Representing Compute Resources for the Grid Information Services

A proposal to be discussed as part of the Gridforum

Gregor von Laszewski

Mathematics and Computer Science Division at Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439 U.S.A.

Steve Fitzgerald

University of Southern Ccalifornia Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6695
U.S.A.

DRAFT: This draft is not spell checked or corrected for gramatical mistakes. Furthermore, it is not for distribution to the public.
Editor: Gregor von Laszewski, will be changed to Brett and Karin
gregor@mcs.anl.gov

Contents

1 Introduction **2**

2 Notation **2**
2.1 Namespaces 2
2.2 Relevant IETF and ITU documents 2

3 Grid::PhysicalResource **2**

4 Grid::ImageResource **3**

5 Grid::ComputeResource **4**

6 Grid::OperatingSystemInformation **5**

7 Grid::MemoryInformation **5**

8 Grid::CacheInformation **6**

9 Grid::BenchmarkInformation **6**

10 Grid::CpuInformation **7**

11 Grid::SystemDynamicInformation **8**

12 Grid::TestBed **8**

1 Introduction

We propose a simple set of objects that allow to describe compute resources in the Grid. The information about the availability or a particular feature of a compute resource is essetnial for building applications that utelize any resource in the Grid. While providing a simple way to express such resources, we hope to encourage tool development based on the uniformity of the data. While doing so, we hope to reduce the development cost.

- where available, compute resource objectclass definitions should be based on industry standards
- compute resource objectclass definitions should enable a uniqe identification of a compute resource
- compute resource objectclass definitions must provide support user added services, like programms available or running on the resource.
- compute resource objectclass definitions must support ownership (allow children) and restrict access for user-specific objects
- compute resource objectclasses definitions must support the capture of personal information about grid users for tracking/reporting purposes (this is actually addressed by the point above in conjunction with the accounting working group)

2 Notation

We have chosen the notation introduced in the GIS-WG-1 proposal. We have not used DSML because the standard does not include some features we like to see supported. In addidtion, one has to note that the current DSML standard is not vewed by the originators as fixed. In order to avoid a lengthy transformation process we have chosen to continue using the format used over the last year.

2.1 Namespaces

We find it essential to use the concept of namespaces in order to distingush objectclasses developed by a variety of contributors. Thus we do not just take, for example, the Grid computeResource classes but should augment them with the namespace "Grid::".

2.2 Relevant IETF and ITU documents

- ...

3 Grid::PhysicalResource

Grid::PhysicalResource

Each Grid Resource can have a Physical and a user specific appearance. The distinction between the two is based on the addition of "Physical" and "Image" to the object name. Most users will only use Physical Resources. An examples for an image resources is a compute resource which could be a heterogenous agglomerated out of many different components but should appear to the user as a single resource. A physical resource can have multiple images(compare [**?**]).

```
Grid::PhysicalResource OBJECT-CLASS ::= {
  SUBCLASS OF { Grid::Top }
  RDN = rn (resourceName)
  AGGREGATES{
     Grid::Locality
  }
  MAY CONTAIN{
    resourceOwner            :: dn,
    resourceAdministrator    :: dn,
    resourceProvider         :: dn,
    description              :: cis,
    documentation            :: cis,
    imageObject              :: dn,
    url                      :: url,
  }
}
```

Attributes

url: a url which can point to the description of the resource

imageObject: (dn) pointer to the dn of the image object

resourceOwner: (dn) refers to organization or person that owns the physical element. Please, note that the name owner is used by netscape internal attributes. Thus it is necessary to have the prefix “resource”. For consistency we add it to administrator and provider

resourceAdministrator: (dn) the systems administrator for this resource

resourceProvider: (dn) the person provides permission to use the resource. In many cases this is the same as the resourceAdministrator.

description: (cis) textual description of the resource

documentation: (cis) pointers to documentation (books, html, ...)

