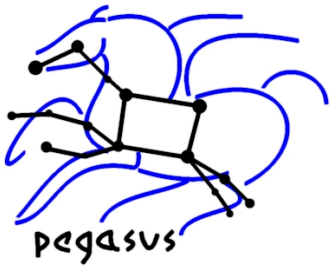
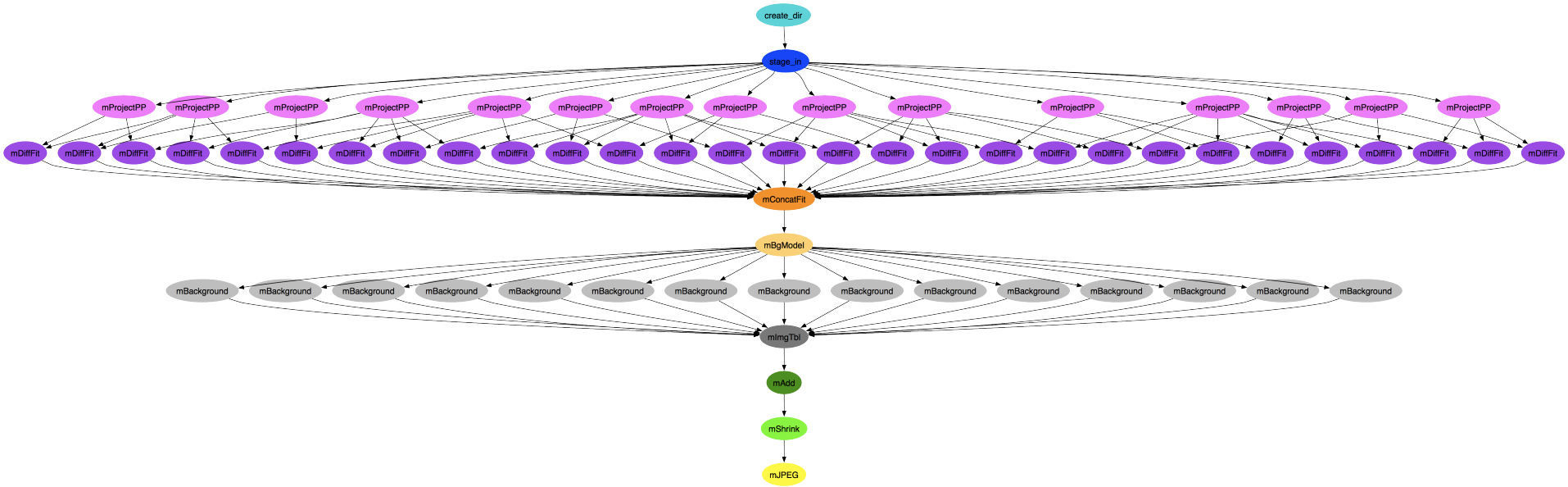
Scientific Workflows and Cloud Computing

Gideon Juve
USC Information Sciences Institute
gideon@isi.edu



Scientific Workflows

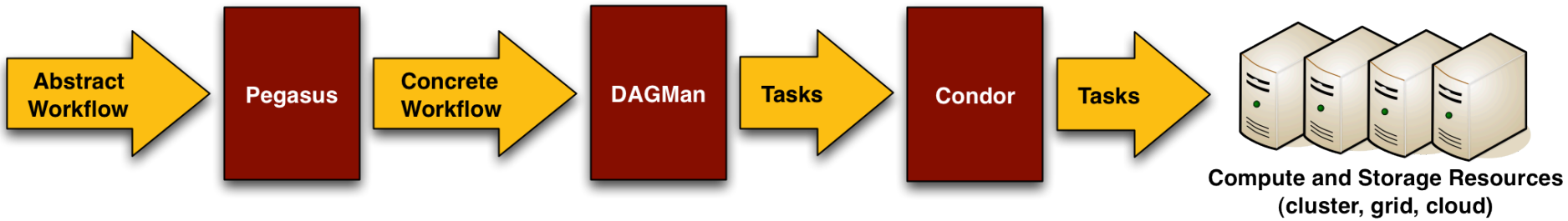
- Loosely-coupled parallel applications
- Expressed as directed acyclic graphs (DAGs)
  - Nodes = Tasks, Edges = Dependencies
- Data is communicated via files



