Big Data Analytics on HPC Architectures

Performance and Cost

Peter Xenopoulos[1], Jamison Daniel[2], Michael Matheson[2] and Sreenivas Sukumar[2]

December 5, 2016

3rd Workshop on Advances in Software and Hardware for Big Data to Knowledge Discovery
[1]Pomona College, Claremont CA 91711
[2]Advanced Data and Workflows Group, Oak Ridge National Laboratory, Oak Ridge TN 37831

Outline

1. Introduction
2. Methods
3. Results
4. Discussion
5. Conclusion

Introduction

The Traditional Role for HPC

- For many scientific domains, simulation traditionally provides the foundation for scientific discovery.
- Popular simulation applications favor traditional HPC architectures that prioritize computational capacity.
- What are the implications of these architecture choices on *data-driven science*?

Where does HPC fit into Data-Driven Science?

- Data-driven science has been thrust into the forefront with an explosion of data from atypical sources such as sensors and social media.
- Data analytics is a multi-faceted problem, encompassing the compute, memory, storage and network layers of an architecture simultaneously.
- These types of jobs are typically left to run on cloud-computing services such as AWS, or CADES, ORNL's private cloud.
- Building local analytics clusters require large capital investment due to the sheer complexity of the hardware needed.

How do current HPC architectures fare in both cost and performance for data analytics jobs?

Methods

OLCF Infrastructure

- We benchmarked a variety of HPC hardware representative of various computing resources available at scientific institutions.
- Each architecture is connected to a Lustre filesystem and has no node-level storage.

	L2 Cache	L3 Cache	RAM
Rhea	16 x 256 KB	40 MB	DDR3
Rhea (GPU)	26 x 256 KB 8-way set associative caches	70 MB 20-way set associative shared cache	DDR4
Titan	8 x 2 MB 16-way set associative shared exclusive caches	2 x 8 MB up to 64-way set associative shared caches	DDR3
Eos	16 x 256 KB 8-way set associative caches	40 MB 20-way set associative shared cache	DDR3

