BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

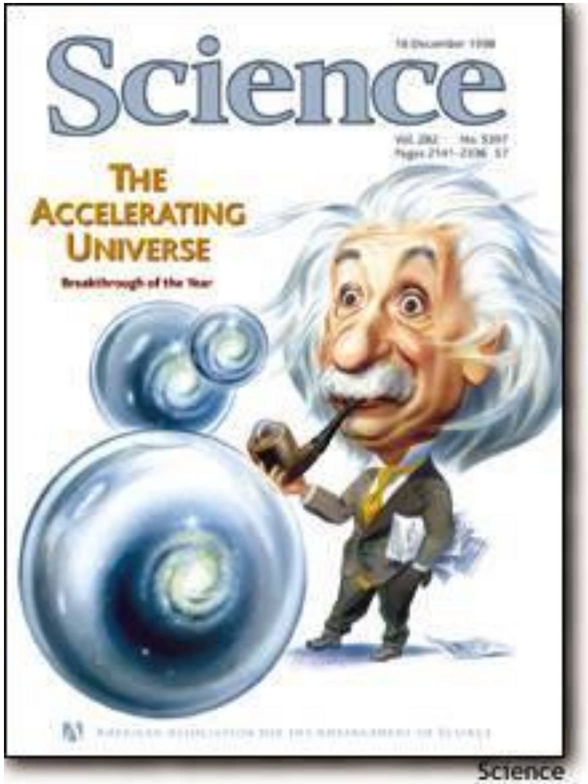
Seeking Supernovae in the Clouds: A Performance Study

Keith R. Jackson, Lavanya Ramakrishnan, Karl J. Runge, Rollin C. Thomas

Lawrence Berkeley National Laboratory

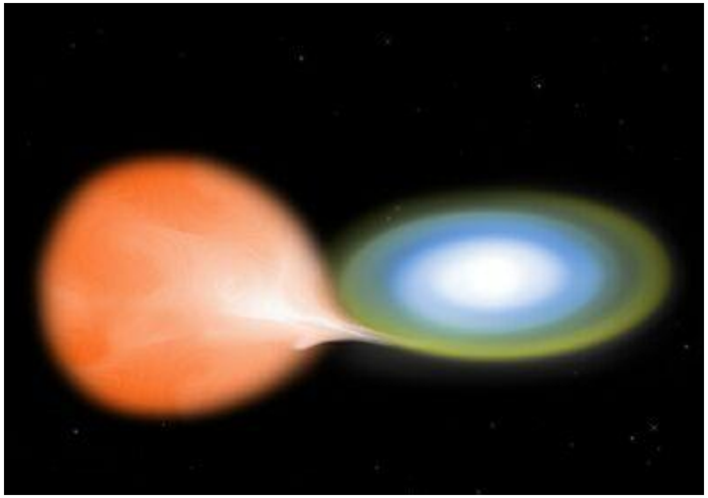
Why Do I Care About Supernovae?

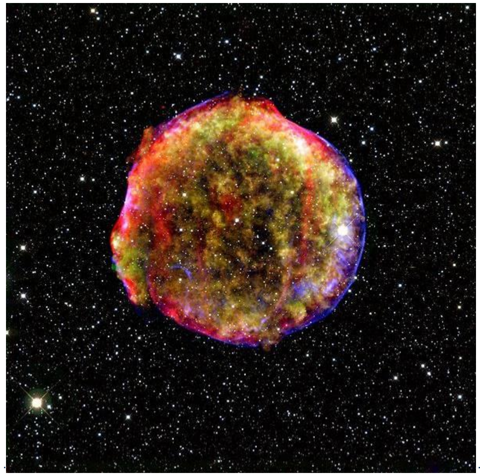
- The rate of expansion of the universe is accelerating
  - Propelled by mysterious new physics dubbed “Dark Energy”
  - This discovery was made by studying the brightness of Type 1a supernovae at varying distances



Type 1a Supernovae

- White dwarf stars that accrete gas from a companion star
- Explode at a critical mass equal to 1.4x the mass of the sun
- Standard amount of fuel creates a “Standard Candle” that can be used to measure their distance





Nearby Supernova Factory (Snfactory)

- Experiment to develop Type Ia supernovae as tools to measure the expansion history of the Universe and explore the nature of Dark Energy
  - Largest data volume supernova search from 2004-2008
  - Over 600 spectroscopically-confirmed supernovae
  - Observed both optically and with a custom designed spectrograph
- Collaboration between:
  - LBNL, Yale in the US
  - LPNHE, IPNL and CRAL IN2P3/CNRS labs in France

