FBSNG v1.2 Resources

Introduction

FBSNG implements the concept of *abstract resources*. The idea behind it is that each individual node of the farm, and the farm as a whole host certain amount of resources. They are called *abstract* because FBSNG does not have any knowledge of what those resources are. Each resource has its name and integer capacity. Each job section or individual process allocates certain amount of resources when it starts, and releases the same amount when it finishes. For example, one can introduce resources like "cpu" and "disk", and even though they mean "CPU power utilization in percents on the node" and "amount of local disk in Gigabytes", which are quite different from user's point of view, FBSNG treats them in exactly the same way.

This document describes the concept of FBSNG resource management.

Resource Classification

FBSNG resources are classified as the following:

- Global resources
- Local resources
- Node attributes

Global Resources

Global resource is such resource that is hosted by the farm as a whole and cannot be associated with any node. Processes share the same global resource regardless of what node they are running on. Two examples of global resources are:

- bandwidth of local farm network
- NFS-shared disk space available on all or some farm nodes.

Local Resources

Local resources are attached to individual nodes. Processes running on different farm nodes do not share the same local resources. Examples of local resources are

- CPU power of individual farm node
- farm node scratch disk space.

Node Attributes

Node attribute is a special kind of local resource. One can think of an attribute as a local resource with unlimited capacity. Processes do not allocate any amount of such resource while they are running. They only require that the node they are running on has this attribute. Examples of attributes are:

- Special software installed on certain nodes (e.g. "encp")
- CPU type (e.g. "Pentium2" or "Pentium3")
- Operating system flavor (e.g. "IRIX", "Linux")
- Color or size of computer's case ("blue" or "mini-tower")

Attributes can be used to logically partition a farm into sub-farms. One can define an attribute like "sub-farm-1" and "sub-farm-2" and split the farm into two (possibly overlapping) parts attaching one or two (or even none) of these attributes to each node of the farm.

Resource Pools

Similar resources can be combined into *resource pools*. With respect to a pool, the resources it consists of are referred to as *underlying resources.* Resource pool allows FBSNG batch job to request certain amount of whatever underlying resource of the pool is available at the moment. Resource pools can be used in the following situations:

- Certain farm nodes have more than one local scratch disk. Local resources representing local disks may be combined into a pool so that batch job may request some space on whichever disk is available.
- Different but compatible versions of the operating system are installed on different farm nodes (e.g. Linux 5.x and 6.x). In such case, node attributes corresponding to different versions of OS (Linux5 and Linux6) can be combined into a pool, which would represent common OS flavor (Linux). Those batch jobs that can run on both versions then may use the pool name (Linux) for their resource requirements, while jobs that require specific OS versions will use corresponding resource name (Linux6, for example).

Currently, certain restrictions apply to resource pool configuration:

- Pools can not contain other pools
- Pools can not contain resources of different types, for example, global and local resources can not be mixed into the same pool. Attributes and other local resources are considered to be different types of resources.

Resource Allocation Levels

A batch job can allocate necessary resources on two levels.

Allocation on Process Level

User can request certain resources to be allocated to each individual process of the job section. In this case each process will consume the same amount of specified resources. For example, user can specify that each individual process will consume 85% (85 units in terms of abstract resources) of CPU power (of abstract resource named "cpu") on the node where the process runs. All types of resources (local, global and attributes) can be allocated at this level. Obviously, total resource allocation for the section will be proportional to number of processes of the section. All local resources allocated to the process are de-allocated when the process finishes.

Section Level Allocation

User can request resources to be allocated to the section as a whole. Since section itself does not run on any particular node, only global resources can be allocated on this level. For example, one can say that no matter how many processes form the section, the section produces one tape worth of data, and therefore requires fixed amount of space on NFS-shared disk. In this case, one would request certain amount of the global resource associated with the disk space on NFS server (e.g. "nfs_disk") to be allocated to the section itself, not to individual nodes. In this case, the resource will remain allocated until last process of the section finishes.

Resource Requirements Specification

FBSNG determines process and section resource requirements based on FBSNG configuration and Job Description File (JDF). One of JDF fields user must specify for

each section of the job is the name of the queue the section is to be submitted to. In FBSNG configuration, each queue is associated with *default process type*. Unless overridden in JDF, the queue default process type becomes the section process type. Process type should be used to define common resource requirements for similar batch processes of the same project or task. Minimal common resource requirements for processes of the same type are defined in FBSNG configuration as the process type *default resource requirements*. JDF fields SECT_RESOURCES and PROC_RESOURCES can be used to request resources *in addition* to the defaults. In case when certain resource is mentioned both in process type default requirements in FBSNG configuration, and in JDF, the greater amount of the resource will be used. PROC_TYPE field of JDF can be used to override default process type associated with the queue.

For example, let us assume that FBSNG configuration defines two queues, "LongQ" and "ShortQ", both with default process type "Worker":

```
%set queue LongQ
default_proc_type = Worker
…

%set queue ShortQ
default_proc_type = Worker
…
```

Two process types are defined in the configuration:

- Type "Worker" requires 100 units of CPU per process;
- Processes of type "IO" must run on nodes with attribute "IO_node" and require 50 units of "CPU" and 1 unit of local disk:

```
%set proc_type Worker
default_proc_resources = cpu:100
…

%set proc_type IO
default_proc_resources = cpu:50 disk:1 IO_node
…
```

JDF consists of two sections "Prestage" and "Run":

```
SECTION Prestage
QUEUE = ShortQ
PROC_TYPE = IO
PROC_RESOURCES = disk:10
…

SECTION Run
QUEUE = LongQ
PROC_RESOURCES = nfs_disk:5 cpu:1 encp
…
```

As you can see, JDF overrides default process type for section "Prestage" from "Worker" (default for "ShortQ" queue) to "IO". Section "Run" assumes processes type "Worker", default for queue "LongQ".

Each process of "Prestage" section will consume:

- 50 units of CPU (default resources for "IO" process type) and
- 10 units of disk (the greater of process type default and JDF specification)
- Only nodes with attribute "IO_node" will be chosen for processes of this section (default for "IO" process type)

Processes of section "Run" will consume:

- 100 units of CPU (even though user requested only 1 unit of CPU in JDF, the greater of process type default requirements for this resource and what is specified in JDF will be used)
- 1 units of disk (process type default)
- 5 units of nfs_disk (additional resource requirement specified in JDF)
- Will run only on nodes with "encp" attribute (additional JDF requirement)

Local Scratch Disks

Although FBSNG resources are considered to be "abstract", i.e. FBSNG does not have any knowledge of what they represent, scratch disk space available on farm nodes (local disk space) is an exception. Certain components of FBSNG recognize resources representing local disk space and treat them in a special way. When a batch process requests certain amount of disk space, FBSNG creates scratch directories on corresponding local disks. Then the batch process receives pointers to such directories from UNIX environment variables. If a batch process is not going to use any considerable amount of local scratch disk space, and therefore does not request any space on local disks, FBSNG creates a scratch directory for the process anyway without allocating any space there.