Table 1: OLCF node level hardware characteristics

OLCF Computing Resources

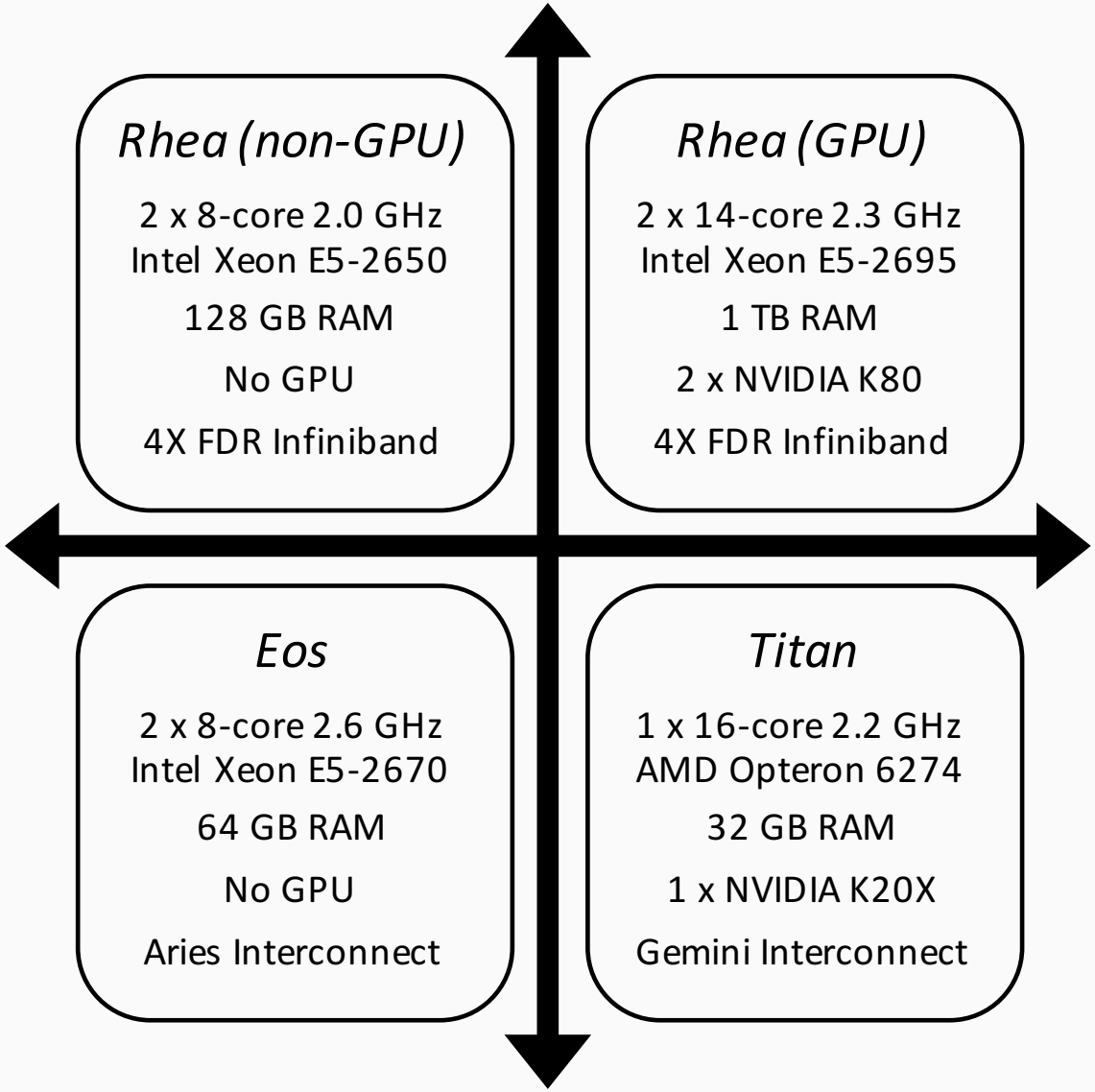


Figure 1: OLCF node level hardware

Analytics Framework

- **pbdR**, developed at ORNL, is a collection of R packages which has focused on bringing parallel R to distributed memory HPC architectures. The communication is done in MPI and pbdR contains extensions for distributed matrix operations and machine learning algorithms.

Algorithms

- Four common algorithms:
  - matrix multiplication
  - singular value decomposition
  - linear regression
  - k-means
- These algorithms represent a wide array of not only matrix operations but also iterative optimization problems commonly observed in data science.

Experimental Design

1. Test each algorithm across Titan, Eos, and both Rhea clusters
2. Benchmark performance of each algorithm on a variety of data sizes and observing scaling properties
3. Record performance for five processes: *Initialization*, *I/O*, *Blockcyclic*, *Computation* and *Finalize*
4. Derive dollar cost for each architecture

Deriving Cost

Typically, cost is defined as

$$T_A = F_A + V_A(t) \tag{1}$$

Where F_A represents the fixed cost of architecture A, which represents a one-time cost of all of the hardware components of an architecture. $V_A(t)$ is the variable cost of an architecture that depends on time. Such factors influencing variable cost are operational (labor) and electrical costs.

Architecture	Electrical Cost / hr	Operational Cost / hr	Total (Variable) Cost / hr	Hardware (Fixed) Cost
Rhea (non-GPU)	$0.0247	$0.0401	$0.0648	$3,650
Rhea (GPU)	$0.0315	$0.0401	$0.0717	$12,600
Titan	$0.0268	$0.0401	$0.0669	$1,415
Eos	$0.0326	$0.0401	$0.0727	$4,100
Amazon Web Services (On Demand)	N/A	N/A	$3.54	$0.00