Data Pipeline

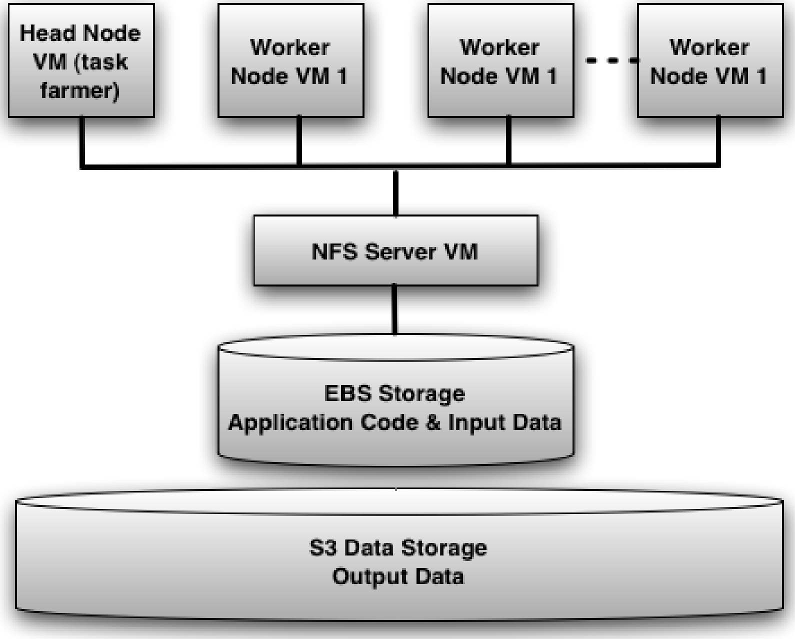
- Custom data analysis codes
  - Run on a standard Linux cluster
    - Jobs submitted to a standard batch queue
  - Controlled by a series of Python scripts
    - Used for coordination
- Complex set of algorithms
  - digital filtering, Fourier transforms, full matrix inversions, and nonlinear function optimization
- Heavily dependent on external packages
  - CFITSIO , the GNU Scientific Library (GSL), scipy, numpy, BLAS, LAPACK
  - Process level parallel
    - Large numbers of serial codes with different inputs

Why Interested in the Cloud?

- Long lived scientific project with most software development upfront
- Using shared clusters at NERSC and CCIN2P3
  - Changes to the clusters necessitate drafting scientists into debugging and rewriting codes
    - Change from 32 to 64 bit OS
- Amazon AWS Cloud provides:
  - Control over OS versions
  - Ability to install packages
  - Root access and ability to use shared “group” account
  - Immunity to externally enforced OS or architecture changes

Amazon AWS Setup

- SNfactory assumes a traditional HPC cluster environment
  - Not an option to re-architect for the Cloud
  - Emulate the cluster environment in the Cloud
- Used c1.medium instance types
  - 2 virtual cores, 2.5 EC2 Compute Units each
  - Most cost effective for Snfactory codes



Data Placement Options

- Elastic Block Store (EBS)
  - Provides a block level storage device that can be attached to an EC2 instance
    - Can be shared amongst instances via NFS
- Simple Storage Service (S3)
  - Highly scalable object store
  - Get/Put key values
  - Simple REST interface

amazon webservices™

Amazon S3

Simple Storage Service

Experimental Setup

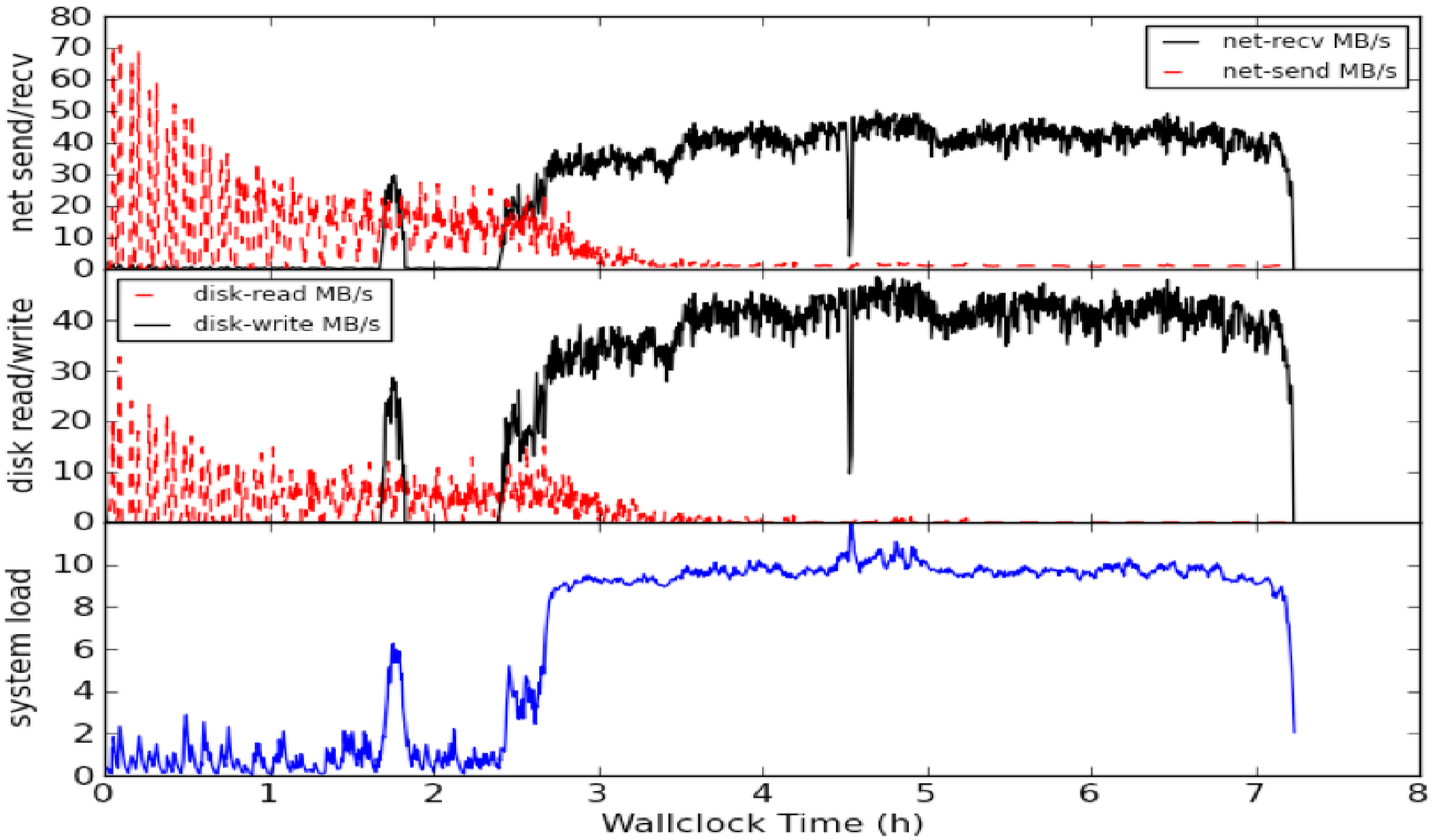
- Examine the data storage options to maximize performance
- Use the linux sar command to collect performance data
- One nights handled by a worker VM
  - ~370 files
  - ~2.7 GB of input data
  - ~9 GB of output data
- 80 core virtual cluster

