Dynamic Balancing of Scientific HPC Applications on Institutional Clusters

HAVERFORD COLLEGE

Mac Lyle

Demands on computing resources are greater than the existing supply. Proposed methods allow existing computing resources to complete more large-scale jobs in a given timeframe.

Limited Resources

- Demand on current computing resources in the scientific community is growing rapidly.
- We are reaching the limit of Moore's law and can't wait for faster machines to solve our problems.
- Need to implement more efficient ways of running more jobs on the hardware that we already have in place.

Why Not Cloud?

Current limitations of the cloud:

- Speed severely limited by network latency.
- Current clouds services aren't set up for HPC workflows.
- High level of cloud administrator knowledge needed.
- Fewer tools for test environments.
- Currently incapable of replicating the performance of institutional clusters.

Benefits of using a cloud system:

- Cheap, elastic, and widely available.
- Viable option for smaller workflows.

Methods

Dynamic Right-Sizing Master-Worker applications

Goal: Dynamically allocate the optimal number of machines to a master-worker application.

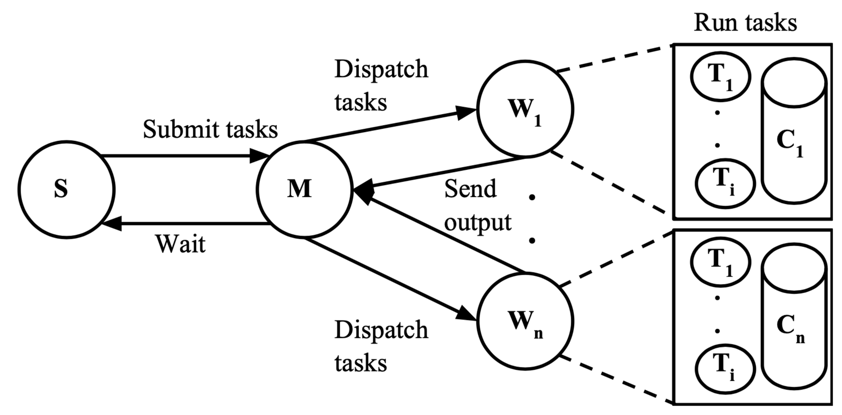
- ***Capacity*** defined as the number of computational nodes that can be effectively utilized by an application
- Correctly allocates machines to the application based on *capacity* during run time.
- Eliminates under and over provisioning of resources to applications.
  - Can increase system throughput and utilization.

Introducing on-demand requests to HPC Clusters

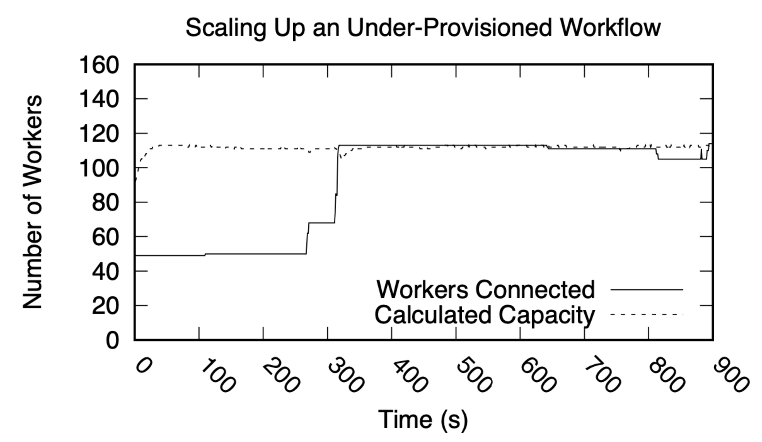
- Traditionally, HPC uses a batch scheduler and on-demand resources have designated computational resources.
- Propose to combine on-demand and batch system resources to increase total computing power.
- Running on-demand requests on a majority batch system.
- Could reduce the investment on on-demand hardware by a massive margin and still be able to handle all cases in a workflow.
- Massive reduction in batch wait times.