Small Montage Workflow

Workflow Management System

- Pegasus – workflow planner
  - Efficiently maps tasks and data to resources
- DAGMan – workflow engine
  - Tracks dependencies, releases tasks, retries tasks
- Condor – task manager
  - Dispatches tasks (and data) to resources



Pegasus WMS

Amazon Web Services (AWS)

- IaaS Cloud
- Services
  - Elastic Compute Cloud (EC2)
    - Provision virtual machine instances
  - Simple Storage Service (S3)
    - Object-based storage system
    - Put/Get files from a global repository
  - Elastic Block Store (EBS)
    - Block-based storage system
    - Unshared, SAN-like volumes
  - Others (queue, key-value, RDBMS, MapReduce, etc.)

Workflows and Clouds

- Benefits
  - User control over environment
  - On-demand provisioning / Elasticity
  - SLA, support, reliability, maintenance
- Drawbacks
  - Complexity (more control = more work)
  - Cost
  - Performance
  - Resource Availability
  - Vendor Lock-In

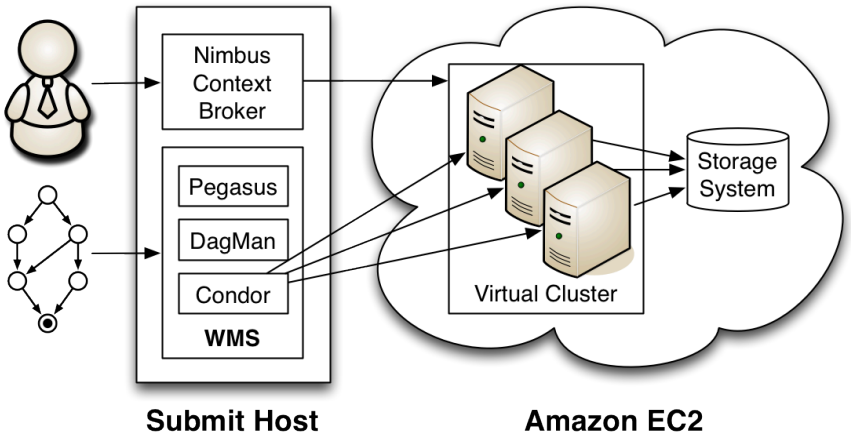
Questions About Clouds

- How can we deploy workflows in the cloud?
  - Install and configure software
  - Execute workflow tasks
  - Store workflow data
- How well do workflows perform in the cloud?
  - Compared to grids and clusters
  - Using various storage systems
- How much does it cost to run a workflow?
  - To provision resources
  - To store data
  - To transfer data

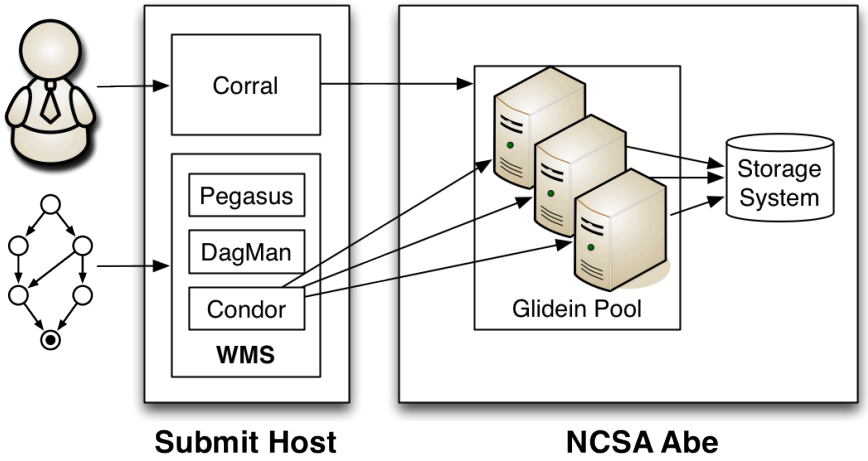
Deploying Workflows in the Cloud

- Virtual Machines/Virtual Machine Images
  - Clouds provide resources, but the software is up to the user
- Virtual Clusters
  - Collections of virtual machines used together
  - Configured to mimic traditional clusters
- Contextualization
  - Dynamically configuring virtual clusters is not trivial
  - Nimbus Context Broker – automates provisioning and configuration of virtual clusters

