Java-based communication in a High Performance Computing environment

Aidan Fries

University of Barcelona

Overview of my PhD

Use of Java for intensive applications in High Performance Computing (HPC) environments

Overview of this presentation

5 Sections:

1. Introduction
2. Intermediate Data Updating (IDU)
3. MPJ Express
4. DpcbTools
5. Summary

Introduction

Introduction - Explanations:

High Performance Computing (HPC): the use of computer clusters or supercomputers (~above 1 Tflops) to solve large computational problems.

Cluster

↓

Node

↓

Processor

↓

Core (processing elements)

Homogeneous or Heterogeneous →

Introduction – Java in HPC

Java advantages:

- OOP, modular, fast development, easily maintain
- Inherently multi-threaded
- Compile once, run anywhere

Militating against Java in HPC

- ~~Perception of poor performance~~ JIT Compiler
- ~~**Lack of Java based communication middleware – which are needed for communication strategies such as MPI**~~ MPJ Express

Introduction - MPI

Message Passing Interface (MPI): a language independent specification of an API that allows processes to communicate with each other.

Processes pass messages (data) amongst each other

API Overview:

- Point-point
  - Send, Receive...
- Collective
  - Broadcast, Scatter, Gather...

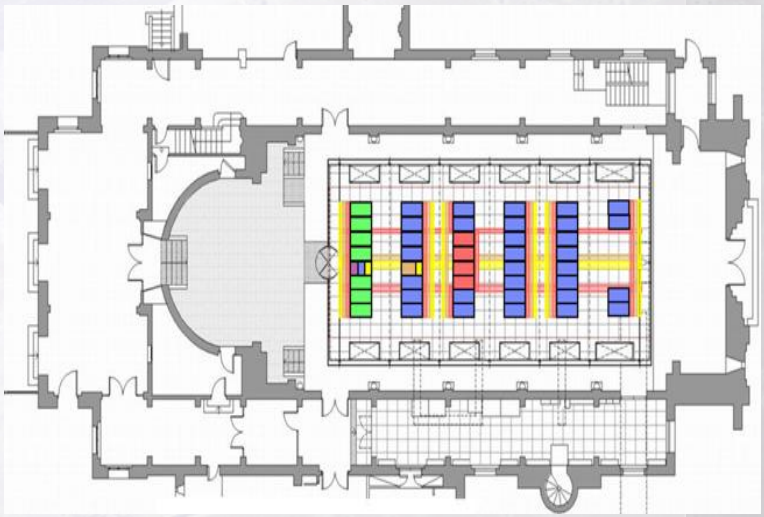
Introduction - DPCB

Data Processing Centre Barcelona (DPCB)

- Barcelona Supercomputing Centre
- Centre of Supercomputing of Catalonia (CESCA)

MareNostrum – relevant info for this talk:

2560 nodes



Global Parallel File System (GPFS)

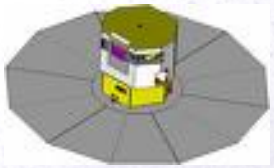
- High performance shared disk
- 270 TB

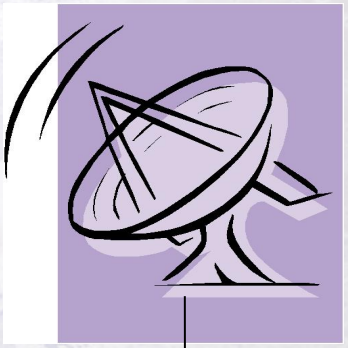
2 networks:

- Gigabit Ethernet - GPFS
- Myrinet

Intermediate Date Updating (IDU)