Input Data	Output Data
EBS via NFS	Local storage to EBS via NFS
Staged to local storage from EBS	Local storage to EBS via NFS
EBS via NFS	EBS via NFS
Staged to local storage from EBS	EBS via NFS
EBS via NFS	Local storage to S3
Staged to local storage from S3	Local storage to S3

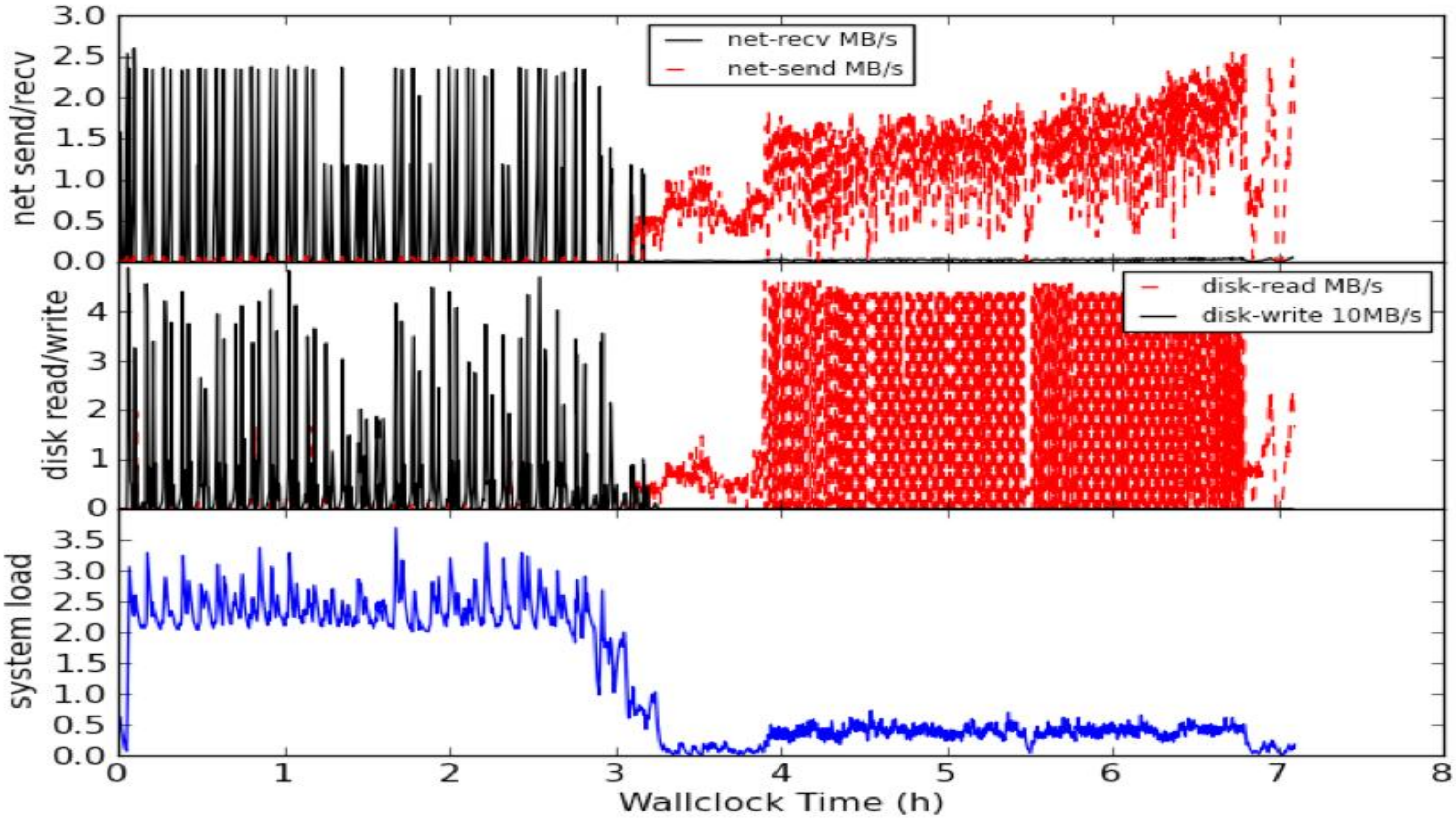
Experiment 1

- Input from EBS via NFS
- Output to local storage and then staged to EBS via NFS after the processing is complete