Execution Environment



Cloud



Grid

Workflow Storage In the Cloud

- Executables
  - Transfer into cloud
  - Store in VM image
- Input Data
  - Transfer into cloud
  - Store in cloud
- Intermediate Data
  - Use local disk (single node only)
  - Use distributed storage system
- Output Data
  - Transfer out of cloud
  - Store in cloud

Resource Type Experiments

- Run workflows on single instances of different resource types (using local disk)
- Goals:
  - Compare performance/cost of cloud resources
  - Compare performance of grid and cloud
  - Characterize virtualization overhead
  - Quantify performance benefit of network/file system

Type	Arch.	CPU	Cores	Memory	Network	Storage	Price
m1.small	32-bit	2.0-2.6 GHz Opteron	1/2	1.7 GB	1-Gbps Ethernet	Local disk	$0.085/hr
m1.large	64-bit	2.0-2.6 GHz Opteron	2	7.5 GB	1-Gbps Ethernet	Local disk	$0.12/hr
m1.xlarge	64-bit	2.0-2.6 GHz Opteron	4	15 GB	1-Gbps Ethernet	Local disk	$0.68/hr
c1.medium	32-bit	2.33-2.66 GHz Xeon	2	1.7 GB	1-Gbps Ethernet	Local disk	$0.17/hr
c1.xlarge	64-bit	2.33-2.66 GHz Xeon	8	7.5 GB	1-Gbps Ethernet	Local disk	$0.68/hr
abe.local	64-bit	2.33 GHz Xeon	8	8 GB	10-Gbps InfiniBand	Local disk	N/A
abe.lustre	64-bit	2.33 GHz Xeon	8	8 GB	10-Gbps InfiniBand	Lustre	N/A

Resource Types Used

Storage System Experiments

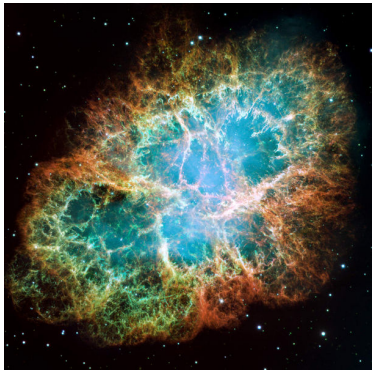
- Investigate different options for storing intermediate data in a virtual cluster
- Goals
  - Determine how to deploy storage systems
  - Compare performance/cost of storage systems
  - Determine which storage system
- Amazon Issues
  - EC2 does not allow kernel patches (no Lustre, Ceph)
  - EBS volumes cannot be shared between nodes
- Use c1.xlarge resources

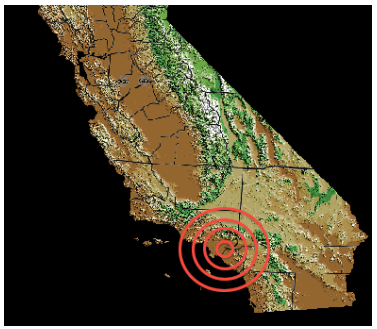
Storage Systems

- Local Disk
  - RAID0 across available partitions with XFS
- NFS: Network file system
  - 1 dedicated node (m1.xlarge)
- PVFS: Parallel, striped cluster file system
  - Workers host PVFS and run tasks
- GlusterFS: Distributed file system
  - Workers host GlusterFS and run tasks
  - NUFA, and Distribute modes
- Amazon S3: Object-based storage system
  - Non-POSIX interface required changes to Pegasus
  - Data is cached on workers

