Best Practice mini-guide "JADE"

SGI Altix ICE at CINES

Tyra Van Olmen, CINES

February 2013

Table of Contents

1. Introduction

The supercomputer JADE is a computing cluster based on a scalar x86 architecture. It consist of 2880 computing nodes with a peak performance of 267 Tflop/s.

Figure 1. SGI Altix ICE Jade, source: CINES

2. System architecture and configuration

2.1. System configuration

Jade is an SGI Altix Ice 8200 scalar supercomputer with a total peak performance of 267 Tflop/s. The cluster is divided in two parts:

- Jade-1:
  - 24 racks
  - 12288 cores
  - Intel Harpertown 3.00 GHz processors
  - 32 GB of physical memory per node
  - 147 Tflop/s
- Jade-2:
  - 21 racks
  - 10752 cores
  - Intel Nehalem 2.8 GHz processors
  - 36 GB of physical memory per node
  - 120 Tflop/s

Each rack includes:

- 1 leader node

- 4 Individual Rack Units (IRU). Each IRU contains 16 computing nodes. Each of these nodes is equipped with 2 quad-core processors

In total, Jade is composed of 2880 computing nodes and 23040 cores.

Processor architecture

Jade-1 is equipped with Intel Harpertown (Xeon Quad-Core E5472) processors, and Jade-2 with Intel Nehalem (Xeon Quad-Core X5560) processors.

Intel Harpertown (Xeon Quad-Core E5472) processor

- quad-core
- speed: 3.00 GHz
- peak performance: *12 Gflop/s* per core
- caches:
  - L1: 32 KB for data and 32 KB for instructions
  - L2: 6 MB shared by 2 cores. 2 L2 caches per processor
- memory bandwidth at the level of the processor: 12.8 GB/s (through FSB)

Intel Nehalem (Xeon Quad-Core X5560) processor

- quad-core
- speed: 2.8 GHz
- peak performance: *11.2 Gflop/s* per core
- caches:
  - L1: 32 KB for data and 32 KB for instructions per core
  - L2: 256 KB per core
  - L3: 8192 KB per processor
- memory bandwidth at the level of the processor: *25.6 GB/s*

Operating system

The operating system on Jade is a Linux SUSE SLES 11 with SGI’s Propack.

Memory architecture

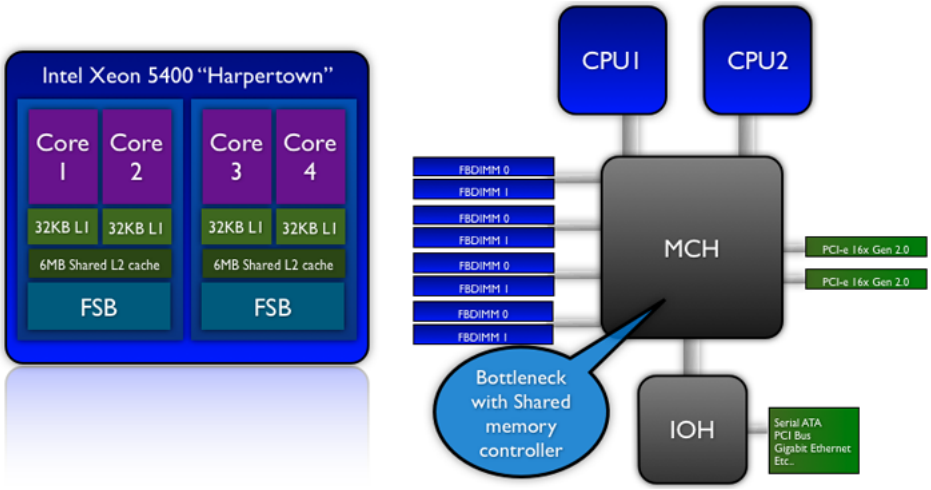
Jade-1

On Jade-1, the nodeboard is composed of:

- 2 sockets
- 8 DIMM slots of 4 GB

- The Memory Controller HUB (MCH) ASIC, also known as the Northbridge, which provides following functions:
  - Uses the Intel Greencreek chipset
  - Provides one independent 1600 MHz front side bus (FSB) interface for each processor socket
  - Has four PCIe ports that enable communication with dual InfiniBand interfaces and the Enterprise Southbridge 2E (ESB-2E) chipset
  - Supports eight Fully Buffered DIMMs (FB-DIMMs) for a memory size of *32 GB* per node
  - Enables 5.3 GB/s of read bandwidth for FB-DIMM memory, which provides a total read bandwidth of *25.6 GB/s* for the 4 dual ranked FB-DIMM channels
  - Enables the System Management Bus (SMBUS) interface
- Two 4x DDR IB HCA provide connections to the InfiniBand interconnect used in the cluster.