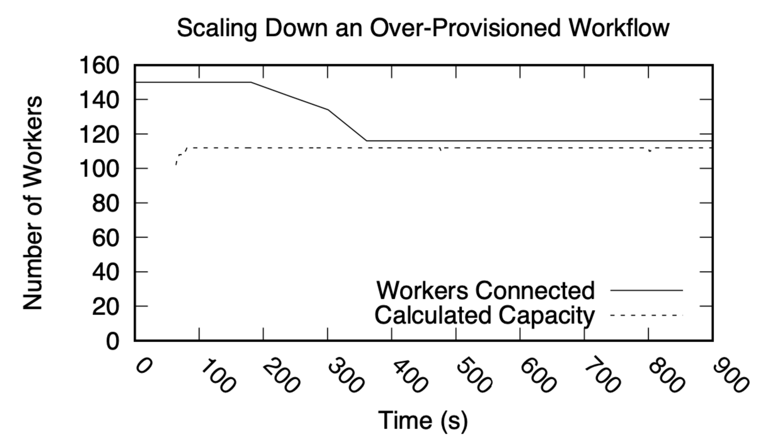
Parameter settings	Static (Baseline)			Dynamic						
	Dedicated batch nodes			W						
	372	304	0	0	5	10	0	5	10	0
	Dedicated on-demand nodes			R						
	0	68	372	0			6			12
Combined utilization	84.4%	80.1%	1.25%	84.9%	85.7%	85.7%	85.3%	85.3%	85.3%	85.3%
Batch utilization	84.4%	78.8%	NA	84.5%	84.4%	84.4%	84.0%	84.1%	84.0%	84.0%
On-demand utilization	NA	1.25%	1.25%	0.38%	1.25%	1.25%	1.25%	1.25%	1.25%	1.25%
Batch wait time (min)	122.5	1002.8	NA	122.0	147.0	147.0	150.0	140.6	150.4	130.0
Rejections	141	0	0	30	3	3	1	1	0	0

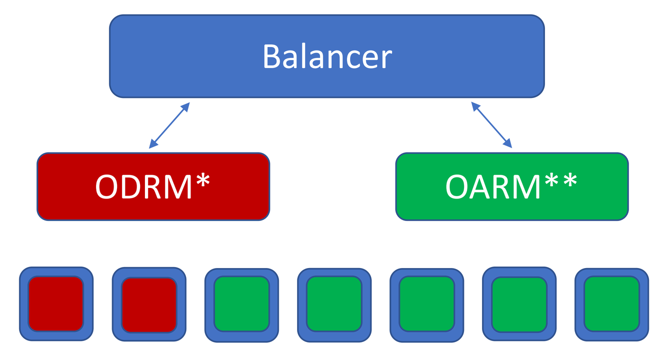
Experimental results of most challenging real-world workflow. Table from Feng Liu 2018.



Top: Master-worker architecture (Nathaniel Kremer Herman, 2018).
Left: Ability of model to scale up/down based on application *capacity* (Nathaniel Kremer Herman, 2018).
Right: Balancer architecture (Feng Liu, 2018).







*On-Demand Resource Manager
**On-Availability Resource Manager

Future Work

- Replication study of both methods using different real-world workflows. Potentially data from researchers at Haverford.
- Testing fault tolerance of the proposed methods.

References:

- Feng Liu, Kate Keahey, Pierre Riteau Jon Weissman (2018). "Dynamically Negotiating Capacity Between On-Demand and Batch Clusters". In: *SC18* (cit. on pp. ii, 5, 11–15, 19).
- Nathaniel Kremer Herman, Benjamin Tovar and Douglas Thain (2018). "A Lightweight Model for Right-Sizing Master-Worker Applications". In: *SC18* (cit. on pp. ii, 9, 10).
- Marco A.S. Netto Rodrigo N. Calheiros, Eduardo R. Rodrigues Renato L. F. Cunha Rajkumar Buyya (2018). "HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges". In: *ACM Computing Surveys* 51.1 (cit. on p. 7).

Read my entire thesis here!

Advisor: John Dougherty