4 Grid::ImageResource

A physical resource can have multiple images and appear differently to different users. Grid::ImageResource

```
Grid::ImageResource OBJECT-CLASS ::= {
  SUBCLASS OF {Grid::Top}
  RDN = rn_image (resourceImageName)
  MUST CONTAIN {
     type :: cis,
     imageOf :: dn
   }
}
```

Attributes

type: (cis) view of provider/user/IP/OSI/...

imageOf: (dn) pointer back to the physical resource

5 Grid::ComputeResource

The compute resource object is extended with many of the objects in this section. Grid::ComputeResource

```
Grid::ComputeResource OBJECT-CLASS ::= {
  SUBCLASS OF { GridPhysicalResource }
  RDN = hn (hostName)
  CHILD OF {
    Grid::organizationalUnit;
    Grid::organization;
    Grid::testbed;
    Grid::Group;
    Grid::Root;
  }
  AGGREGATES {
    Grid::OperatingSystemInformation
    Grid::MemoryInformation
    Grid::CacheInformation
    Grid::BenchmarkInformation
    Grid::CpuInformation
    Grid::SystemDynamicInformation
  }
  MAY CONTAIN {
    canonicalSystemName         :: cis,
    manufacturer                :: cis,
    model                       :: cis,
    serialNumber                :: ces ::singular,
    machineHardwareName         :: cis,
    hostID                      :: ces ::singular,
    type                        :: cis
  }
}
```

Attributes

canonicalSystemName: (cis) a string indicating the architecture-manufacturer-operatingSystem as provided by gnu's config.guess, e.g., sparc-sun-solaris2.5.1.

manufacturer: (cis) the manufacturer of the compute resource

model : (cis) the model of the compute resource. On many systems this is different from the machine hardware name.

serialNumber : (ces) the serial number of the compute resource

machineHardwareName: (cis) the machine hardware name as given out by the vendor

hostID : (ces) the host id number as given by the vendor

type : (cis) the type of the compute resource This includes one ore more out of the following list:

- Workstation - for a workstation
- PC - for a personal computer
- SIMD - for a SIMD compute resource
- MIMD - for a MIMD compute resource

- SM - for a compute resource using shared memory between multiple nodes
- DM - for a compute resource using distributed memory between multiple nodes

6 Grid::OperatingSystemInformation

Operating system related attributes are always created in conjunction with a Grid::ComputeResource. It contains information about the resources operating system. Obtaining operating system information is not as straight forward as one might think. Most of the information can be obtained with scripts like config.guess and sysinfo as well as their appropriate counter parts on Windows machines. In many cases it is sufficient to use the canonical system name which is used during autoconfigure. A modified os-guess script, which is based on the GNU config.guess exists which simplifies the identification of the attribute values related to the operating system information.

Grid::OperatingSystemInformation

```
Grid::OperatingSystemInformation AGGREGATE-CLASS ::= {
  MAY CONTAIN {
    operatingSystemName     :: cis :: singular,
    operatingSystemVersion  :: cis :: singular,
    operatingSystemRelease  :: cis :: singular,
    operatingSystemType     :: cis :: singular,
  }
}
```

Attributes

OSName : (cis) stores the name of the OS

OSVersion: (cis) stores a version number

OSRelease: (cis) stores the release version and release are sometimes different

OSType : (cis) POSIX, BSD, AT&T a feature which can be useful in future

7 Grid::MemoryInformation

Grid::MemoryInformation is always created in conjunction with a Grid::ComputeResource. It contains both highly dynamic and relatively static information about the resources memory. Currently many of these fields are not used by Grid related software. Nevertheless, we would like to keep this object for future purposes.

Grid::MemoryInformation

```
Grid::MemoryInformation AGGREGATE-CLASS ::= {
  MAY CONTAIN {
    Physicalmemorysize          :: int,
    freePhysicalMemory          :: int,
    physicalMemoryAccessTime    :: cisfloat,
    virtualMemorySize           :: int,
    freeVirtualMemory           :: int,
    totalSwapSpace              :: int,
    freeSwapSpace               :: int,
    pageFaultRate               :: cisfloat,
  }
}
```

Attributes

physicalMemorySize : (int) specifies the total size of the main memory in KB

freePhysicalMemory : (int) specifies the free main memory in KB

physicalMemoryAccessTime : (cisfloat) specifies the average access Time of the main memory in ms

virtualMemorySize: (int) specifies the virtual memory size in KB note: define virtual memory

freeVirtualMemory: (int) specifies the free virtual memory in KB

totalSwapSpace : (cis,numericStringSyntax) specifies the total swap space in KB

freeSwapSpace: (int) specifies the free swap space in KB

pageFaultRate: (cisfloat) specifies the page fault rate (pages/s)