Table 2: Node-level hourly costs of OLCF resources

Cost Comparison

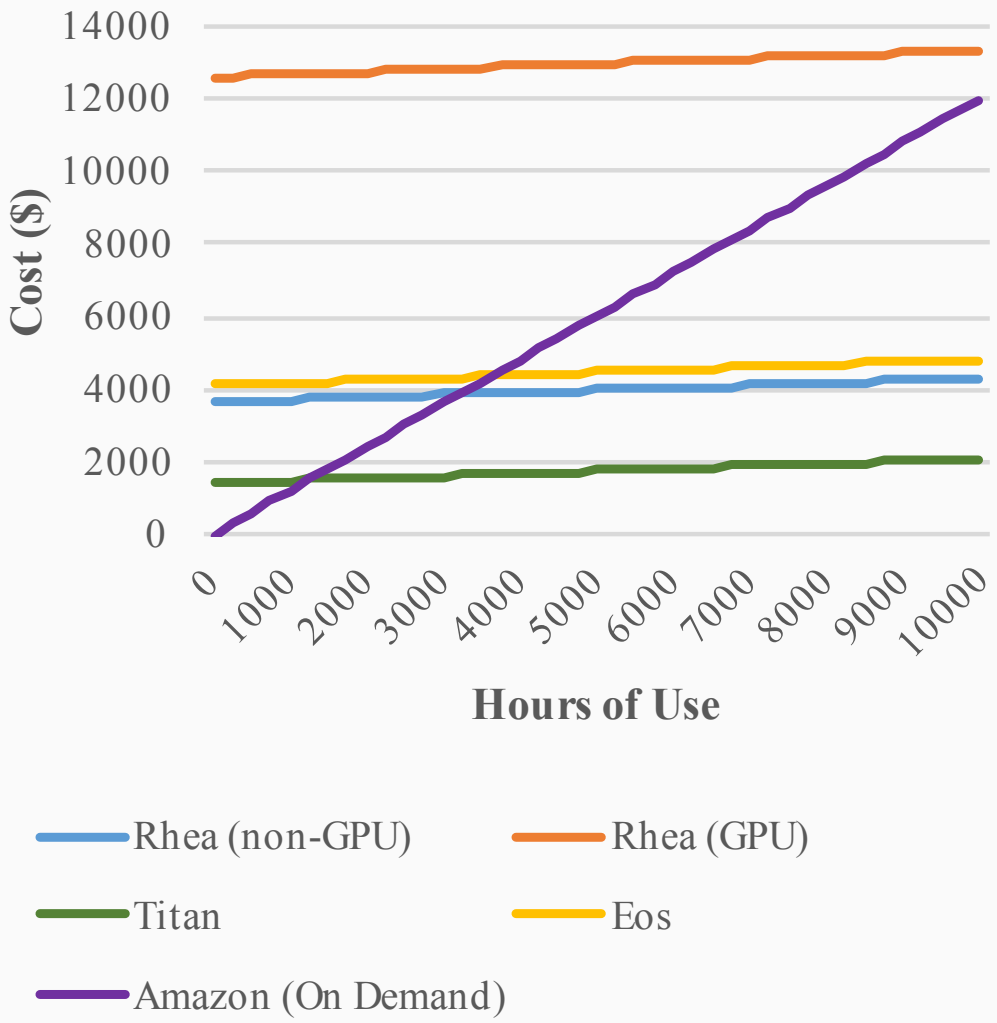


Figure 2: Cost per hour of OLCF resources vs. AWS (one node)

Results

Singular Value Decomposition Scaling Results

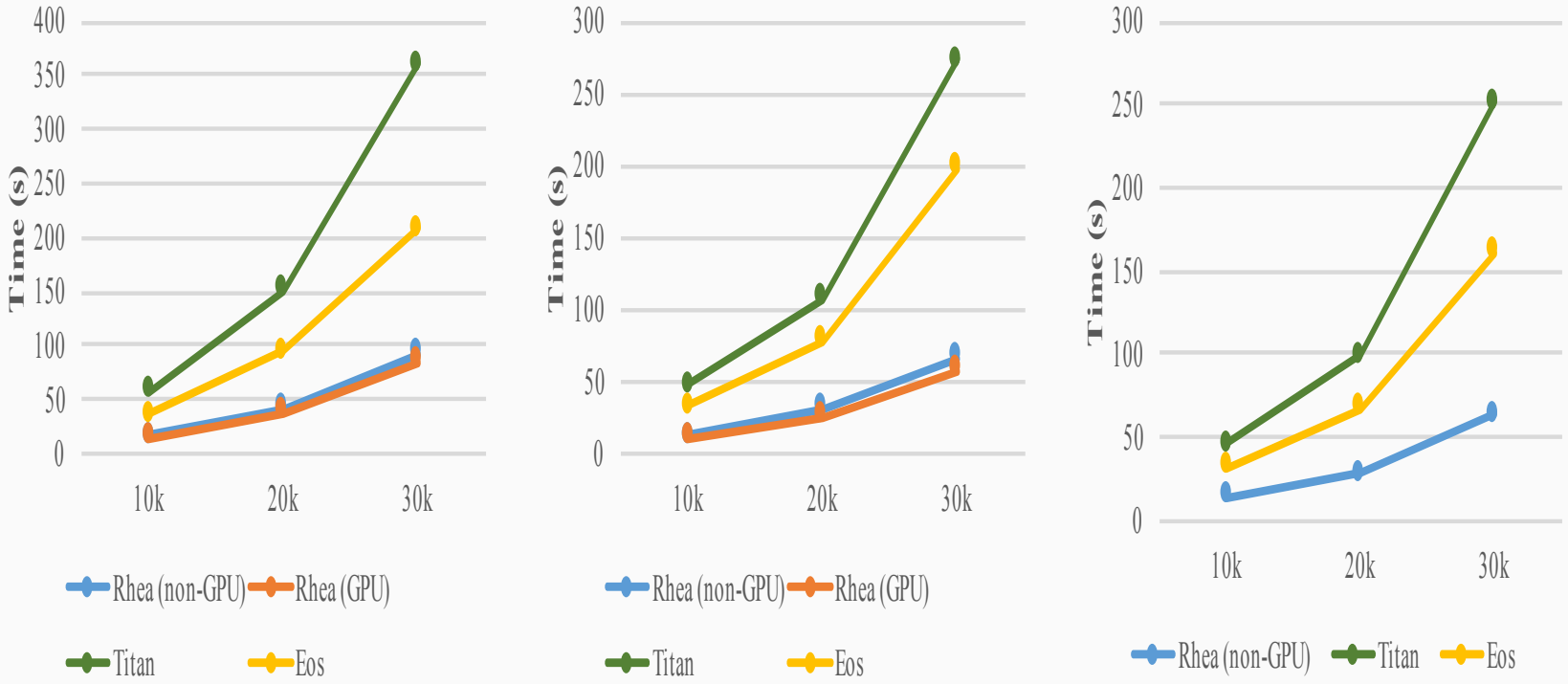


Figure 3: Benchmark run times of 1, 2 and 5 node jobs for randomized singular value decomposition by matrix size ($n \times n$)