NFS Server


net send/recv
net-recv MB/s
net-send MB/s
disk read/write
disk-read MB/s
disk-write MB/s
system load
Wallclock Time (h)
80
70
60
50
40
30
20
10
0
40
30
20
10
0
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8

Worker Node

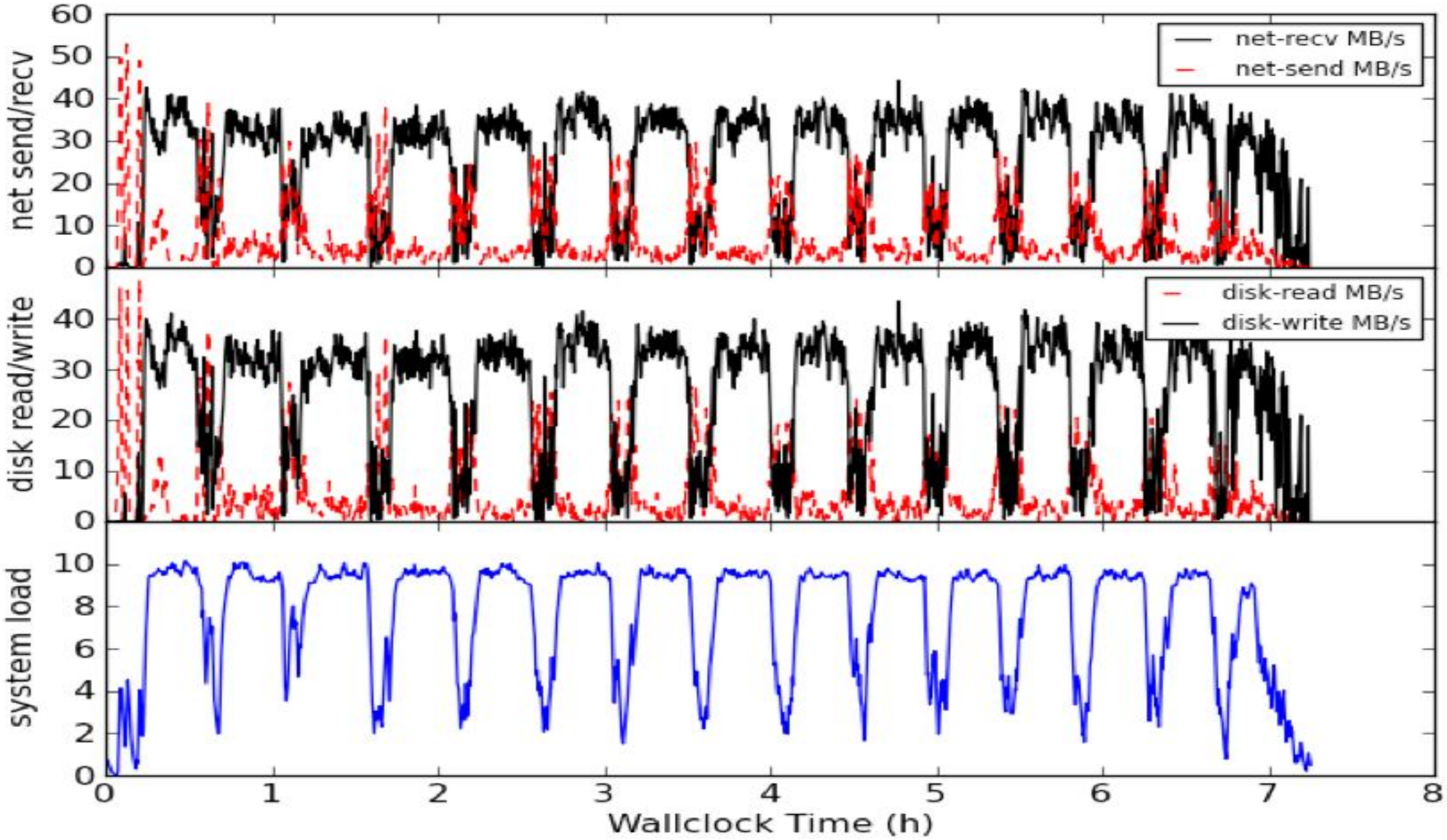


net-recv MB/s
net-send MB/s
net send/recv
3.0
2.5
2.0
1.5
1.0
0.5
0.0
disk-read MB/s
disk-write 10MB/s
disk read/write
4
3
2
1
0
system load
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
8
Wallclock Time (h)