Figure 2. Node from Jade-1

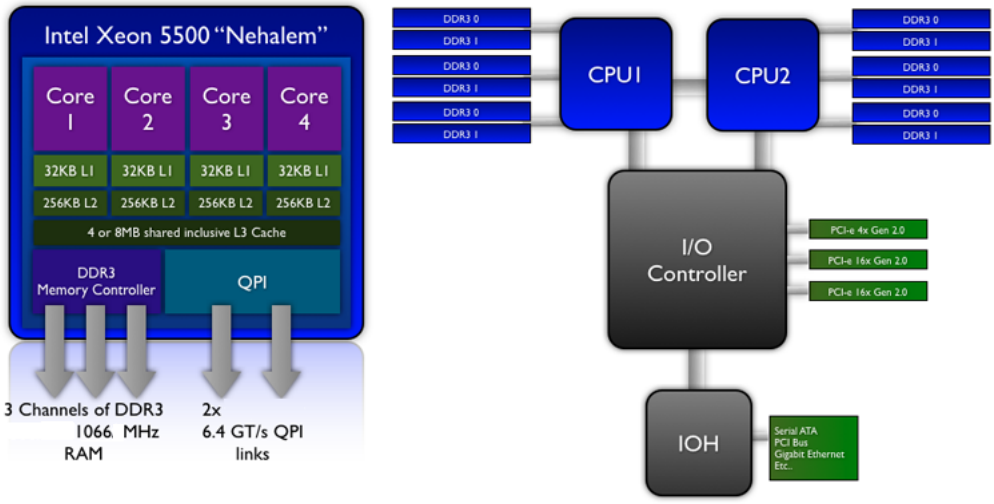


Jade-2

On Jade-2, the nodeboard is composed of:

- 2 sockets
- 2 banks of 6 DIMM slots (of 2 and 4 GB)
- Built-in DDR3 memory controller to enable direct processor-to-memory connection, which provides following functions:
  - Replaces traditional front-side bus architecture with Intel Quick Path Interconnect (QPI)
  - Supports 12 DDR3 DIMM for a total memory size of *36 GB* per node
  - Provides a total read bandwidth of *25.6 GB/s* for one DDR3 DIMM bank
- One 4x DDR IB dual port HCA provides connections to the InfiniBand interconnect used in the cluster.

Figure 3. Node from Jade-2



Network infrastructure

The network between the nodes is InfiniBand (IB 4x QDR) double plan and non-blocking. The network topology is a partial hypercube of dimension 9.

I/O subsystem

The file system is Lustre with 700 TB of raw data.

2.2. Filesystems

There are 4 file systems on Jade: `/home`, `/work`, `/scratch` and `/data`.