IDU - Within the data reduction chain





Calibration of PSF & CDM
DPCB

IDU
DPCB

IDT
ESAC

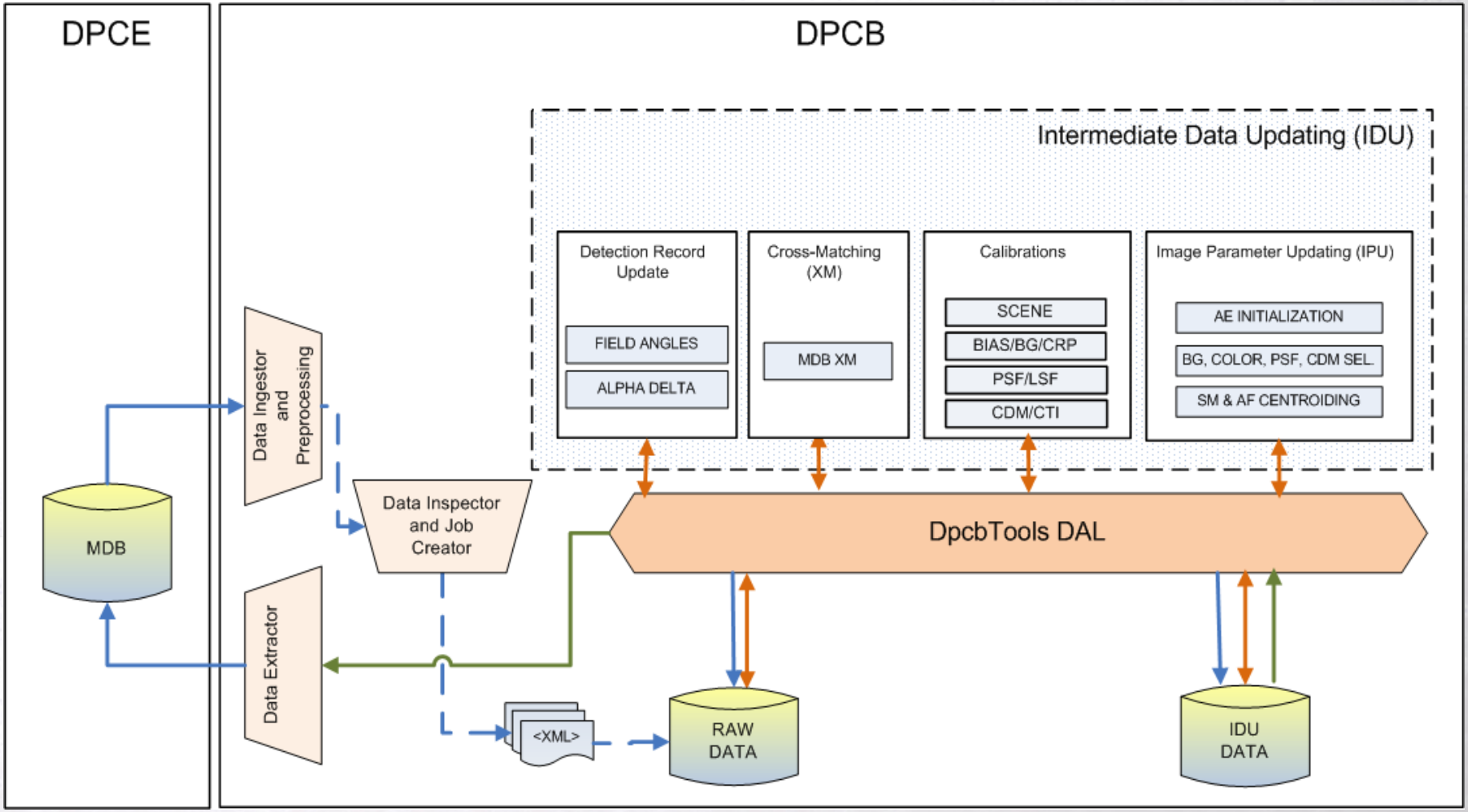
AGIS & Photometry
ESAC - IoA

IDU - Overview

- Intermediate Data Updating (IDU)
- Part of the core data reduction chain of processes
- Composed of a number of sub-processes
- Will run at the DPCB, one or more times per cycle.
- All existing data processed each time.
- The most data intensive process within the reduction chain.

It represents a large data management challenge

IDU - overview



IDU – scaling test results

Scaling tests:

1. Increasing the number of nodes along with the number of job files
2. Increasing the no. job files
3. Increasing the number of nodes

Note: the purpose of these tests was not to test the current implementation of IDU (which is still quite basic), but instead, to test how well the system (including network speeds) scales