Experiment 2

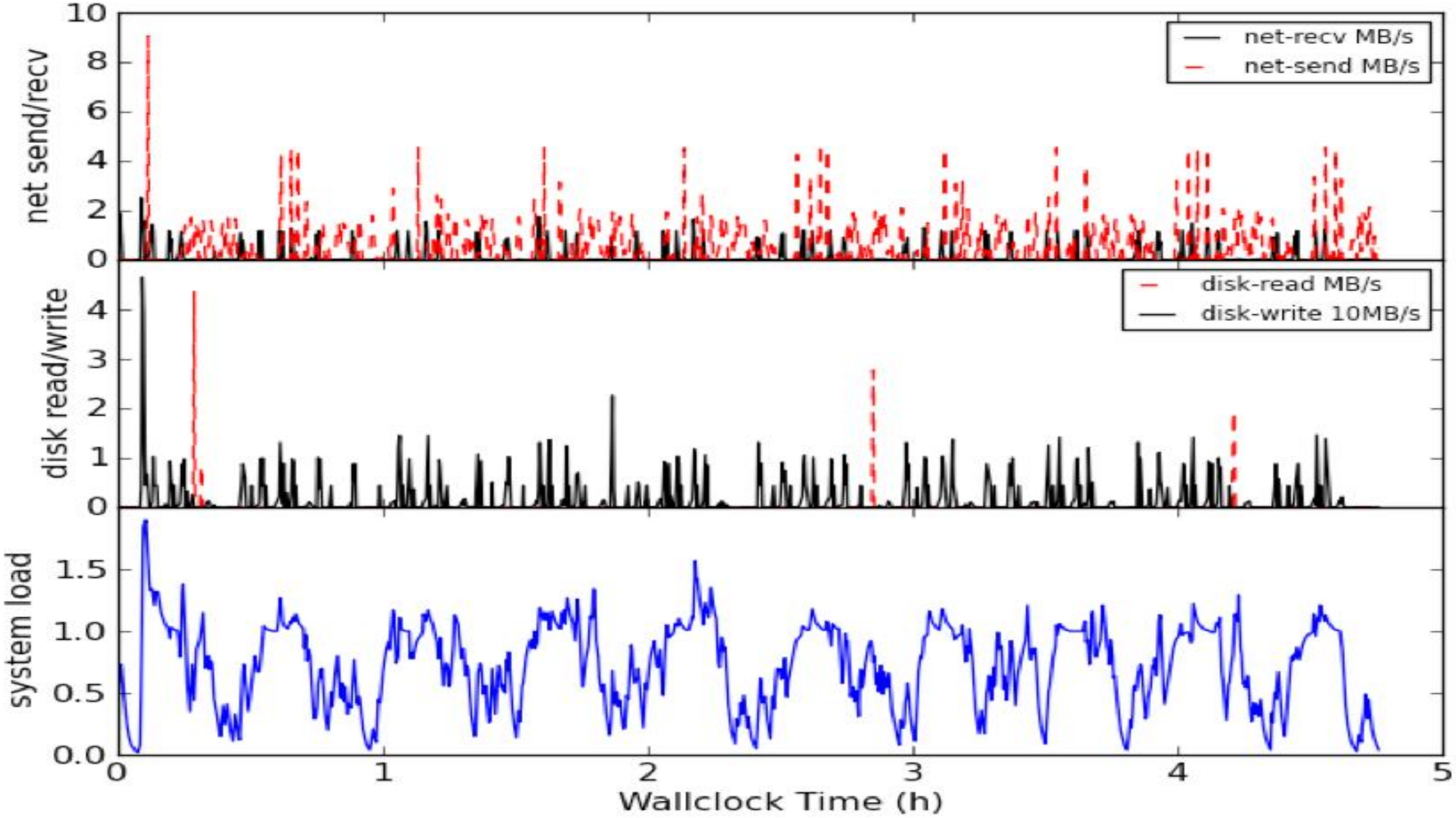
- Input data read from EBS via NFS
- Output data written directly to EBS via NFS
  - Interleave the I/O with data processing

NFS Server



net send/recv
disk read/write
system load
net-recv MB/s
net-send MB/s
disk-read MB/s
disk-write MB/s
Wallclock Time (h)
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8