K-means Scaling Results

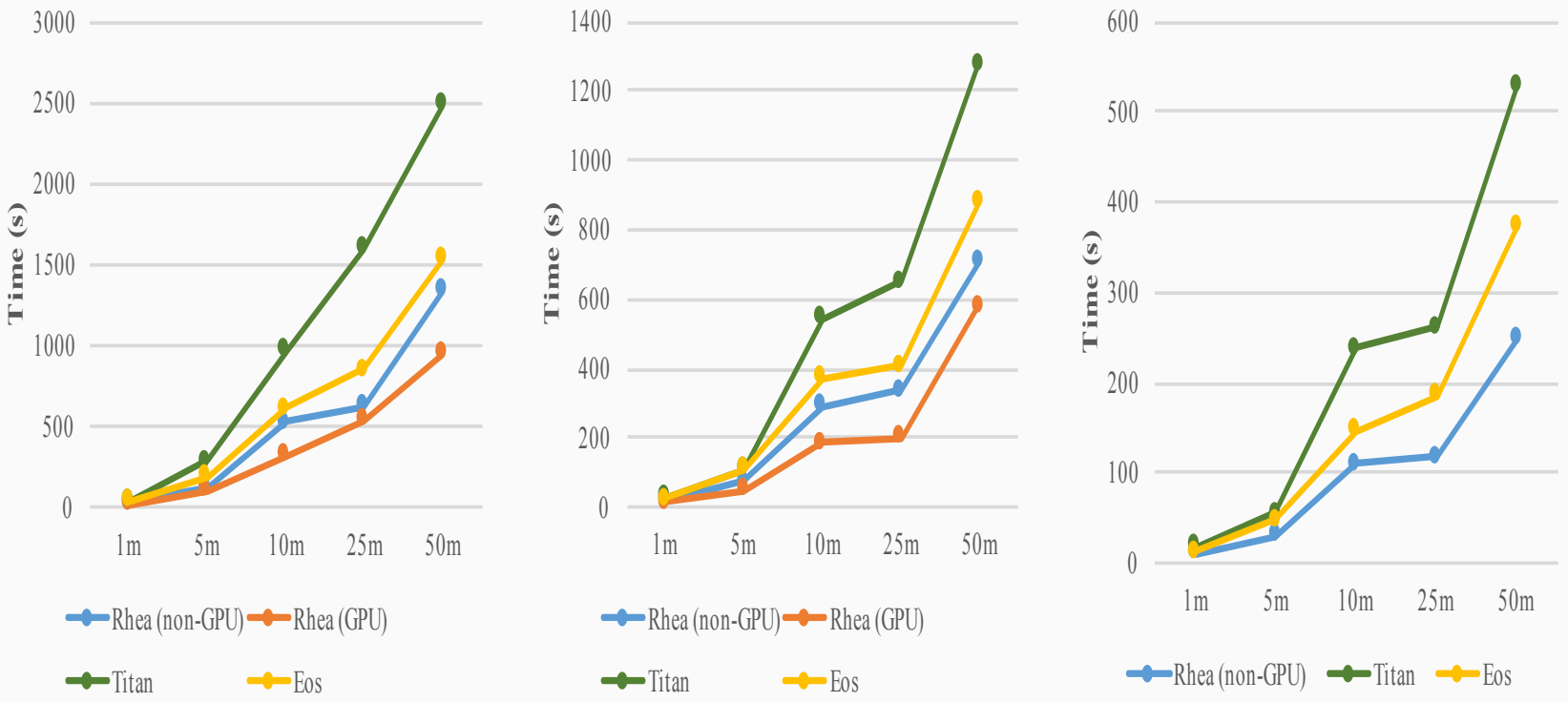


Figure 4: Benchmark run times of 1, 2 and 5 node jobs for k-means by number of observations

Where are these jobs spending their time?