IDU – Scaling test1

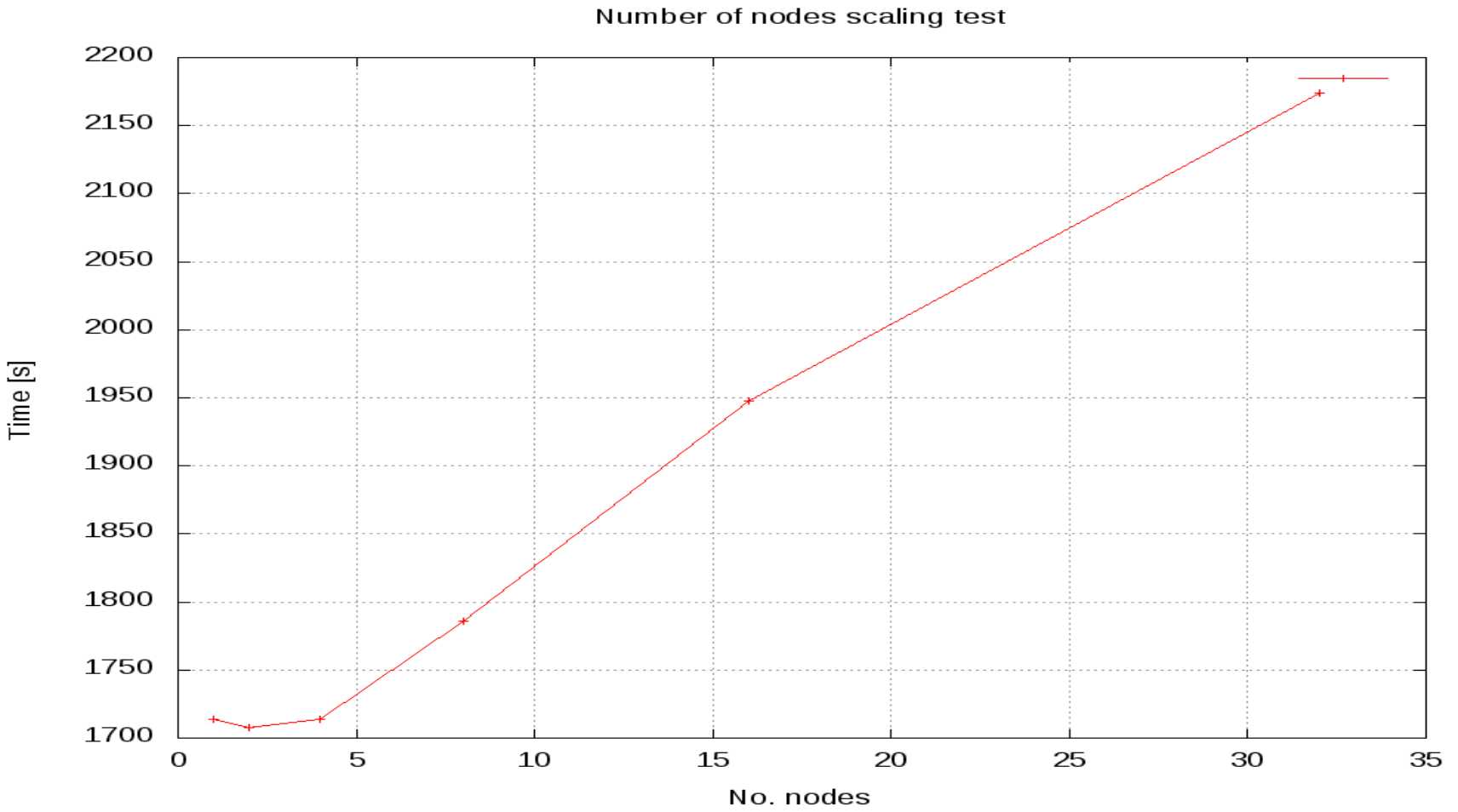
Increasing the number of nodes along with the number of job files

Test	No. job files	No. nodes	No. tasks	No. tasks per node	No. job files per node	Time Taken
1.1	20	1	4	4	20	1714
1.2	40	2	8	4	20	1708
1.3	80	4	16	4	20	1714
1.4	160	8	32	4	20	1786
1.5	320	16	64	4	20	1948
1.6	640	32	128	4	20	2174

