DE LA RECHERCHE À L'INDUSTRIE

Slurm Layous Framework

www.cea.fr

June 2015 | CEA/DAM DIF

Slurm Layouts Framework
Outline

Layouts and Entities

Entities attributes

Implementation details

Slurm Layouts Framework

Layouts and Entities

Slurm Layouts Framework
Layouts and entities

Framework based on 2 distinct notions :

Entity

- **Anything can be described as an entity, it is just :**

  - A name
  - A type

Layout

- **A generic representation of the arrangement of components and their properties associated to an aspect of the system**
  - Different arrangements : hierarchical (tree), [...]
  - Different aspects : racking, power supply, power consupmption, ...

Slurm Layouts Framework
Layouts and entities

Layouts provide information to entities by

- Linking them according to a particular logic (tree, [...])
- Adding them a set of attributes (Key/Value pairs)
- Defining their own internal entities to better fulfill their purposes
- keeping consistency among attributes values across entities based on keys inheritance relations

Layouts provide transversal discoveries

- Entities may be associated to multiple layouts
- Starting from an entity and a layout, neighborhood can be discovered easily

Slurm Layouts Framework

Entities attributes

Slurm Layouts Framework
Entities attributes

Attributes management

- **Attributes are « described » in layouts implementations**
- **They are automatically handled by the Framework, meaning :**
  - The parsing of layouts configuration file is automated
  - Values associated to Keys are automatically created using the associated types
  - Key/Value pairs are automatically integrated in their corresponding entities
  - Read-Only vs Read-Write values (to avoid update of some static states)

