 = 40%
- Compared to an ideal topology each blade still gets 33/40 = 83% of its required external bandwidth.
- Conclusion
  With proper process placement the balance of internal and external IB bandwidth is very close to the requirements of LM.
  The slight lack of bandwidth per blade might b[illegible] noticed with artificial benchmarks but not with LM.

Summary

- LM runs trouble free and with good performance on Linux clusters.
  Linux performance with Itanium is now equivalent to HP-UX.
- HP SFS has significant advantages for cluster operation and I/O performance.
  In combination with LM Async I/O there is a substantial performance gain.
- Parallel efficiency on Linux clusters is quite acceptable up to 500 CPUs.
  The realtime requirements of 30 Minutes for a 24 hours forecast with model LMK can be achieved with clusters of ~200 CPUs.
- Despite its „low“ clock rate Itanium is still faster than any other processor.

Final Conclusion

- It is possible to fulfil the special requirements of met apps on Linux and industry standard server-based clusters.


invent