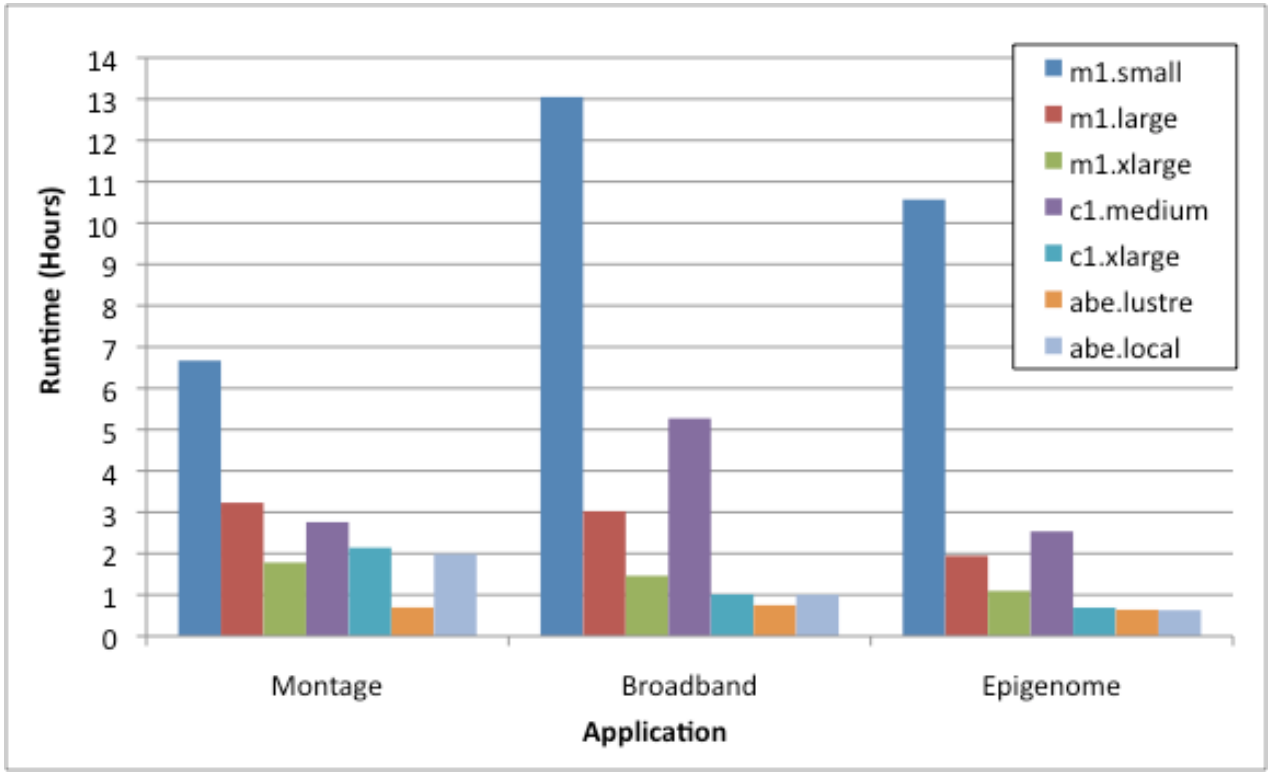
Example Applications

- Montage (astronomy)
  - I/O: High
  - Memory: Low
  - CPU: Low
- Epigenome (bioinformatics)
  - I/O: Low
  - Memory: Medium
  - CPU: High
- Broadband (earthquake science)
  - I/O: Medium
  - Memory: High
  - CPU: Medium