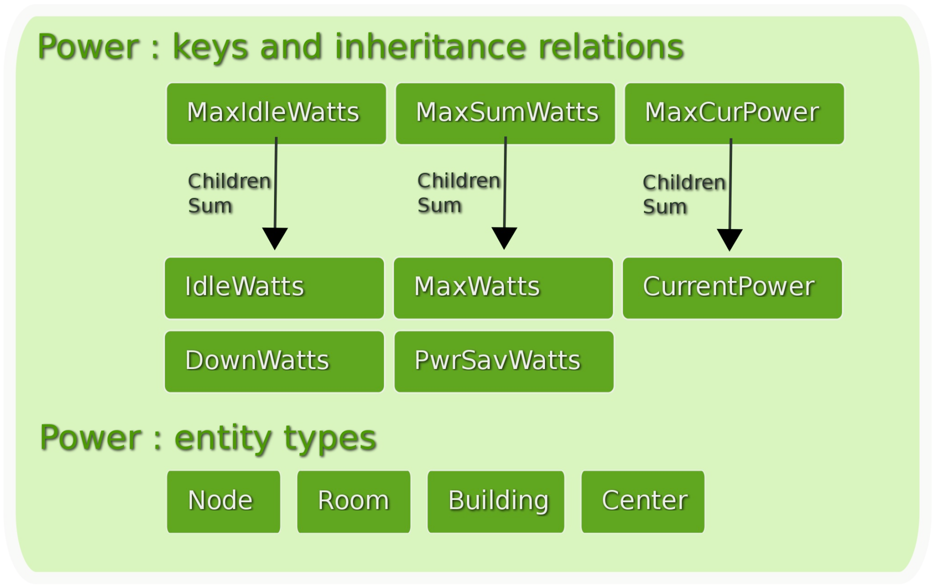
Valid types

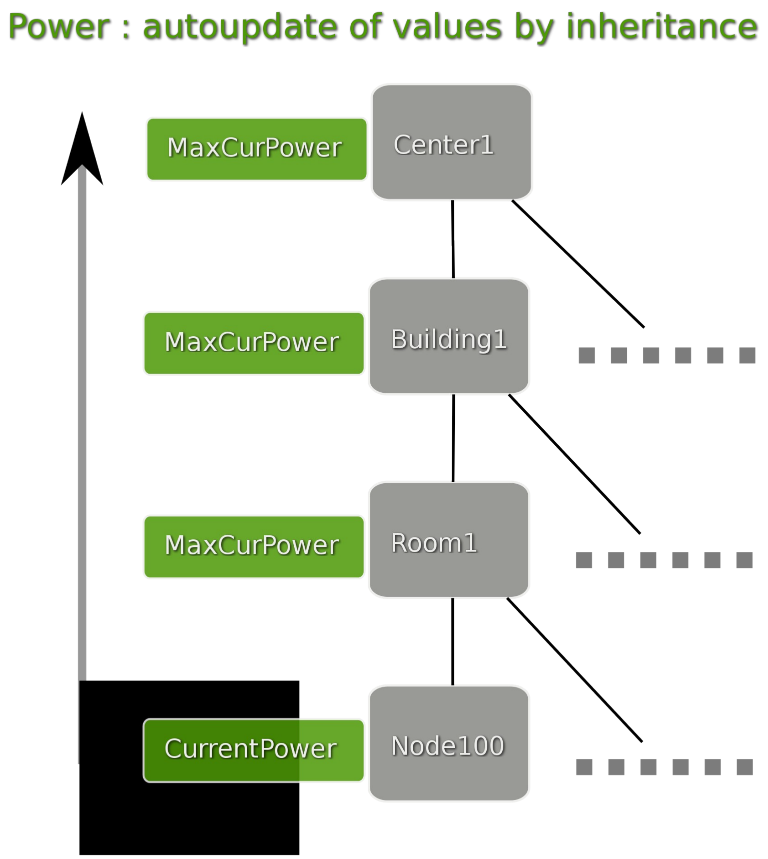
- **String (char*), Boolean, uint16_t, uint32_t**
  float, double, long double, « custom »

Attributes inheritance

- **Inheritance enables to ease consistency and usage of the attributes by :**
  - Letting the framework automatically manage counter-effect of updates
  - Letting the developpers focus on dealing with set/get + walk through the K/V pairs of the different entities/layouts, limiting his direct access to the underlying system
- **Inheritance types are currently (tree relation model)**
  - Children count, min value, max value, average value, sum
  - Parent[s] min value, [max value, average value, sum,] fair share
  - Note that only a single parent is supported for now because of the tree relation model

Slurm Layouts Framework
Entities attributes





Slurm Layouts Framework

Implementation details

June 2015 | CEA/DAM DIF

Layouts Framework in Slurm (15.08)

- **« Base » layout, default layout defining :**
  - all the compute nodes as entities with an opaque reference to the Slurm node pointer associated
  - A single root to create the basic flat tree of the available nodes
- **« Layouts » slurm.conf parameter :**
  - To specify the layouts to load at startup
- **« layouts.d/ » directory to store layouts conf file :**
  - Exp : « /etc/slurm/layouts.d/power.conf »
- **Layouts states dump to state files like other Slurm states :**
  - Dumped as classic but expanded layouts configuration files

Layouts Framework in Slurm

- **Available layouts :**
  - Base : embedded layout
  - Unit : unit testing oriented layout, enabling to validate all the internal logic
  - Power : layout dedicated to power consumption information, used to aggregated power consumption of the system and apply power capping to the system
    - Integrated by Bull, based on CEA power capping prototype with Slurm
  - More in the future ? Topology, Racking,

Layouts Framework in Slurm

- **Current API :**
  - layouts_entity_get_kv_type(char* layout, char* entity, char* key)
  - layouts_entity_get_kv_flags(char* layout, char* entity, char* key)

  - layouts_entity_set_kv(char* layout, char* entity, char* key, ...)
  - layouts_entity_set_kv_ref(char* layout, char* entity, char* key, ...)
  - layouts_entity_get_kv(char* layout, char* entity, char* key, ...)
  - layouts_entity_get_kv_ref(char* layout, char* entity, char* key, ...)

  - layouts_entity_push_kv(char* layout, char* entity, char* key)
  - layouts_entity_pull_kv(char* layout, char* entity, char* key)

  - layouts_entity_setpush_kv[_ref] (...)
  - layouts_entity_pullget_kv[_ref] (...)
  - ...
  - layouts_entity_get_mkv[_ref] (...)