Note: all times are given in seconds

IDU – Scaling test1

Increasing no. of nodes along with no. of job files



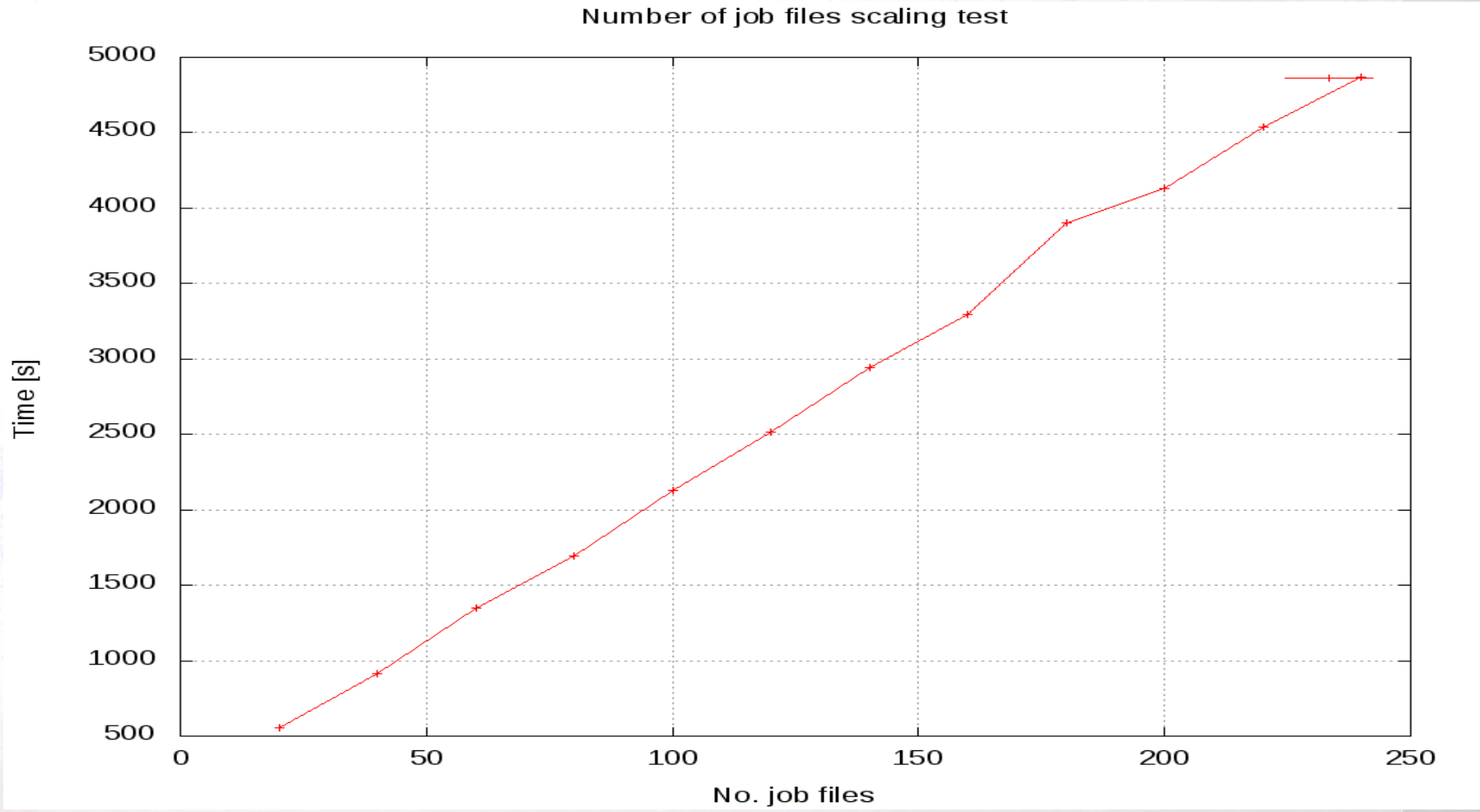
IDU – Scaling test2

Increasing the no. job files

Test	No. job files	No. nodes	No. tasks	No. tasks per node	No. job files per node	Time Taken
2.1	20	4	16	4	5	554
2.2	40	4	16	4	10	914
2.3	60	4	16	4	15	1350
2.4	80	4	16	4	20	1696
2.5	100	4	16	4	25	2128
2.6	120	4	16	4	30	2516
2.7	140	4	16	4	35	2941
2.8	160	4	16	4	40	3294
2.9	180	4	16	4	45	3902
2.10	200	4	16	4	50	4129
2.11	220	4	16	4	55	4540
2.12	240	4	16	4	60	4871