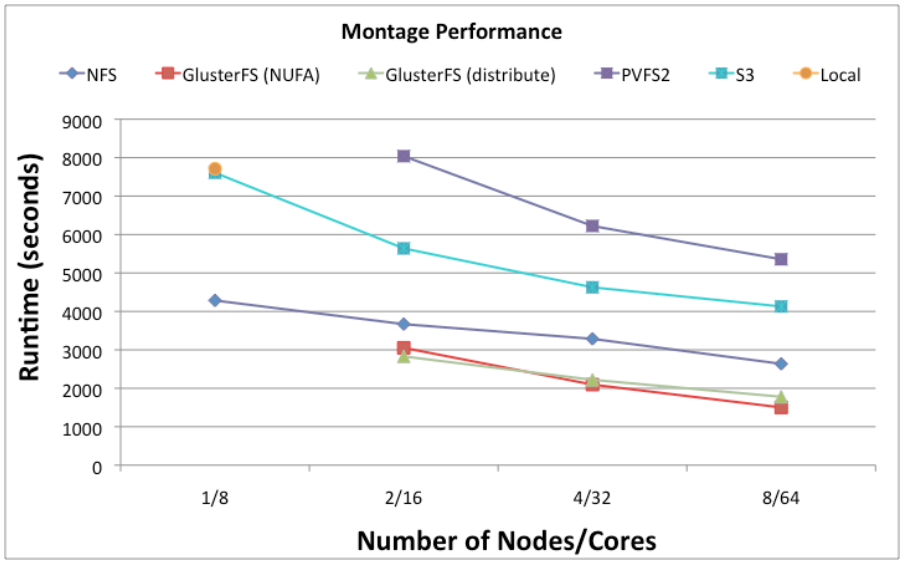


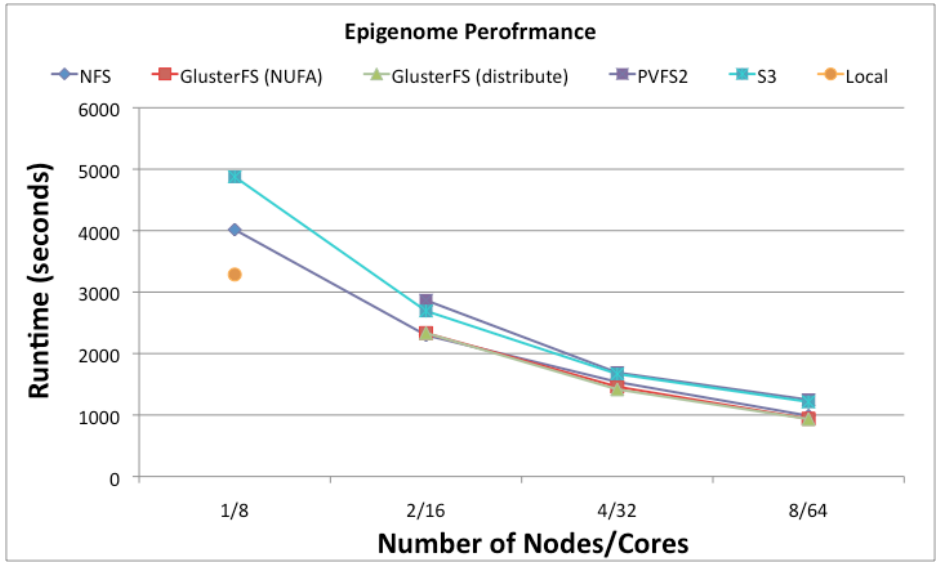
Resource Type Performance



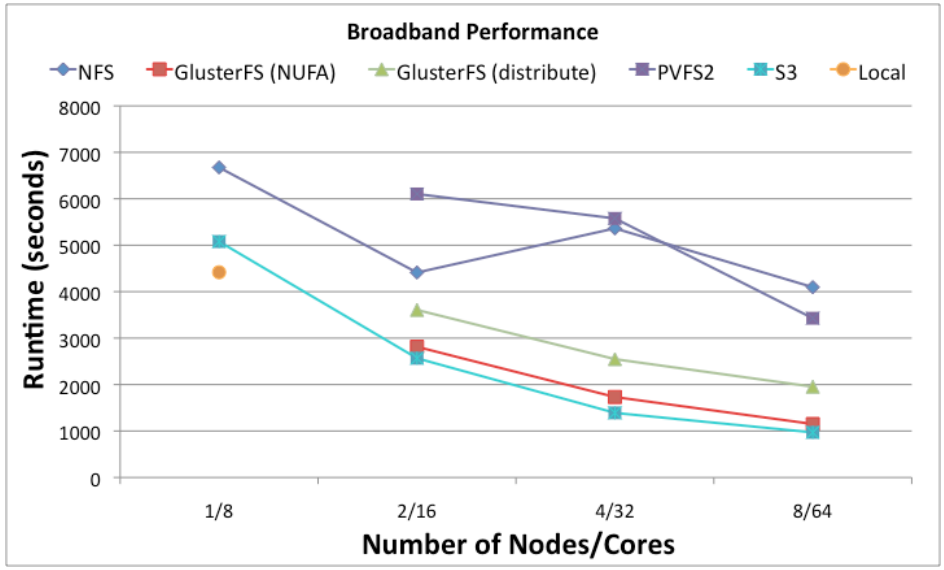
- Virtualization overhead is less than 10%
- Network/file system are biggest advantage for grid
- c1.xlarge is good, m1.small is bad
- Montage (high I/O) likes Lustre, Epigenome (high CPU) doesn't care