Figure 4. Organization of the file systems at CINES

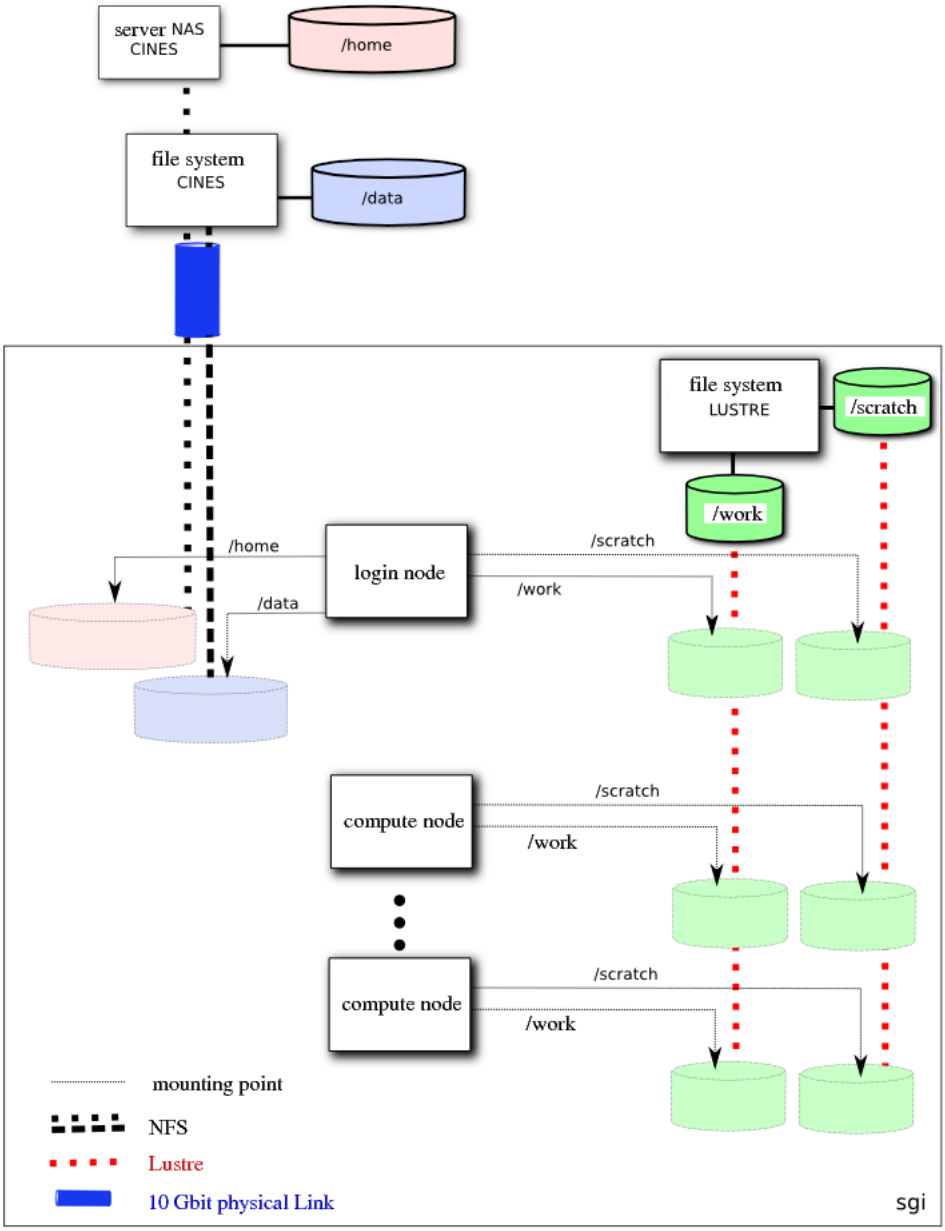


Table 1. Best usage of the file systems

Space	Usage	Protocol	Surface/bandwidth*	Limitations	Access	Services
`/home`	home directory	NFS	2 TB < 2*70 MB/s	Quotas 1 Go per user	Connexion nodes	µ-Versions backups
`/data`	Storage of compute results	NFS	540 TB disk 2*600	100000 files per unix group	Connexion nodes	Long term mass storage and backups

Space	Usage	Protocol	Surface/bandwidth*	Limitations	Access	Services
			TB band < 4*700 MB/s			
/work	Home directory on compute nodes / software	LUSTRE	26 TB < 4 GB/s	Quotas : 50 GB per unix group	Connexion nodes & compute nodes	none
/scratch	data access from within applications	LUSTRE	602 TB < 21 GB/s	Quotas : 4 TB per Unix group	Connexion nodes & compute nodes	none

The file systems are managed by Lustre. The general architecture of Lustre is composed of 2 metadata servers (MDS) in an active/passive mode and 28 data servers (OSS).

The metadata are stored in an SGI IS220 bi-controller storage bay, and the data on 11 x IS4600 bi-controller storage bays.