Worker Node

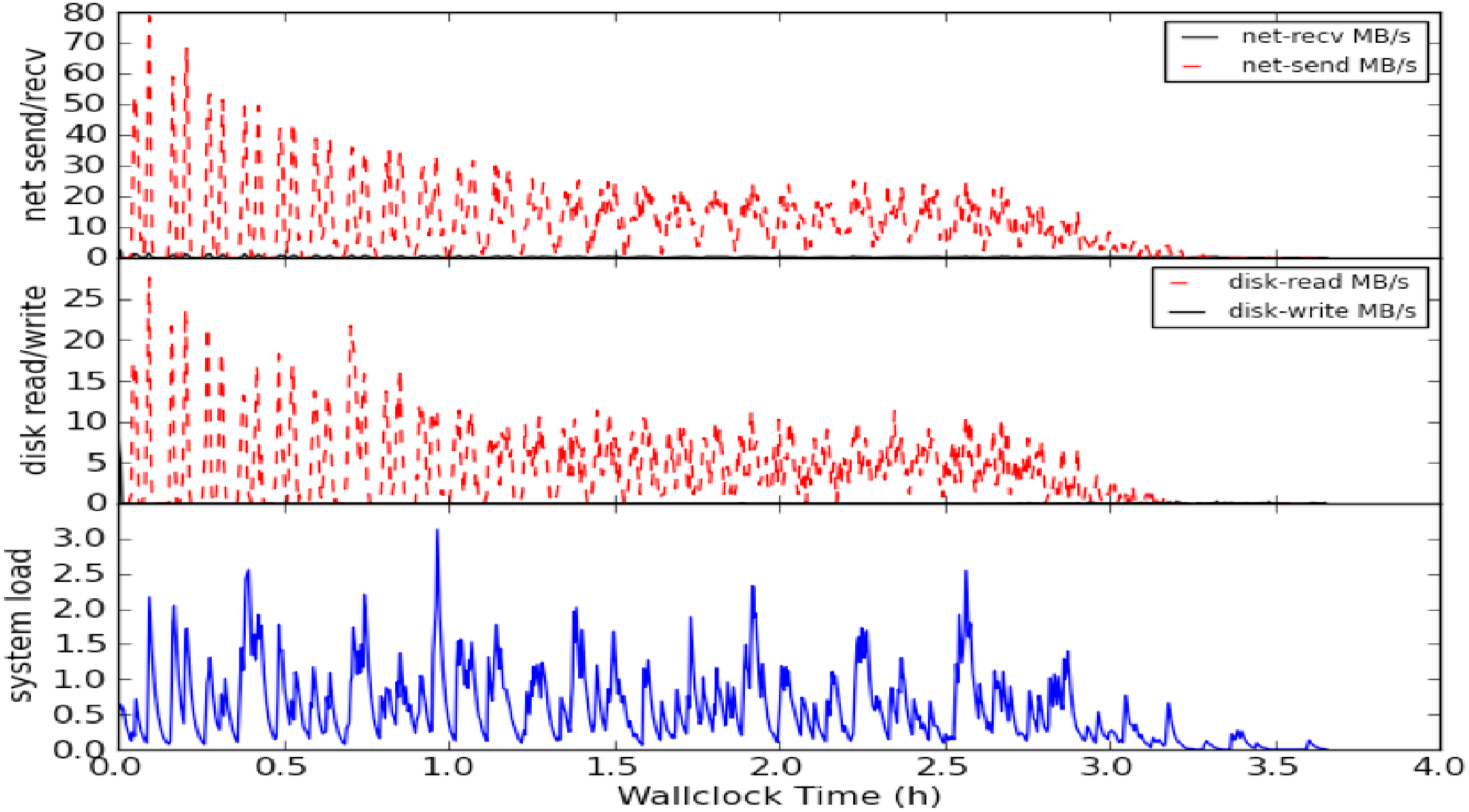


net send/recv
net-recv MB/s
net-send MB/s
disk read/write
disk-read MB/s
disk-write 10MB/s
system load
Wallclock Time (h)

