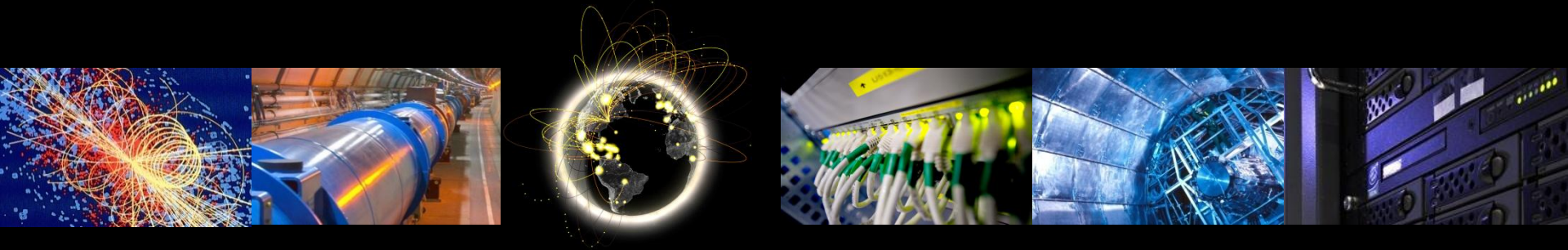


# Multi-core support in IS

Maria Alandes Pradillo, CERN


CERN IT Department, Grid Technology Group

GDB 12<sup>th</sup> September 2012



# CE extensions pre-GDB July

- Multicore use cases to be addressed

| cores/job specification  |  |   |
|---|--|---|
|   | Sites offer a discrete choice of cores/job, e.g. 1,4,8   | Sites offer a range of choices of cores/job, e.g. 1-8   |
| <b>VO requests an exact number of cores/job, e.g. NCORES=4</b>  | The implementation at the site level is supposed to be reasonably easy. The match of jobs to resources is difficult unless there is an agreement between sites and VOs on the available values. The values may differ from site to site. Jobs have to be tailored to the site. | The match is reasonably easy. The implementation and the optimization at the batch system level may be very difficult.  |
| <b>VO requests a range of cores/job, e.g. MINCORES=4; MAXCORES=8;</b>                                       | The match is reasonably easy. The implementation at the site level is supposed to be reasonably easy. Since the VO doesn't in general need an exact number of cores it should not be difficult for the VO. The site chooses NCORES as they prefer in the specified range.      | The match is very easy, the implementation and the optimization at the batch system level may be difficult, but still the site has the possibility to choose NCORES to optimize resource usage. |

# IS proposal

- IS proposal to address the use cases

| Use Case                        | Glue | Attribute (single value attributes, that is one instance per Share/CE) | Syntax (numbers in ascending order)                   | YAIM configuration | JDL  |
|---------------------------------|------|--|---|--------------------|--|
| whole nodes                     | 1.3  | GlueCECapability   | GlueCECapability: wholenode                           | (1)                | Requirements = other.GlueCECapability=="wholenode";              |
|                                 | 2.0  | GLUE2EntityOtherInfo in the Share class                                | GLUE2EntityOtherInfo: wholenode                       | (2)                | Requirements = Member ("wholenode", other.GLUE2EntityOtherInfo); |
| multicore                       |      | GlueCECapability   | GlueCECapability: multicore                           | (1)                | Requirements = other.GlueCECapability=="multicore";              |
|                                 | 2.0  | GLUE2EntityOtherInfo in the Share class                                | GLUE2EntityOtherInfo: multicore                       | (2)                | Requirements = Member ("multicore", other.GLUE2EntityOtherInfo); |
| fixed multi-core                | 1.3  | GlueCECapability   | GlueCECapability: RequestableCores=1[,2 [,...,N]]     | (1)                | (3)  |
|                                 | 2.0  | GLUE2EntityOtherInfo in the Share class                                | GLUE2EntityOtherInfo: RequestableCores=1[,2 [,...,N]] | (2)                | (3)  |
| variable multi-core             | 1.3  | GlueCECapability   | GlueCECapability: RequestableCores=1[-N]              | (1)                | (3)  |
|                                 | 2.0  | GLUE2EntityOtherInfo in the Share class                                | GLUE2EntityOtherInfo: RequestableCores=1[-N]          | (2)                | (3)  |
| Any number of cores up to a max | 1.3  | GlueCECapability   | GlueCECapability: MaxSlotsPerJob=N                    | (1)                | (3)  |
|                                 | 2.0  |  | GLUE2ComputingShareMaxSlotsPerJob=N                   | (2)                | Requirements = other.GLUE2ComputingShareMaxSlotsPerJob==2;       |

- All details can be found in:

<https://tomtools.cern.ch/confluence/display/IS/GLUE2UseCaseMultiCoreJobs>

# Open issues (I)

- Syntax issues

- One attribute per value. i.e.:

```
GLUE2EntityOtherInfo: RequestableCores=4  
GLUE2EntityOtherInfo: RequestableCores=8  
GLUE2EntityOtherInfo: RequestableCores=16
```

- Allows for an easy JDL 😊
- Increases the volume of published information 😞
  - Need to assess impact. i.e. How many attributes are published in practice.

- One single attribute. i.e.:

```
GLUE2EntityOtherInfo: RequestableCores=4,8,16
```

- More complex JDL due to the lack of proper classads primitives 😞
- Simple and compact published information 😊

# Open issues (II)

- Realistic use cases for multicore support?

- Is this really needed?

```
GLUE2EntityOtherInfo: RequestableCores=4-32  
GLUE2EntityOtherInfo: RequestableCores=2,4,8
```

- Isn't it enough something like...

```
GLUE2EntityOtherInfo: multicore
```

- With the mechanism in place in the WN which tells the job how many cores it is allowed to use
      - Is this mechanism available?

- What about...

```
GLUE2ComputingShareMaxSlotsPerJob=16  
GLUE2EntityOtherInfo: multicore_granularity=4
```

- would allow requests for 4,8,12,16 cores to be fulfilled.

# Open Issues (III)

- Do we really need to publish wholenodes or multicore at all?
  - It seems we need to publish wholenodes since this is not provided by all CEs
  - For multicore it would be enough to know the maximum number of cores
    - If a CE can do 8 and 16 cores, why not 12?
      - Maybe scheduling is not optimal for certain combination of cores/job?
      - Feedback from sysadmins needed!



## Conclusions:

- IS can publish any reasonable configuration (complex parsing to be taken into account)
- We don't have enough understanding of the use cases we need to satisfy
  - What batch systems can actually do?