Storage System Performance





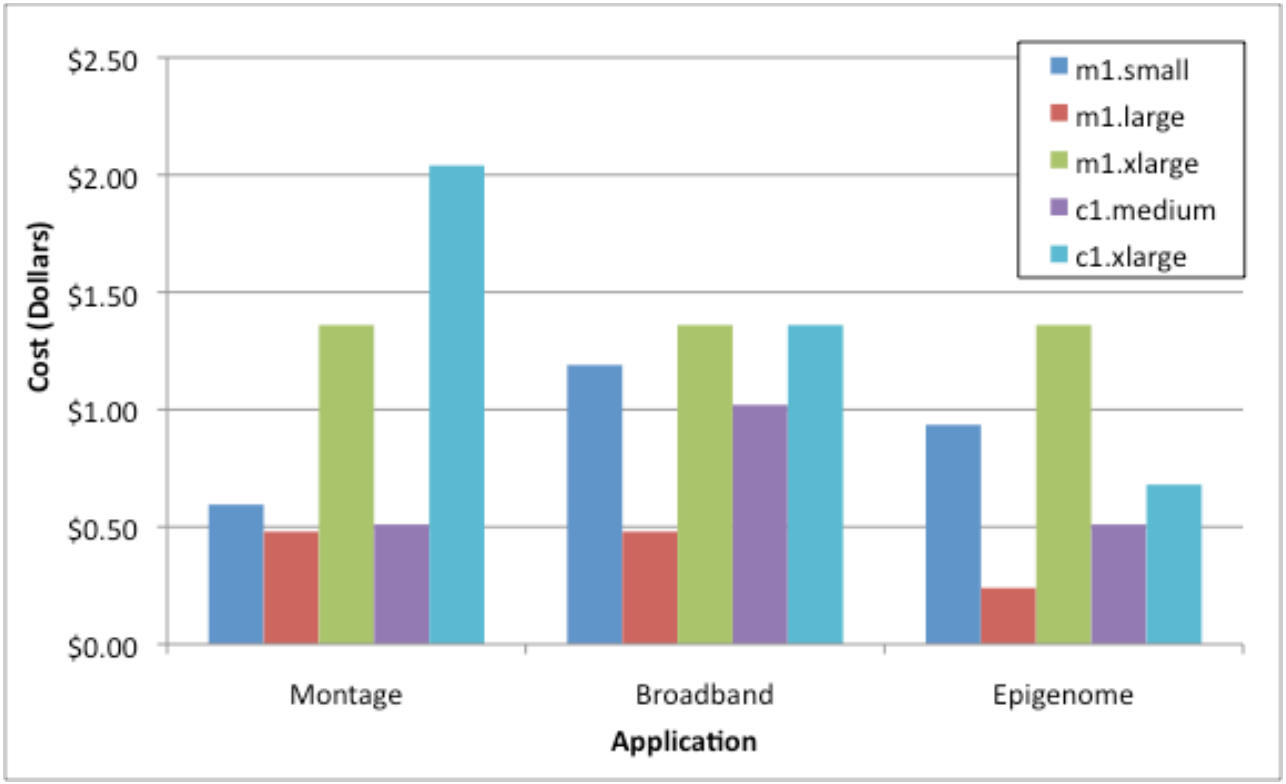
- GlusterFS (NUFA) is best overall
- Epigenome file system doesn't matter
- NFS, PVFS perform relatively poorly
- S3 performs poorly when reuse is low, and # files is large