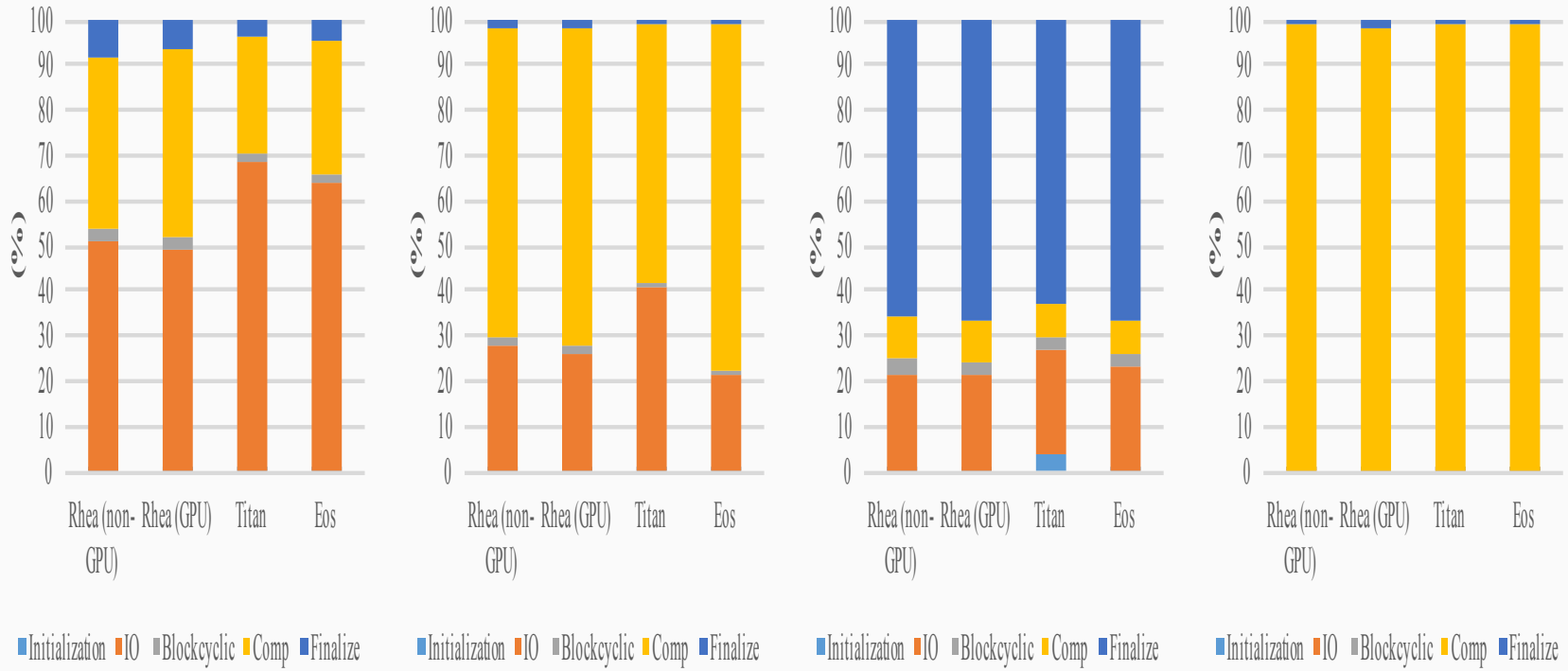


Figure 5: Breakdown of run times for a $20,000 \times 20,000$ matrix for matrix multiplication and randomized SVD and 10 million rows for linear regression and k-means

Discussion

Performance Comparisons

- Low level cache, memory bandwidth and I/O are all important considerations from a data analytics perspective.
- Big data jobs are inherently memory intensive – efficient cache usage and memory bandwidth may provide insights to performance

Low-Level Cache

- Consistent with proposed characteristics of data-analytics applications, we explore the cache-bound nature of our architectures.
- We compare two algorithms: a Monte Carlo simulation of Pi (low cache usage) and a data-science applications, K-means
- We additionally compare these simulation time results across architectures

