University of Manitoba

Computer Science Department
Advanced Networking Research Laboratory

Hierarchical, Container-based Grid Resource Management

Rasit Eskicioglu, Muthucumaru Maheswaran, and Peter Graham

Abstract

Resource management is a fundamental problem in grid-based systems. Providers of computational service want to be sure that grid users do not use more resources than they should, users want to be sure that they get the resources they want when they need them, and both parties are concerned about accurate resource accounting. In this poster, we present a new architecture for performing resource management in grids based on "resource containers". Our architecture generalizes previous work done on distributing resource containers to provide a flexible and extensible framework for grid resource management.

The Problem

An important requirement of grid systems is effective resource handling, which involves:

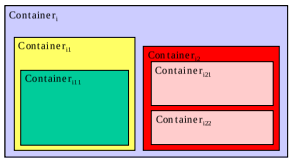
(i) *Resource discovery* - the location of available resources in the grid environment,
(ii) *Resource allocation* - the assignment of discovered resources to specific applications, and
(iii) *Resource management* - which oversees the use of allocated resources to verify that resource allocations are not exceeded, to ensure that applications can use the allocated resources to meet their execution requirements, and to record information on actual resource usage.

Grid Resource Management Requirements

- *Distribution* - an obvious requirement for grids
- *Autonomy of systems* - cannot trust remote jobs to not abuse unmanaged resources
- *Flexibility* - must be capable of supporting as yet unforseen resource use patterns
- *Accuracy* - in control and accounting
- *Cooperation* - between hosts in the grid to enable other functionalities, such as co-scheduling.
- *Performance* - minimize overhead

Containers - The Concept

- Containers [1] are general units of resource management that are not tied to processes or other scheduling units
- Developed to allow for accurate accounting and control of resource use in server systems
  - where resource management should be tied to applications, not a single execution unit.
- Sounds like they might make sense in grids
- It is possible to allow containers to be allocated from within other containers
  - resulting a "hierarchy" of containers, such as the one shown below:



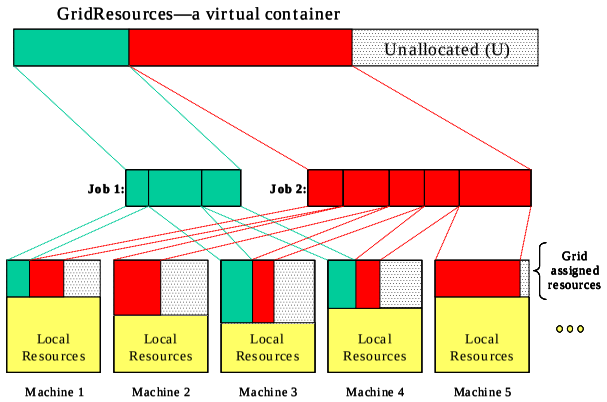
Containers - Use in Grids

- Containers, as defined by Banga, et al. [1] are local to a single machine
- They must be extended to operate in a distributed environment to be useful in grids
- Aron, et al. [2] extended containers as "cluster reserves" for use in a cluster-based system
- Vahdat [3] first proposed containers for use in grids (focused on security issues)

Grid Containers - the Model

- Each grid node creates a *physical* (node-level) container that captures the resources it will make available to grid
- A grid job will use a subset of these resources on multiple machines
  - exploits nesting of containers
- *Virtual* (grid-level) containers permit management of grid resources
  - at multiple levels, if appropriate (e.g., hierarchical jobs)

Grid Containers - an Example



Grid Containers - Some Details

- Each *physical* container encapsulates a set of resources, such as compute, memory, and storage
  - may be nested
- Each *virtual* container represents a collection of resources provided by a set of physical containers
  - directly or indirectly (through other virtual containers)
- Physical container management
  - provided by host operating systems (OSes)
  - grid applications interact with the host OS through a local application (e.g., a grid resource manager daemon) that controls the grid-assigned resources
- Virtual container management
  - one or more levels of management for grid jobs
    - request resources be added or deleted to grid jobs by lower level resource managers (ultimately, node-local grid resource managers)

Achieving Requirements

- *Distribution* - via virtual containers
- *Autonomy* - nodes control what resources are given to the grid
- *Flexibility* - arbitrary hierachy and application-specific algorithms for virtual containers
- *Accuracy* - provided by containers
- *Cooperation* - provided by flexible algorithms with autonomy contraints
- *Performance* - implementation dependent, but communication is minimal

Conclusions

- Distributed containers offer promise for grid management
  - meet all the stated requirements
- Compatible with other grid algorithms, such as resource discovery
- Can be used to help provide advanced functionality
  - co-scheduling for HPC applications
  - QoS guarantees

Related Work

[1] G. Banga, P. Drushnel, and J. C. Mogul. Resource Containers: A New Facility for Resource Management in Server Systems. *In Proceedings of the 3rd USENIX Symposium on Operating Systems Design and Implementation (OSDI'99)*, pp 45-58, Januaury, 1999.

[2] M. Aron, P. Drushnel, and W. Zwaenepoel. Cluster Reserves: A Mechanism for Resource Management in Cluster-based Network Servers. *In Proceedings of the International Conference on Measurement and Modelling of Computer Systems (SIGMETRICS2000)*, pages 90-101, June, 2000.

[3] A. Vahdat. Toward Wide-Area Resource Allocation. *In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, pages 930-936, June, 1999.

Contract Information

Rasit Eskicioglu
University of Manitoba
Computer Science Department
Winnipeg, Manitoba R3T 2N2 CANADA
rasit@cs.umanitoba.ca