IDU – Scaling test2

Increrasing the no. of job files

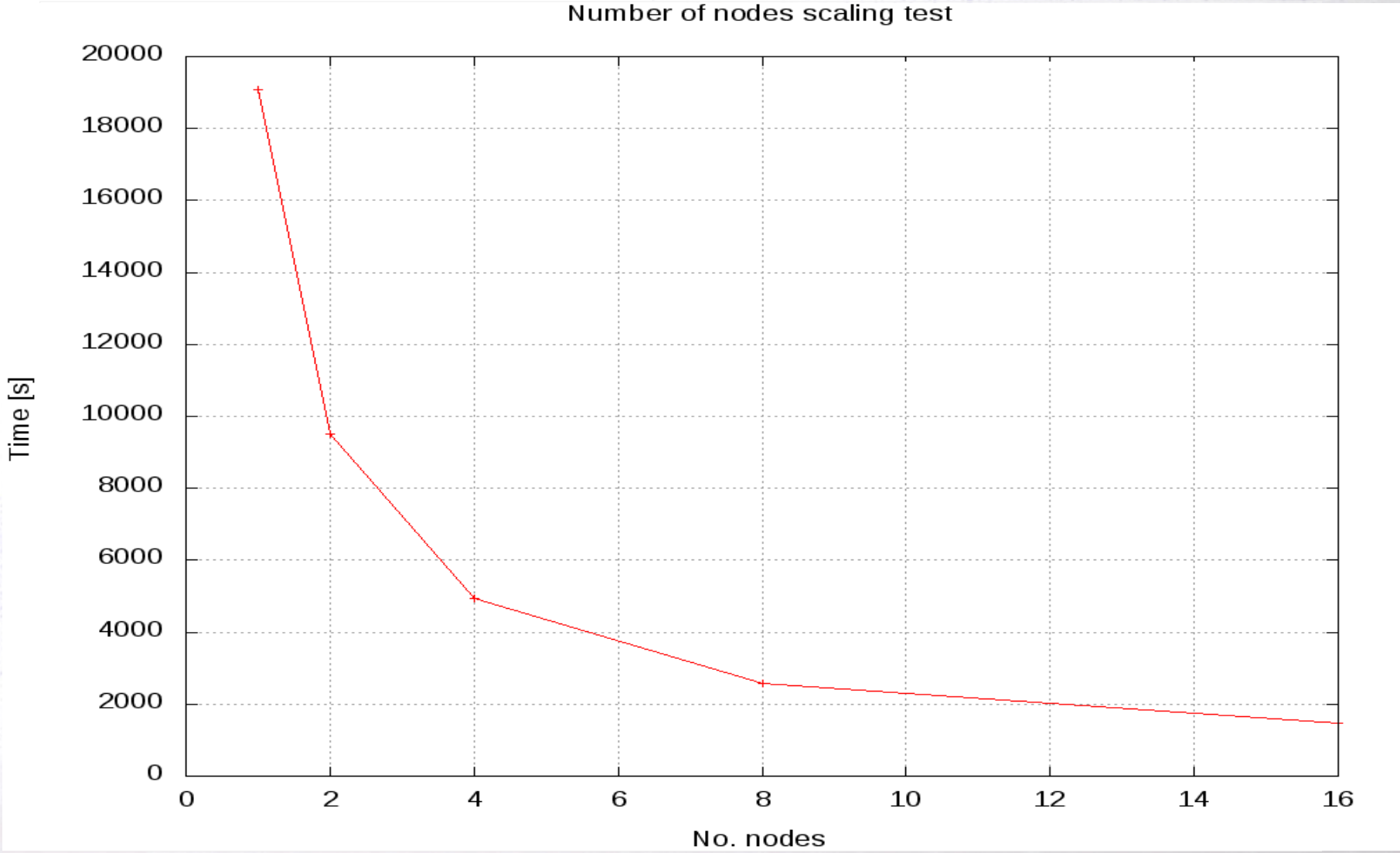


IDU – Scaling test3

Increasing the no. job files

Test	No. job files	No. nodes	No. tasks	No. tasks per node	No. job files per node	Time Taken	Percentage decrease in Time
3.1	240	1	4	4	240	19084	-
3.2	240	2	8	4	120	9504	50%
3.3	240	4	16	4	60	4932	48%
3.4	240	8	32	4	30	2588	47%
3.5	240	16	64	4	15	1465	43%