	Rhea (non-GPU)	Rhea (GPU)	Titan	Eos
Sim. Time (s)	0.929	0.735	0.432	0.471

Table 3: Monte Carlo Pi Simulation Using 10,000 Points

Low-Level Cache

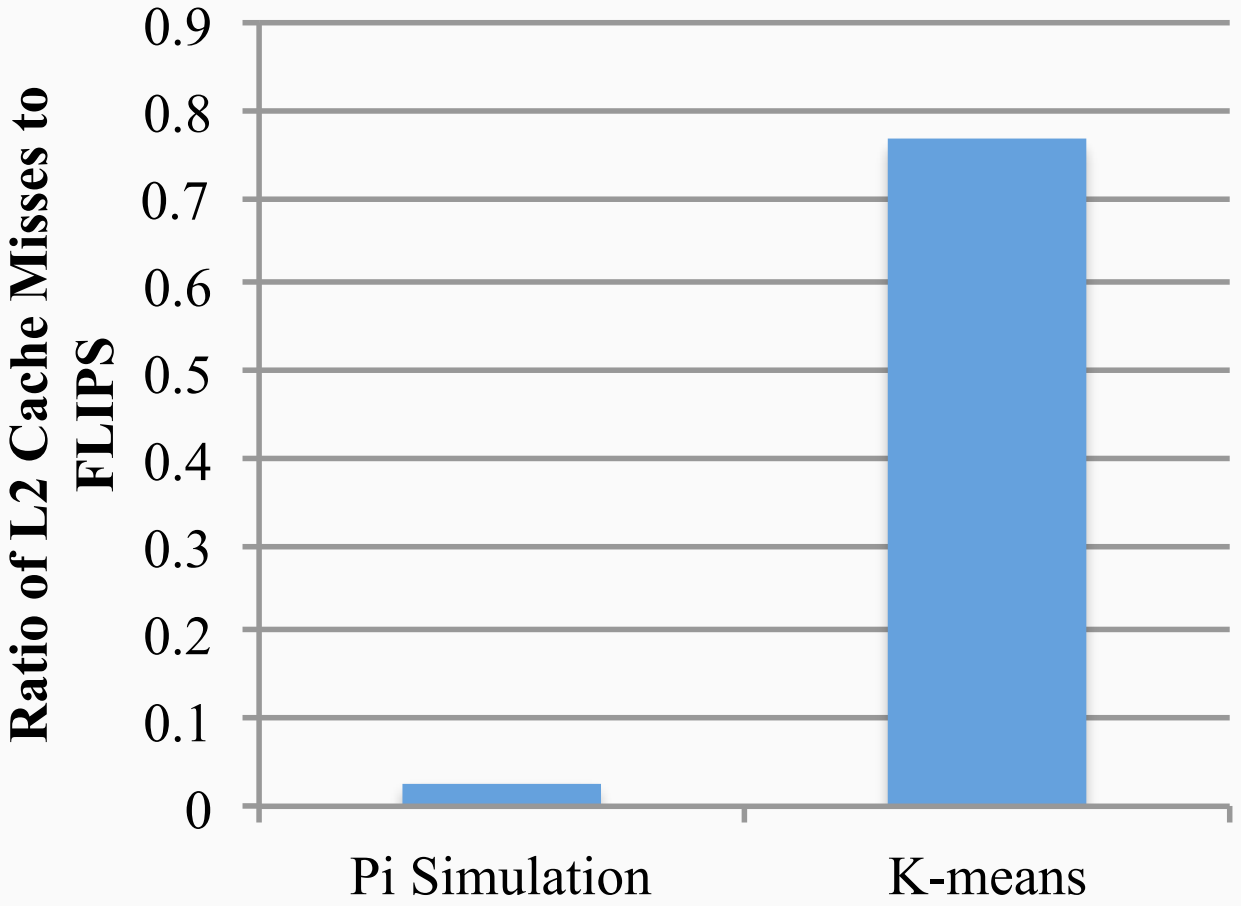


Figure 6: Cache Comparisons of Applications on Titan

Memory Bandwidth Benchmarks

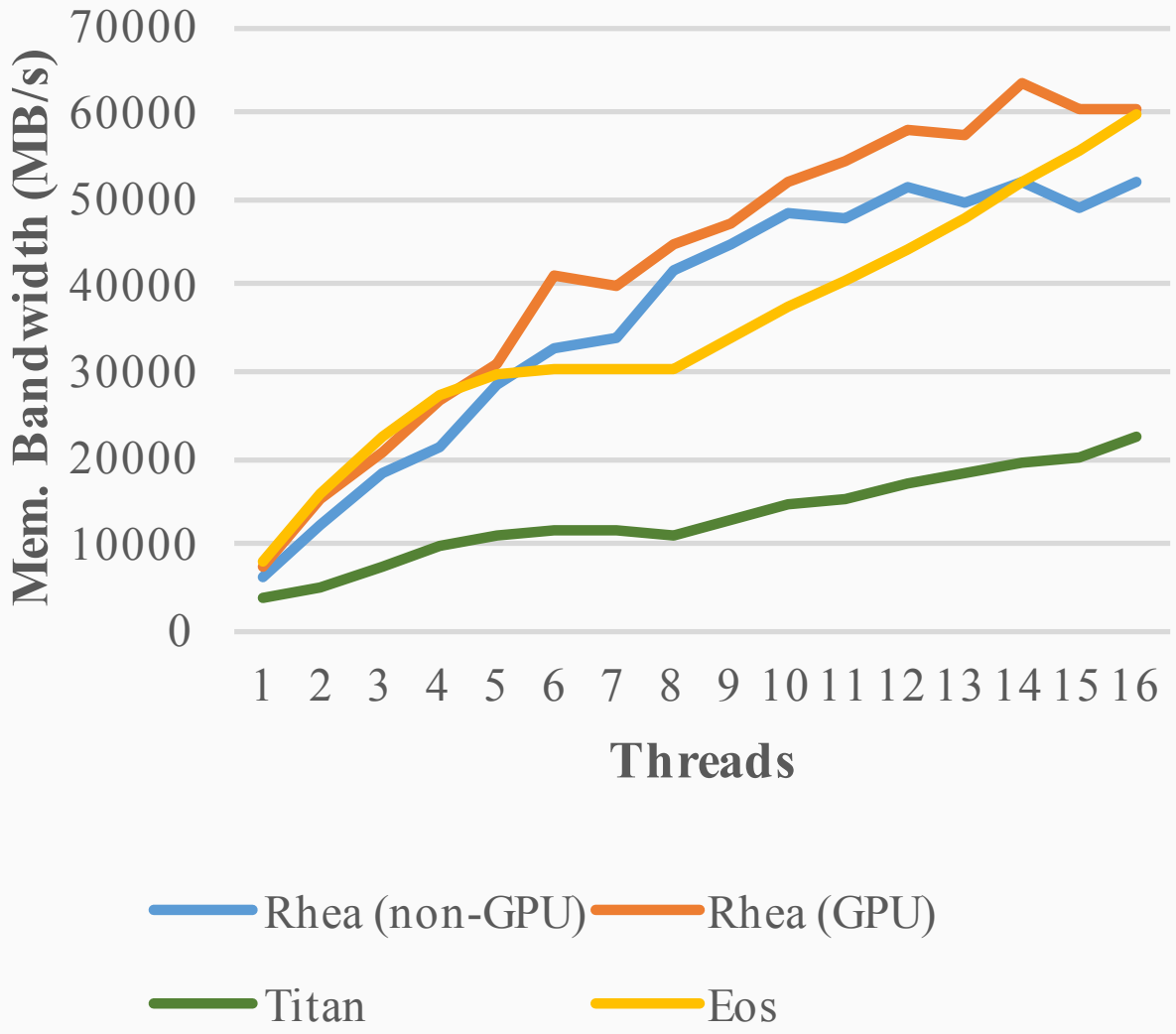
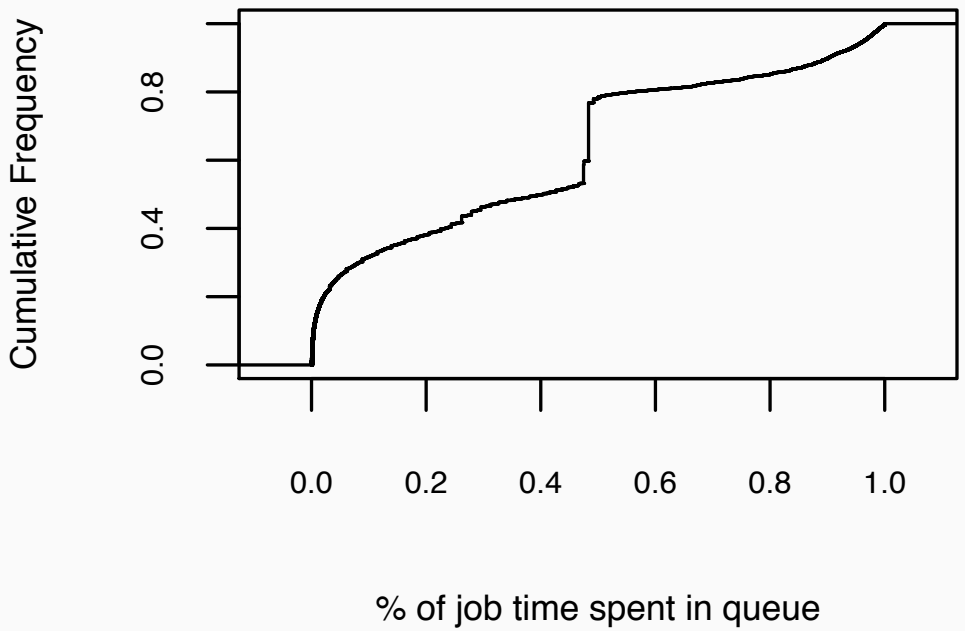


Figure 7: OLCF memory bandwidth by thread count

Queue Considerations

- HPC systems typically rely on queuing mechanisms for compute jobs
- For some systems, these times can prohibit on-demand analytics



Cost Analysis

- In order to gain performance, either through increased memory throughput at the node level, or by scaling out resources to decrease queue times at the system level, there is an associated cost
- For example, in the Rhea (GPU) nodes, memory attributes over 40% of total fixed cost.
- Queue policies in HPC facilities introduce significant resource idling by reserving cores for large and short lived jobs

What is a data science workload?