8 Grid::CacheInformation

Grid::CacheInformation

```
Grid::CacheInformation AGGREGATE-CLASS ::= {
  MAY CONTAIN {
    totalDataCache          :: int,
    totalInstructionCache   :: int,
  }
}
```

Attributes

totalDataCache : (int) specifies the total data cache size in K

totalInstructionCache: (int) specifies the total instruction cache size in K

9 Grid::BenchmarkInformation

Grid::BenchmarkInformation information is always created in conjunction with a Grid::ComputeResource. It contains benchmark information for the resource.
Benchmarking is a difficult issue in computer science. The numbers presented here are used for a simple indication of the computational power of a compute resource. In many cases a benchmark with a user application is necessary.

Grid::BenchmarkInformation

```
Grid::BenchmarkInformation AGGREGATE-CLASS ::= {
  SUBCLASS OF GridComputeResource
  RDN = hn (hostName)
  MAY CONTAIN {
    SPECint92       :: cisfloat,
    SPECfloat92     :: cisfloat,
    lapack100       :: cisfloat,
    lapack500       :: cisfloat,
    lapack1000      :: cisfloat,
    mflops          :: cisfloat,
  }
}
```

Attributes

SPECint92: (float) This attribute stores the SPECint92 rating of the Grid::ComputeResource.

SPECfloat92: (float) This attribute stores the SPECfloat92 rating of the Grid::ComputeResource.

lapack100: (float) This attribute stores the LAPACK rating of the Grid::ComputeResource for solving a Matrix of size 100.

lapack500: (float) This attribute stores the LAPACK rating of the Grid::ComputeResource for solving a Matrix of size 500.

lapack1000 : (float) This attribute stores the LAPACK rating of the Grid::ComputeResource for solving a Matrix of size 1000.

mflops : (float) This attribute stores a mflop rating of the Grid::ComputeResource.

10 Grid::CpuInformation

Grid::CpuInformation

The CPU information is always created in conjunction with a Grid::ComputeResource. It contains both highly dynamic and relatively static information about the resources processor(s) as well as current load information.
On some systems it might be possible to find a particular floating point unit. Though not essential, it is possible to store this information along with the cputype.

```
Grid::CpuInformation AGGREGATE-CLASS ::= {
  MAY CONTAIN {
    cpuType                    :: cis,
    fpuType                    :: cis,
    cpuCount                   :: int,
    cpuSpeed                   :: cisfloat,
    cpuLoad1                   :: cisfloat ::single,
    cpuLoad5                   :: cisfloat ::single,
    cpuLoad15                  :: cisfloat ::single,
    cpuLoadModified            :: cisdate  ::single,
  }
}
```

[1]

Attributes

cpuType: (cis) The of the computer processor (Pentium, Sparc, RS6000, MIPS, ...)

fpuType: (cis) the type of the floating point processor

cpuCount: (cisfloat) the number of CPU's in the compute resource

cpuSpeed: (cisfloat) the clockrate of the CPU's in MHz

cpuLoad1: (cisfloat) the load average in the last minute On UNIX System the cpuload is acquired for example via the uptime command.

cpuLoad5: (cisfloat) the load average in the last five minutes

cpuLoad15: (cisfloat) the load average in the last fifteen minutes

cpuLoadModified: (cisdate) the time at which the load averages was last modified. It is important to note that the concept of a ttl does not really apply

[1] specify what the units are.

very well for the CPU load attributes. On many systems the update of the information regarding the cpuload for one minute will cause the update for the other fields as well.

11 Grid::SystemDynamicInformation

This object is planned to be replaced by the software daemon. For portability reasons it is left in this document. There is no information about the usage of this object available at the present time.

Grid::SystemDynamicInform

```
Grid::SystemDynamicInformation AGGREGATE-CLASS ::= {
  MAY CONTAIN {
    heartBeat ::cis,
    bootTime :: cis,
    numberOfInteractiveUsers :: cis #numericStringSyntax
  }
}
```

Attributes

heartBeat : (cis) the last time the resource was known to be alive

bootTime : (cis) the last time the resource was known to be rebooted

numberOfInteractiveUsers : (int) How does this differ (if at all) from numberOfBatchUsers, etc.