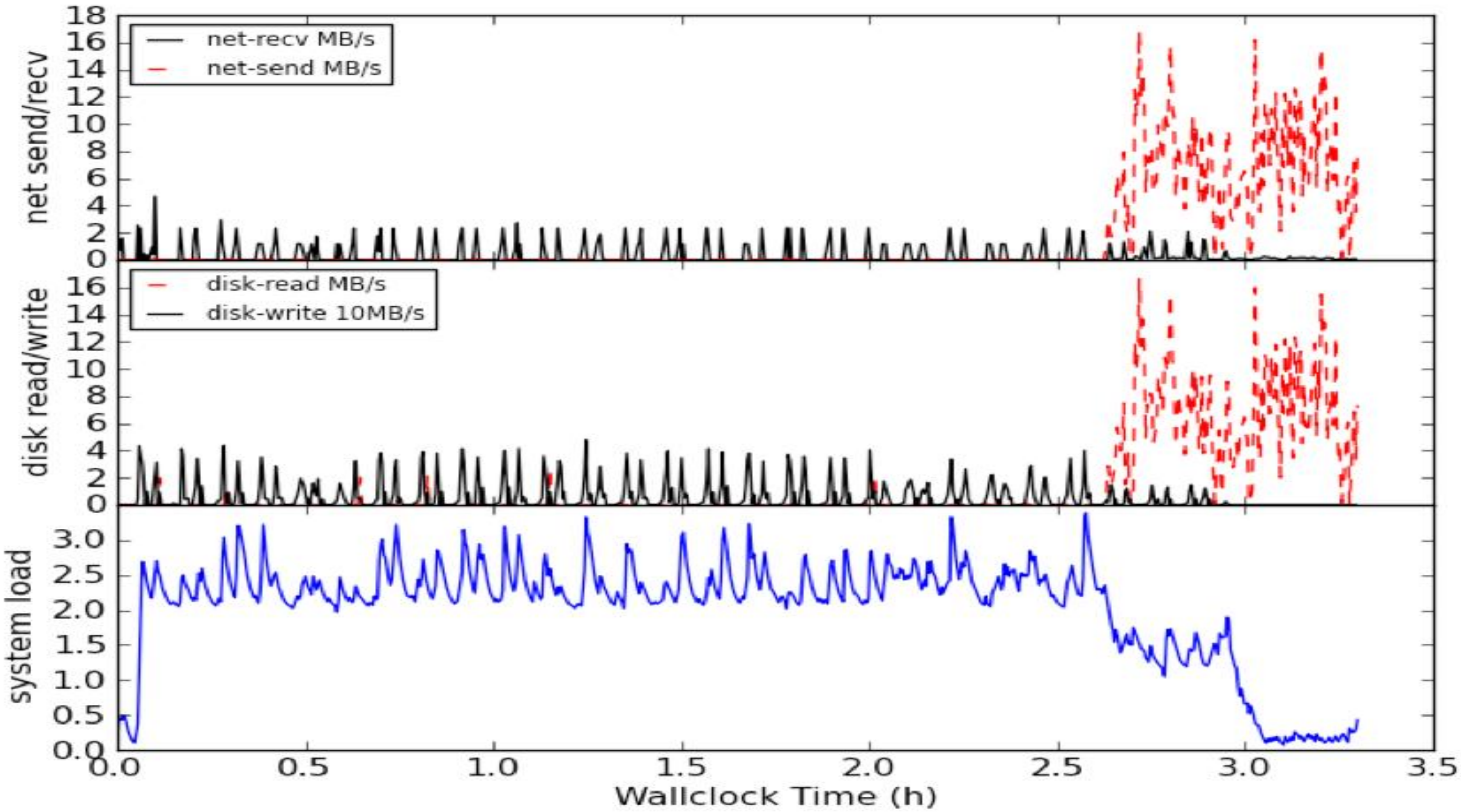
Experiment 4

- Input data read from EBS via NFS
- Output written to local storage and then to S3

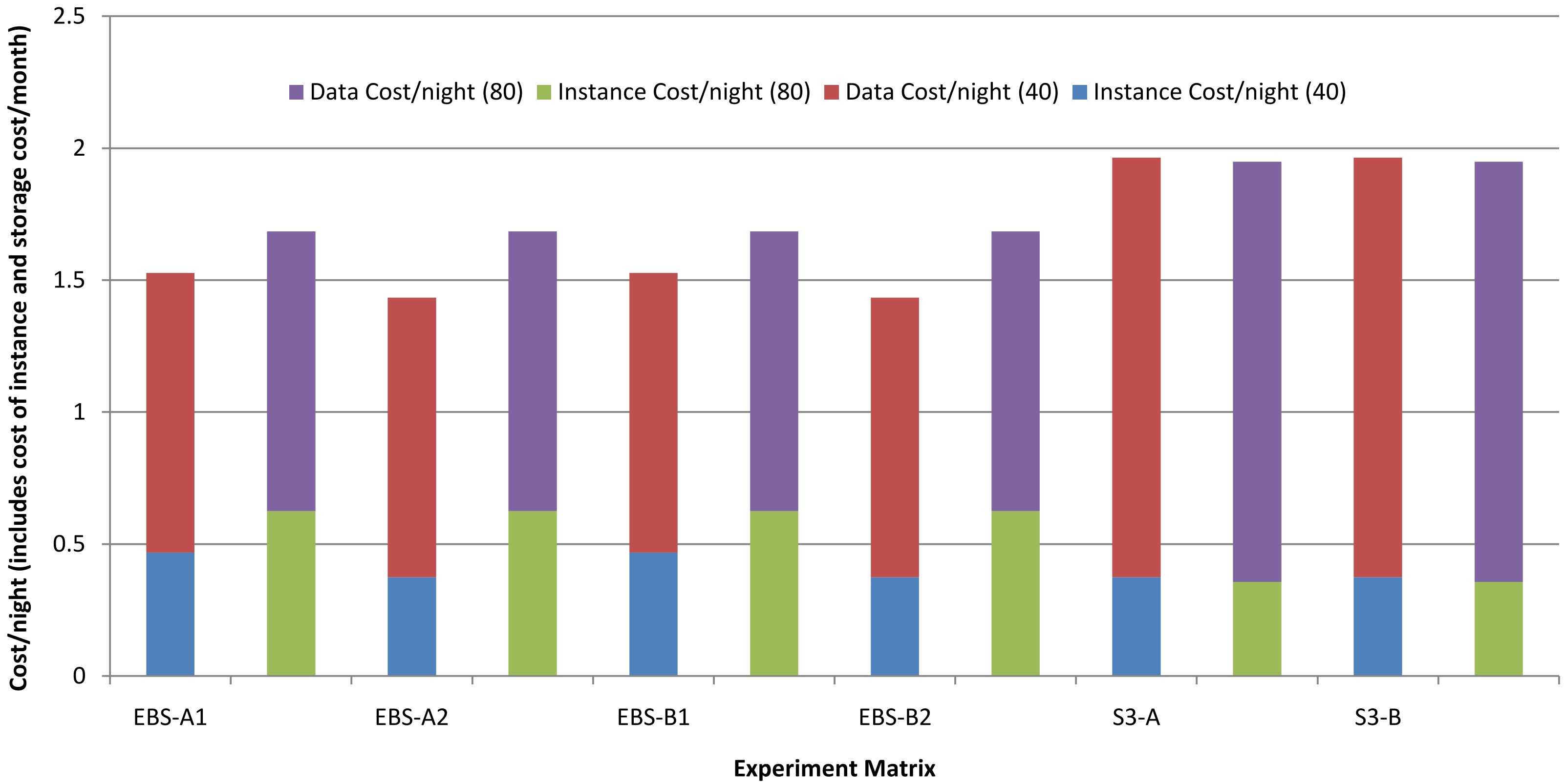
NFS Server


net send/recv
80
70
60
50
40
30
20
10
0
net-recv MB/s
net-send MB/s
disk read/write
25
20
15
10
5
0
disk-read MB/s
disk-write MB/s
system load
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Wallclock Time (h)