IDU – Scaling test3



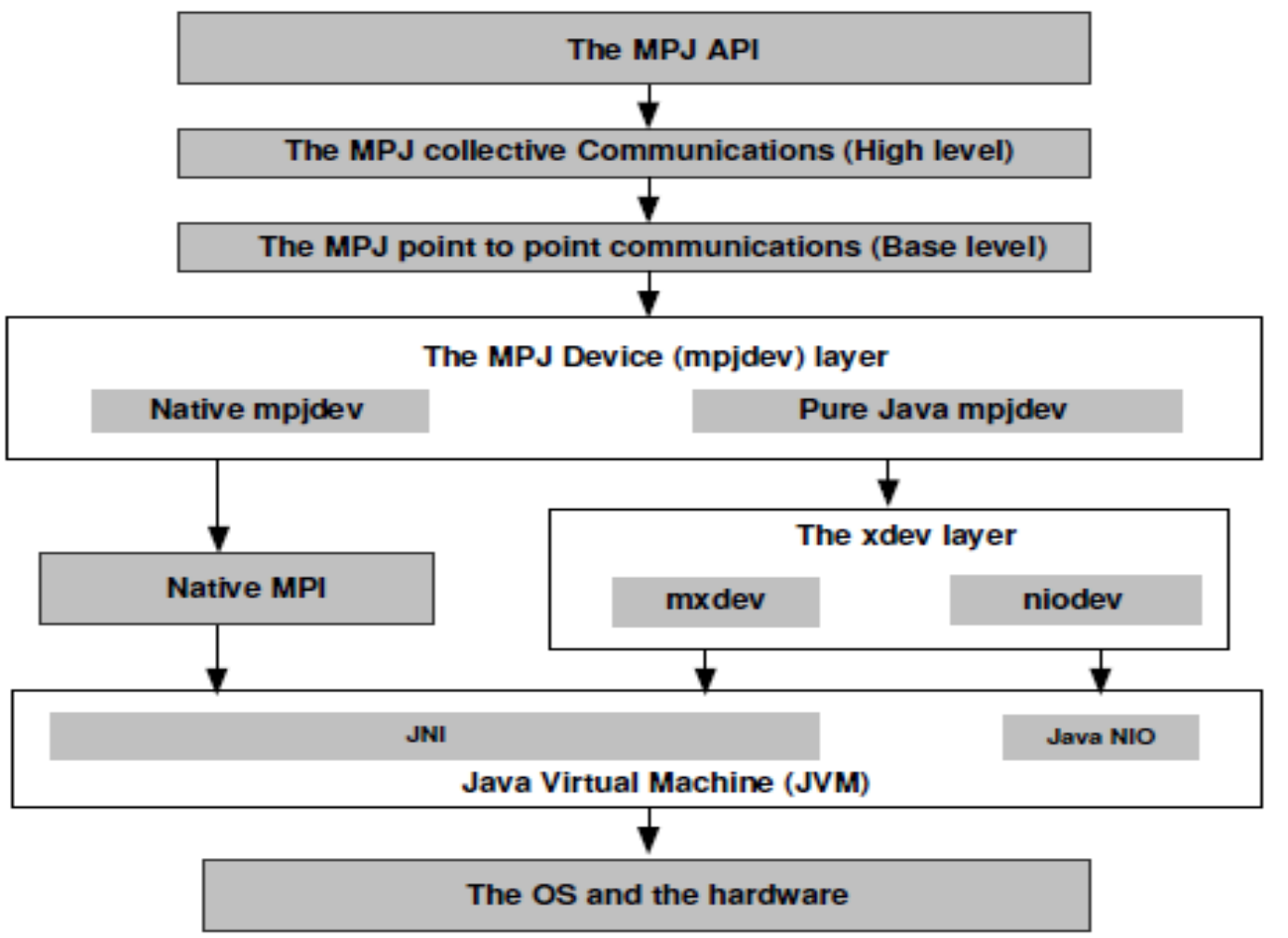
MPJ Express

MPI implementations in Java

	Pure Java Impl.	Socket impl.		High-speed network support			API		
		Java IO	Java NIO	Myrinet	InfiniBand	SCI	mpiJava 1.2	JGF MPJ	Other APIs
Jcluster	✓	✓							✓
Parallel Java	✓	✓							✓
mpiJava				✓	✓	✓	✓		
P2P-MPI	✓	✓	✓				✓		
MPJ Express	✓		✓	✓			✓		
MPJ/Ibis	✓	✓		✓				✓	
F-MPJ	✓	✓		✓	✓	✓	✓		