The MDS and OSS are both based on Altix XE 250 machines, with following configuration:

Table 2. Lustre file system configuration

MDS	OSS
• 2 Quad-Core Xeon X5472 3.0GHz/12M/1600MHz 120W processors • 8 x 4GB FB-DIMM (2 x 2GB, 667MHz) • 2 internal disks SATA, 250GB 7200 tours/min, configured in RAID-1 (mirror) • 1 HCA InfiniBand (X8 PCIe, low profile) with 1 port 4X DDR • 1 HBA LSI FC949ES PCIe Low Profile with 2 ports FC 4 Gb • 1 redundant alimentation of 900W	• 2 Quad-Core Xeon X5472 3.0GHz/12M/1600MHz 120W processors • 2 x 4GB FB-DIMM (2 x 2GB, 667MHz) • 1 internal disk SATA, 250GB 7200 tours/min • 1 HCA InfiniBand (X8 PCIe, low profile) with 1 port 4X DDR • 2 HBA LSI FC949ES PCIe Low Profile with 2 ports FC 4 Gb

The metadata are stocked on a SGI Infinite Storage 220 storage bay with 1 bi-controller with 4 FC 4Gb ports on host side and 12 SAS 15000 trs/min disks of 146GB for a total brut space of 1,75 TB and 875 useful GB (RAID 1).

The data are stocked in 10 SGI Infinite Storage 4600 storage bays, with following configuration:

- 1 bi-controller with 16 FC 4Gb ports on host side
- 16 drawers of 16 slots (256 slots in total)
- 16 SATA 500 GB 7200 trs/min disks per drawer (which is 16x16x500GB = 128 raw TB per bay)

for a total space of 602 useful TB for /scratch, and 26 useful TB for /work.

Each Lustre server (mds01, mds02, oss01 to oss20, as well as sgiadm) is connected to one InfiniBand plan: ib1. Each node is physically connected to the InfiniBand switch of one of the computing racks.

Configuration of the storage bays

- *Metadata*

The parameters of the two groups are:

- blocs of 64 KB
- activated read cache
- activated write cache
- desactivated prefetch

- *Data*

  The disks of the IS4600 bays are split as follows:

  - 24 groups of 10 disks in RAID6 (8+2), each one containing a volume of 4 TB.
  - 2 groups of 6 disks in RAID6 (4+2), each one containing a volume of 2 TB
  - 4 disks which serve as hot-spare

Each group with a pair id is monitored by the controller A, with an impair id by the controller B. This allows an increased performance in normal functioning. In a degraded functioning (if a controller breaks down or is not accessible), the second controller can take over the management of the group.

The 24 first volumes, affected to luns 0 to 23, are used for the creation of the OSTs of the file system “scratch”, whereas the 2 remaining volumes (luns 24 and 25) are used for the creation of the OSTs of the file system “local”.

The characteristics for all the groups are:

- blocs of 128 KB
- activated read cache
- activated write cache
- mirrored cache
- desactivated prefetch

In a group of 4 OSS, the luns 0 to 11 are associated with the 2 first OSS, and the luns 12 to 25 are associated with the 2 last OSS.

The 2 first OSS « see » the same volumes, which allows the transfer the handling of those volumes from one server to the other without reconfiguring the bay (active/passive redundancy). The same for the 2 last OSS.

3. System Access

Application for an account

Academic researchers from French laboratories can apply for core hours via the website edari.fr [http://edari.fr], European users can obtain core hours via the program HPC-Europa or PRACE (DECI).

For the opening of an account, consult www.cines.fr [http://www.cines.fr].

How to reach the system

- Login using username and password

  Users with a username and password can login to Jade by SSH: **ssh jade.cines.fr**

- Login with a certificate

Users with a valid certificate must use gsissh. Use following commands if you come from:

- a PRACE door-node or a PRACE site: **gsissh your-prace-user@jade-prace.cines.fr**
- the internet network : from your machine : you have to install the *gsissh* tool on your machine **gsissh your-prace-user@service4.cines.fr**

(In both cases gsissh uses the default port 2222.) Alternatively, you can use a Java webstart application called *gsissh-term*.

4. User environment and programming

4.1. Generic x86 environment

- GNU:
  - Compilers : gfortran, gcc, g++
- Intel (last version: intel/12.1.3):
  - Compilers : ifort, icc, mpifort, mpicc
  - Libraries : MKL (Math Kernel Library), intelmpi
- OpenMPI
- Perl, python, java, ruby, editors vi, gedit and emacs

Unified PRACE environment

The Unified PRACE environment is available on the node service4. Available modules are the Intel Fortran and C/C++ compilers, intelmpi and SGI MPT/2.02 libraries, scalapack, lapack, blacs, fftw, and transfer tools as Globus, gtransfer, and tgftp.