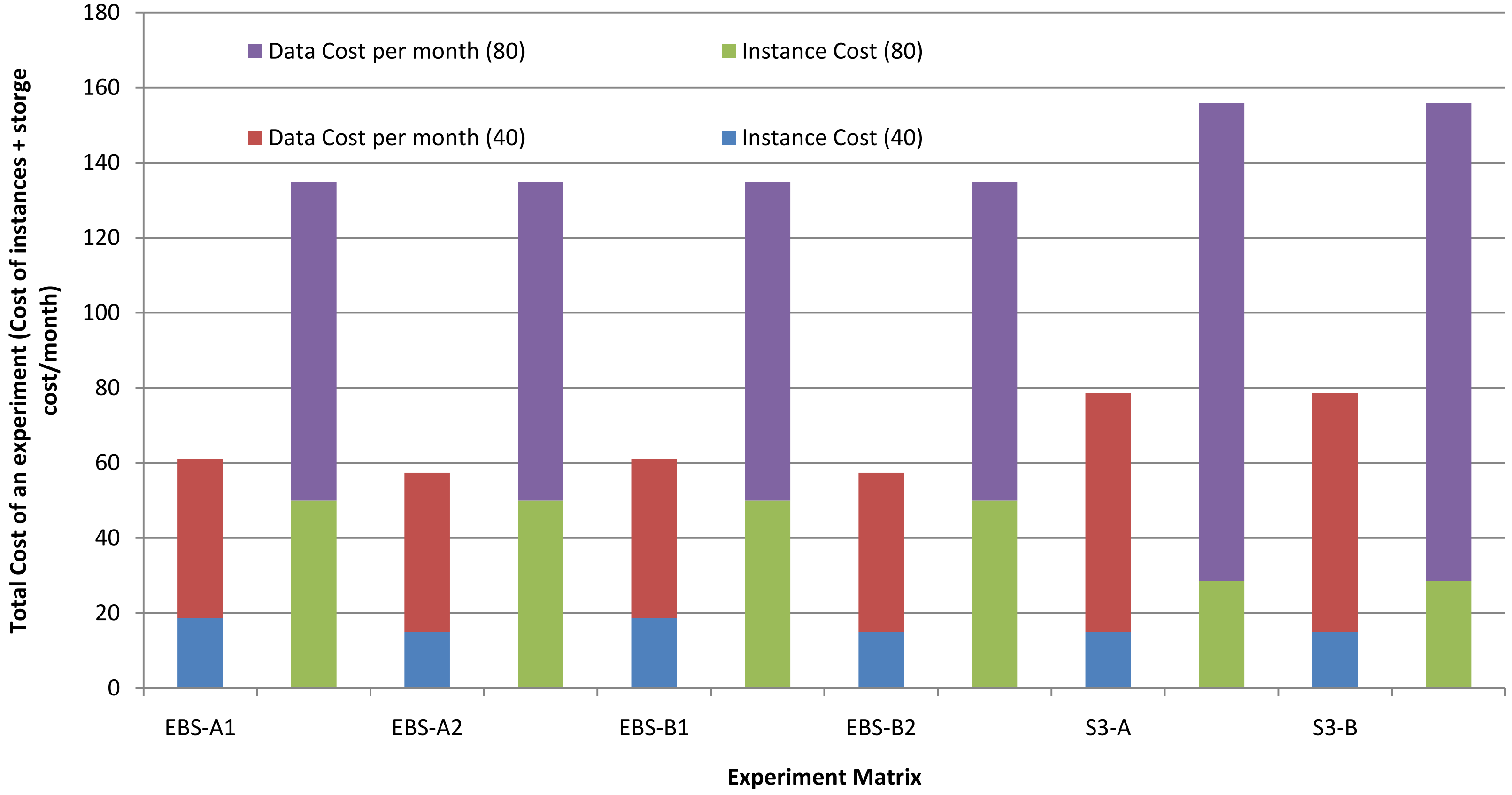
Worker Node


net-recv MB/s
net-send MB/s
disk-read MB/s
disk-write 10MB/s
net send/recv
disk read/write
system load
Wallclock Time (h)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

Cost of Analyzing One Night of Data



Costs of Full Experiment and 1 Month Data Storage



Failures

- Saw a significant aggregate rate of failure during our testing
  - Unable to allocate all 80 cores was the most common
    - Need to architect your system to adapt to the number of actual resources acquired
  - Unable to access the “user-data” passed in at instantiation time for customization
  - Failure to boot properly
  - Failure to configure the network
  - Poor performing nodes

Conclusions

- Porting your scientific application to the Amazon AWS Cloud requires significant work
- Must architect your application to handle failure
- Benchmarking your application is essential
  - Establish what instance type is most cost effective for your application
    - Cheaper per/node cost does not always translate to the cheapest overall cost
  - Understand the cost/performance tradeoffs for the various storage options
- Directly porting existing scientific applications does not best utilize the features available in the AWS Cloud

Acknowledgements

- This work was funded in part by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH112
- Amazon for providing AWS access
- The Magellan team at NERSC

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

InfoWorld 2009 TECHNOLOGY OF THE YEAR AWARDS
Best Cloud Platform
amazon web services™

Office of Science
U.S. Department of Energy