Towards multi-objective resource selection

Dineshkumar Rajagopal, David Glesser, Yiannis Georgiou

16-09-2015
Motivations

- Evolution of HPC platforms architecture makes resource management more complex than before.
- Managing and selecting resources based upon various criteria/objectives will enable the RJMS to become more adaptive.
- Study upon SLURM to provide a flexible and easy to use architecture for multi-objective resource selection based on the layouts framework.
Multi-objective scheduling research

- MOEBUS Research Project (http://moebus.gforge.inria.fr/)
  - 4 years ANR (French funded) project started October 2013
- Study the design of multi-objective optimization algorithms for some combinations of objectives (performance, fairness, energy consumption, etc.).
- Find the right balance between theoretical analysis and practical implementation.
Multi-objective scheduling research

- Multi-objective problem is an optimization problem, and not really a decision one.
- The central problem is the trade-off between all feasible solutions. There are a lot of "good" solutions.

**Definition:**
A solution is Pareto optimal if no solution is as good as it is for all the objectives and is better for at least one objective.
Pareto optimal points and curve
Pareto optimal points and curve
What is the size of the optimum of an optimization problem?

- Single objective problems: usually, only one value of the solution
- Multi-objective problems: exponential number of solutions (or even infinite number of solutions).
Resource Selection within SLURM

- Internal representation based on bitmaps
- Node-bitmap and core-bitmap used under different contexts for scalability purposes
- Based mainly upon select plugin (i.e. linear, cons_res) in conjunction with plugins such as gres, topology, etc
Enhance Resource Selection

**Proposal:** Take advantage of the flexibility of the layouts framework to extend resource selection towards multi-objective

- Map bitmaps to layouts for resources availabilities
- Layouts may provide additional details for the resources (such as power consumption, data locality, temperature, racking, etc)
Prototype:
Map resource availabilities using 2 layouts
Prototype:
Map resource availabilities using 2 layouts

Network Topology
Available Sum Cpus, Available Sum Nodes

Node Architecture
AllocatedSumCores, NumSumCores, AllocatedNodes

Entity=node0
Prototype:
Extend resource availabilities with power consumption details using 3 layouts
Prototype:
Extend resource availabilities with power consumption details using 3 layouts

- **Network Topology**
  Available Sum Cpus, Available Sum Nodes

- **Topology Power**
  Max sum Power, Idle Sum Power, Max Power Sum Nodes, Idle Power Sum Nodes

- **Node Architecture**
  AllocatedSumCores, NumsSumCores, AllocatedNodes

Entity=node0
Consider a use case of heterogeneous architecture with different types of nodes and power consumptions (homogeneous under a certain level of switches).

Goal is to favor the low power consumption nodes, competing with availability and topology aware scheduling.
Experimentations

Power consumption comparison of 2 policies cons_res and cons_res_power

Type of collection – Calculated Energy Consumption
- cons_res - 1391678750 Joules
- cons_res_power - 1340426380 Joules

Time (sec)
Ongoing Works

▶ Evaluating and enhancing our prototype has not been finished
  - Scalability issues: more important than when using just one layout
  - Need Layouts API functions to change key/values across different layouts for the same entity
  - Trying to design a solution that would be easy to add other parameters/constraints such as temperature data locality, racking, etc

▶ Need of algorithms from theoretical research to provide good optimizations

▶ Practical solution for now provide parameters within SLURM (administrator or user side) to assign factors on each objective
Thanks