Cost Components

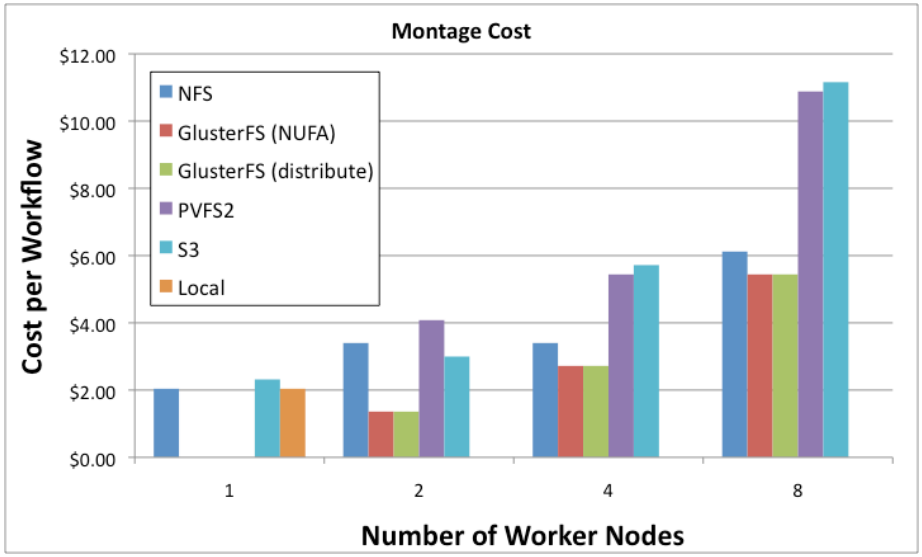
- Resource Cost
  - Cost for VM instances
  - Billed by the hour
- Transfer Cost
  - Cost to copy data to/from cloud over network
  - Billed by the GB
- Storage Cost
  - Cost to store VM images, application data
  - Billed by the GB-month, # of accesses

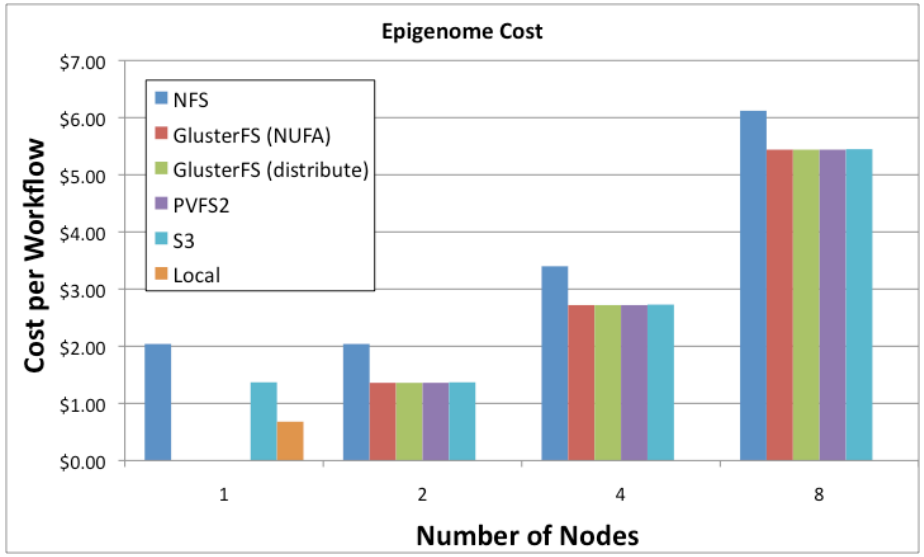
Resource Cost (by Resource Type)