Let us start by considering a ”unit” of analytics as a collection of analytics jobs and define this unit of analytics as:

- 1,000 matrix multiplications on $20,000 \times 20,000$ matrices
- 1,000 singular value decompositions on $30,000 \times 30,000$ matrices
- 1,000 linear regressions on 50 million observations
- 1,000 k-means clustering on 50 million observations with 2 features
- Using 2 nodes

Scaling Performance and Cost

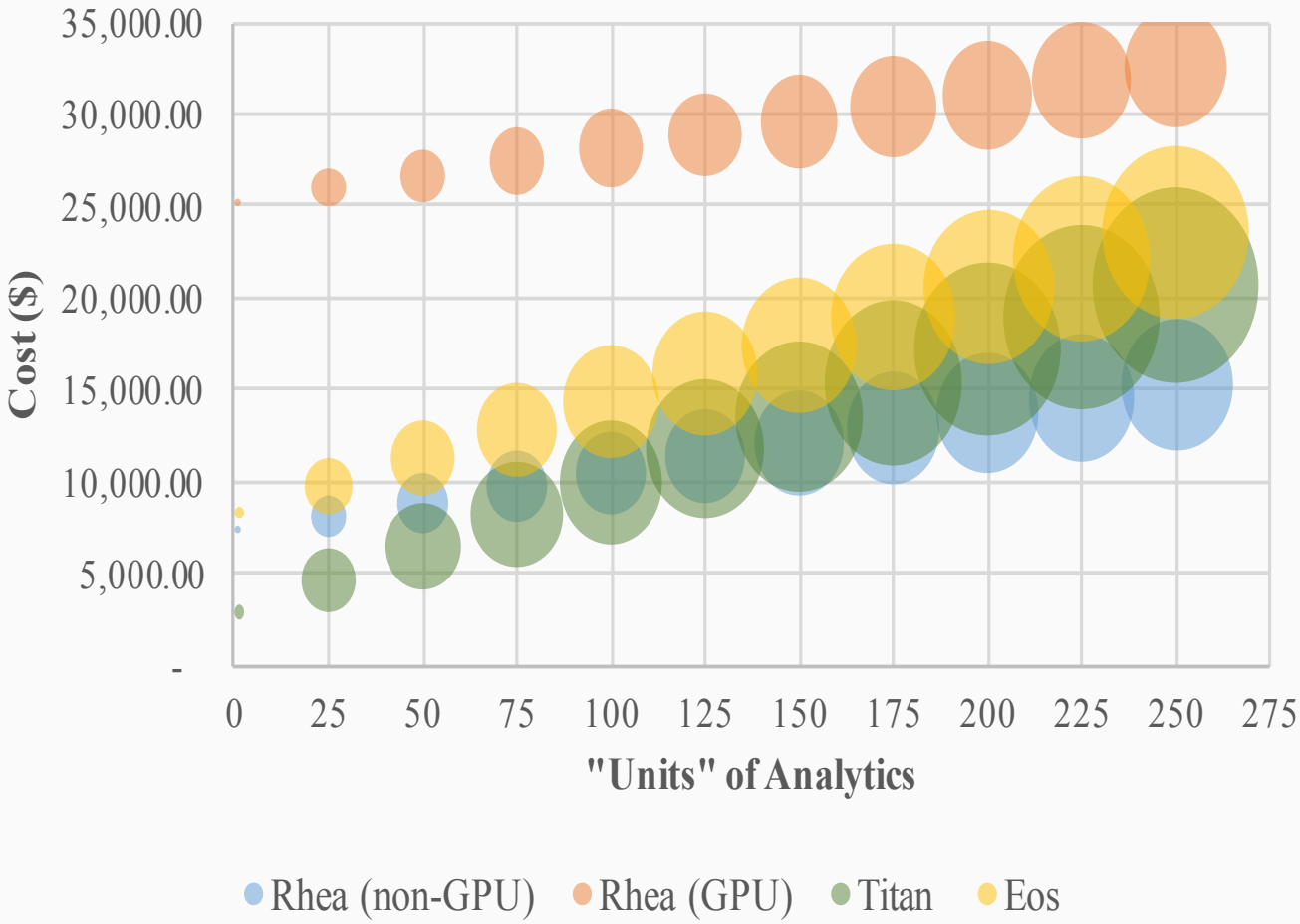


Figure 9: Relationship between quantity of analytics, cost and performance

Data Volume and Cost

- The volume of data needs are an important cost consideration
- More data requires more nodes, which drives up fixed and variable cost
- This is where we see that fixed costs help reduce costs in the long run
- For example, the Rhea fat-node architecture is cheapest to process 100TB of data, and AWS is the most expensive

Limitations

- High variation in variable cost among architectures across organizations
- Electrical costs especially varies across locales
- No benchmarking or consideration of GPU or interconnect

Conclusion

Conclusion

- ”Fat” node structures, with large amounts of memory and high memory bandwidth, are better suited for big data analytics, delivering up to **3x** speedup.
- Due to their flexibility and availability, cloud computing infrastructures are best suited to small or experimental jobs, but cost at scale for data analytics favors HPC.
- Furthermore, this cost structure favors fat nodes due to their ability to fit more data onto one node, which reduces both fixed and variable cost.
- Further research is needed at a more comprehensive cost-performance model, as well as quantifying the role of other hardware in data science, such as a GPU or interconnect.

Questions?