12 Grid::TestBed

Grid::TestBed

The testbed object allows to create a DIT subrtree which is typically created during experiments for example at Supercomputing or HPDC. An organization could host its own testbed to distinguish it from the normal operational services.
To distinguish such temporary resources we also recommend to include the "object-class: testbed" in each of its objects. This allows to restrict searches to exclude testbed information. Ideally one would maintain such a testbed on a separate LDAP server.

```
Grid::TestBed OBJECT-CLASS ::= {
  SUBCLASS OF { GlobusTop }
  RDN = tn (testbed)
  CHILD OF {
    Grid::Top
  }
  MAY CONTAIN {
    description        :: cis,
    mail               :: cis,
    administrator      :: cis,
    adminPassword      :: cis,
  }
}
```

Attributes

description: (cis) the description of the testbed, its purpose and a pointer to a url (the url should be in its own field)

mail: (cis) the mail address for the testbed, this is a broadcast address

administrator: (cis) the administrator of the testbed

adminPassword: (cis) the administrator testbed password the password is only read and writable by the administrator

References

Index

administrator, 9
adminPassword, 9
Attribute
    administrator, 9
    adminPassword, 9
    bootTime , 8
    canonicalSystemName, 4
    cpuCount, 7
    cpuLoad1, 7
    cpuLoad15, 7
    cpuLoad5, 7
    cpuLoadModified, 7
    cpuSpeed, 7
    cpuType, 7
    description, 3, 8
    documentation, 3
    fpuType, 7
    freePhysicalMemory , 6
    freeSwapSpace, 6
    freeVirtualMemory, 6
    heartBeat , 8
    hostID , 4
    imageObject, 3
    imageOf, 3
    lapack100, 7
    lapack1000 , 7
    lapack500, 7
    machineHardwareName, 4
    mail, 9
    manufacturer, 4
    mflops , 7
    model , 4
    numberOfInteractiveUsers , 8
    OSName , 5
    OSRelease, 5
    OSType , 5
    OSVersion, 5
    pageFaultRate, 6
    physicalMemoryAccessTime , 6
    physicalMemorySize , 6
    resourceAdministrator, 3
    resourceOwner, 3
    resourceProvider, 3
    serialNumber , 4
    SPECfloat92, 7
    SPECint92, 7
    totalDataCache , 6
    totalInstructionCache, 6
    totalSwapSpace , 6
    type, 3
    type , 4
    url, 3
    virtualMemorySize, 6

bootTime , 8

canonicalSystemName, 4
cpuCount, 7
cpuLoad1, 7
cpuLoad15, 7
cpuLoad5, 7
cpuLoadModified, 7
cpuSpeed, 7
cpuType, 7

description, 3, 8
documentation, 3

fpuType, 7
freePhysicalMemory , 6
freeSwapSpace, 6
freeVirtualMemory, 6

Grid::BenchmarkInformation, 6
Grid::CacheInformation, 6
Grid::ComputeResource, 4
Grid::CpuInformation, 7
Grid::ImageResource, 3
Grid::MemoryInformation, 5
Grid::OperatingSystemInformation, 5
Grid::PhysicalResource, 2
Grid::SystemDynamicInformation, 8
Grid::TestBed, 8

heartBeat , 8
hostID , 4

imageObject, 3
imageOf, 3

lapack100, 7
lapack1000 , 7
lapack500, 7

machineHardwareName, 4
mail, 9
manufacturer, 4
mflops , 7
model , 4

numberOfInteractiveUsers , 8

Objectclass
    Grid::BenchmarkInformation, 6

Grid::CacheInformation, 6
Grid::ComputeResource, 4
Grid::CpuInformation, 7
Grid::ImageResource, 3
Grid::MemoryInformation, 5
Grid::OperatingSystemInformation, 5
Grid::PhysicalResource, 2
Grid::SystemDynamicInformation, 8
Grid::TestBed, 8
OSName , 5
OSRelease, 5
OSType , 5
OSVersion, 5

pageFaultRate, 6
physicalMemoryAccessTime , 6
physicalMemorySize , 6

resourceAdministrator, 3
resourceOwner, 3
resourceProvider, 3

serialNumber , 4
SPECfloat92, 7
SPECint92, 7

totalDataCache , 6
totalInstructionCache, 6
totalSwapSpace , 6
type, 3
type , 4

url, 3

virtualMemorySize, 6