Guillermo Taboada
University of A Coruña

MPJ Express


The MPJ API
The MPJ collective Communications (High level)
The MPJ point to point communications (Base level)
The MPJ Device (mpjdev) layer
Native mpjdev
Pure Java mpjdev
Native MPI
The xdev layer
mxdev
niodev
JNI
Java NIO
Java Virtual Machine (JVM)
The OS and the hardware

MPJ Express - running

Job Script

```
#!/usr/bin/csh
#
# @ initialdir = .
# @ job_name = myr_2_2
# @ output = myr_2_2%j.out
# @ error =  myr_2_2%j.err
# @ total_tasks = 2
# @ tasks_per_node = 1
# @ wall_clock_limit = 00:05:00

# writing machine name to file
/usr/local/bin/sl_get_machine_list -j $SLURM_JOBID > machines

# start mpj daemon
mpjboot machines

sleep 1

# start the test
mpjrun.sh -np 2 -dev mxdev -Djava.library.path=$MPJ_HOME/lib PingPong 100 1 4194304

# kill mpj daemon
mpjhalt machines
```

MPJ Express – Gigabit Ethernet

```
Starting process <1> on <s19c2b08>
Starting process <0> on <s19c2b05>
 PingPongBench with 2 process and 100 iterations per array size
 T Min(s)          T Avg(s)       T(us)      Bw(MB/s)       NPROCS     size
0.000121          0.000146        121        0.0083          2          1
0.000118          0.000195        118        0.0169          2          2
0.000119          0.000149        119        0.0336          2          4
0.000126          0.000196        125        0.0636          2          8
0.000119          0.000151        119        0.1342          2          16
0.000127          0.000166        126        0.2523          2          32
0.000122          0.000229        122        0.5243          2          64
0.000123          0.000166        122        1.0445          2          128
0.000133          0.000158        132        1.9312          2          256
0.000137          0.000162        136        3.7413          2          512
0.000152          0.000180        151        6.7531          2          1024
0.000173          0.000215        173        11.8319         2          2048
0.000187          0.000220        186        21.9131         2          4096
0.000215          0.000323        214        38.1774         2          8192
0.000286          0.000315        285        57.3618         2          16384
0.000434          0.000462        434        75.4330         2          32768
0.000747          0.000789        747        87.7084         2          65536
0.001932          0.002359        1932       67.8376         2          131072
0.003179          0.003554        3179       82.4592         2          262144
0.005907          0.006404        5906       88.7634         2          524288
0.011266          0.011671        11266      93.0725         2          1048576
0.021793          0.022515        21793      96.2289         2          2097152
0.043041          0.044247        43040      97.4496         2          4194304
 PingPongBench ******* TEST COMPLETED *********
```

MPJ Express – Myrinet

```
Starting process <0> on <s10c5b06>
Starting process <1> on <s12c3b06>
 PingPongBench with 2 process and 100 iterations per array size
 T Min(s)          T Avg(s)      T(us)     Bw(MB/s)       NPROCS    size
0.000027          0.000038       26         0.0374         2          1
0.000027          0.000036       26         0.0749         2          2
0.000027          0.000038       27         0.1472         2          4
0.000028          0.000050       27         0.2893         2          8
0.000028          0.000038       27         0.5785         2          16
0.000029          0.000043       28         1.1185         2          32
0.000029          0.000040       28         2.2370         2          64
0.000029          0.000039       29         4.4006         2          128
0.000029          0.000038       29         8.8012         2          256
0.000031          0.000045       30         16.7773        2          512
0.000030          0.000046       29         34.6367        2          1024
0.000035          0.000044       34         58.8345        2          2048
0.000043          0.000055       43         94.3953        2          4096
0.000066          0.000106       66         123.5959       2          8192
0.000111          0.000127       110        148.1026       2          16384
0.000249          0.000276       249        131.3947       2          32768
0.000400          0.000450       399        164.0083       2          65536
0.000704          0.000754       703        186.2317       2          131072
0.001349          0.001424       1349       194.2600       2          262144
0.002819          0.002906       2818       186.0111       2          524288
0.005817          0.005992       5816       180.2626       2          1048576
0.011607          0.011704       11607      180.6773       2          2097152
0.023186          0.023504       23186      180.8965       2          4194304
 PingPongBench ******* TEST COMPLETED *********
```