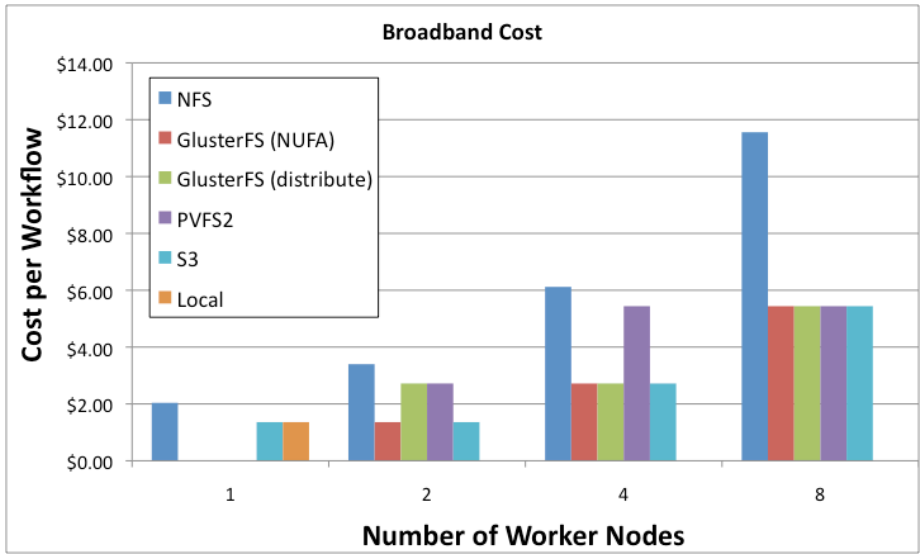
- The per-workflow cost is not bad
- m1.small is not the cheapest
- m1.large is most cost-effective
- Resources with best performance are not cheapest
- Per-hour billing affects price/ performance tradeoff

Resource Cost (by Storage System)





- Cost tracks performance
- Adding resources does not reduce cost (except in unusual cases)
- S3, NFS are at a disadvantage because of extra charges



Transfer Cost

Application	Input	Output	Logs
Montage	4291 MB	7970 MB	40 MB
Broadband	4109 MB	159 MB	5.5 MB
Epigenome	1843 MB	299 MB	3.3 MB

Transfer Sizes

Application	Input	Output	Logs	Total
Montage	$0.42	$1.32	< $0.01	$1.75
Broadband	$0.40	$0.03	< $0.01	$0.43
Epigenome	$0.18	$0.05	< $0.01	$0.23

Transfer Costs

- Cost of transferring data to/from cloud
  - Input: $0.10/GB (first 10 TB, free till June 30)
  - Output: $0.17/GB (first 10 TB, now $0.15)
- Transfer costs are a relatively large
  - For Montage, transferring data costs more than computing it
- Costs can be reduced by storing input data in the cloud and using it for multiple workflows

Storage Cost

- Storage Charge
  - Price for storing data
  - Per GB-month
- Access Charge
  - Price for accessing data
  - Per operation

- S3
  - Storage: $0.15 / GB-month
  - Access: PUT: $0.01 / 1,000
  - GET: $0.01 / 10,000
- EBS
  - Storage: $0.10 / GB-month
  - Access: $0.10 / million IOs

Application	Volume Size	Monthly Cost
Montage	5GB	$0.66
Broadband	5GB	$0.60
Epigenome	2GB	$0.26

Storage of Inputs in EBS

Image	Size	Monthly Cost
32-bit	773 MB	$0.11
64-bit	729 MB	$0.11

Storage of VM images in S3

Conclusions

- Deployment and Usability
  - Easy to start using, but some work is required to generate images and automate configuration
  - Tools like Nimbus Context Broker can help
  - Little maintenance, good reliability
- Performance
  - Not bad given resources, but not as good as dedicated clusters & grids
  - VM overhead is less than 10% for apps tested
  - c1.xlarge has best performance overall
  - Avoid using m1.small

Conclusions

- Cost
  - m1.small is not always the cheapest resource
  - Transferring data is relatively expensive
  - Store inputs long-term if possible
  - Using multiple nodes is not cost-effective

Web Resources

- Pegasus
  - http://pegasus.isi.edu
- Condor/DAGMan
  - http://cs.wisc.edu/condor
- Nimbus Context Broker
  - http://www.nimbusproject.org/
- Amazon Web Services
  - http://aws.amazon.com