4.2. System specific environment

There is a wide range of software packages targeted at different scientific domains installed on Jade. These packages are accesible with modules environment. To see which modules are available, type:

module avail

Modules environment

For more information on using `Modules` please refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

4.2.1. Batch system

The job scheduler on Jade is PBS (Portable Batch System). Job duration is limited to 24h except some special classes accessible to authorized users only, which allow a walltime up to 120h.

Jobs can be MPI, OpenMP, or hybrid MPI/OpenMP. On Jade-2 there are 16 SMT (Simultaneous Multi Threading) threads per node available (2 threads per core).

The maximum number of physical cores for a job is 8192.

Using PBS Batch System

For more information on using Portable Batch System please refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

4.2.2. System specific compilers and libraries

The specific libraries on Jade are those from SGI : MPT which is SGI's MPI library.

Portability and compatibility with other x86 systems

Availability of standard copilers combined with the Unix/Linux operating system provides good portability. Scientific applications need recompilation if ported from another x86 system.

4.3. Known issues

Warning

On Jade the nodes are diskless, without swap memory. A memory overflow systematically kills the job.

5. Tuning Applications and Performance analysis

5.1. General optimization for x86 architecture

For detailed information on code optimization on x86 architecture refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

5.2. System specific optimization

To use the SSE (Streaming SIMD Extensions) instruction set compile with intel compiler and on Jade 1 the **-xSSE4.1** and on Jade 2 the **-xSSE4.2** flag.

5.2.1. Runtime choices and options

MPT environment variables:

`MPI_BUFFER_MAX = 0`	This variable specifies a minimum message size, in bytes, for which the message will be considered a candidate for single-copy transfer.
`MPI_IB_RAILS = 2`	With this environment variable set to 2, the MPI library will try to make use of multiple available separate IB fabrics and split its traffic across them.
`MPI_BUFS_PER_HOST = default value * 4`	
`MPI_BUFS_PER_PROC = default value * 4`	Determines the number of shared message buffers (16 KB each) that MPI is to allocate for each host/proc. These buffers are used to send and receive long inter-host messages. Default value is 96 pages (1 page = 16KB). Valid: 48 - 1 million pages in groups of 128KB
`MPI_DSM_DISTRIBUTE`	`MPI_DSM_DISTRIBUTE` activates NUMA job placement mode. This mode ensures that each MPI process gets a unique CPU and physical memory on the node with which that CPU is associated. Currently, the CPUs are chosen by simply starting at relative CPU 0 and incrementing until all MPI processes have been forked.
`MPI_OMP_NUM_THREADS, MPI_OPENMP_INTEROP, MPI_DSM_DISTRIBUTE`	These 3 variables have to be used together. `MPI_OMP_NUM_THREADS` is set to a colon separated list of positive integers, representing the val-

ue of the `OMP_NUM_THREADS` environment variable for each host-program specification on the mpirun command line. Setting the variable `MPI_OPENMP_INTEROP` modifies the placement of MPI processes to better accommodate the OpenMP threads associated with each process.

5.2.2. Memory optimizations

To take advantage of caches use data access patterns in a spatial and time locality.

5.2.3. I/O optimizations

To optimize I/O on a directory `dir`, just after creating the directory, type: **lfs setstripe –c 20 dir**

This command will divide all the files you will put in `dir` into stripes and spread these stripes over 20 OSS. In this manner the stripes can be accessed more quickly in read or write mode.

5.2.4. Specific optimized libraries

The Intel Math Kernel Library is optimized for the processors of Jade. The SGI MPT library is optimized for the network topology.

5.3. Available generic x86 performance analysis tools

On Jade the following generic performance analysis tools are available: Scalasca, tau, valgrind, gprof.

Further details

For detailed information on how to use these tools refer to the PRACE Generic x86 Best Practice Guide [http://www.prace-ri.eu/IMG/pdf/Best-Practice-Guide-Generic-x86.pdf].

5.4. System specific performance analysis tools

MPInside is a tool developed by SGI and aims at analyzing application performance. You can use the tool without re-compiling or re-linking your program. MPInside produces raw data in simple text format, which you can process with scripting tools like awk.

To launch a job with MPInside, you should add the MPInside library location to the `LD_LIBRARY_PATH`, as well as the bin location to the `PATH` and type: **mpiexec MPInside <cmd>**