DpcbTools

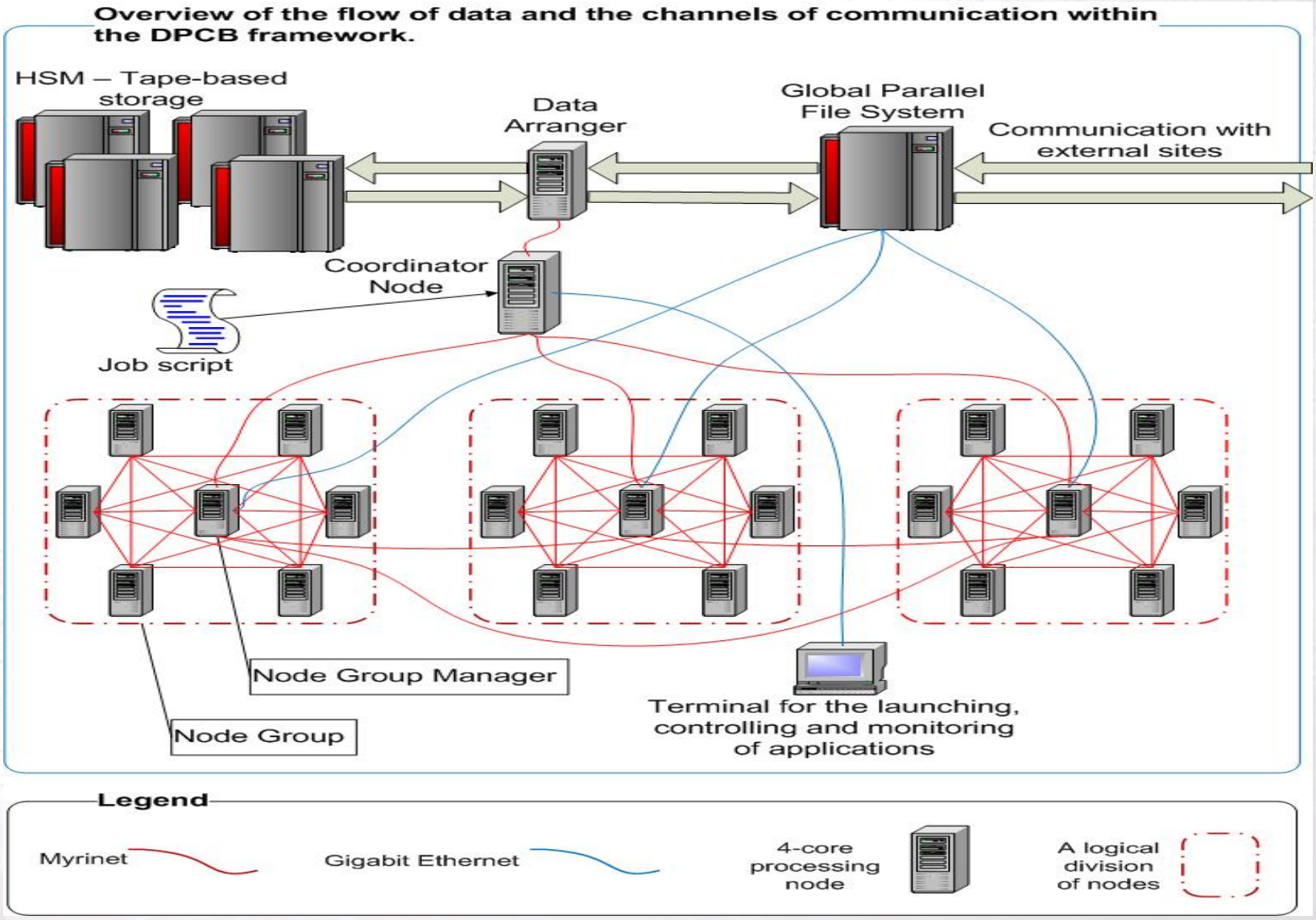
DpcbTools

Common toolbox for Gaia software running at DPCB. Allow Gaia software to make best use of available hardware.

Functionality to include:

- DAL – to allow access to the data repositories
- Creation, launching and controlling jobs
- Monitoring
- Communication
    - Built on top of MPJ Express/F-MPJ

DpbcTools - communication



Summary

- Java not commonly used in HPC – but great potential
- MPJ Express – stable, extendable java implementation of MPI
- DpcbTools could make use of MPJ Express to allow for data distribution, collection at DPCB