



Infrastructure Outsourcing in Multi-Cloud Environment

Kate Keahey,

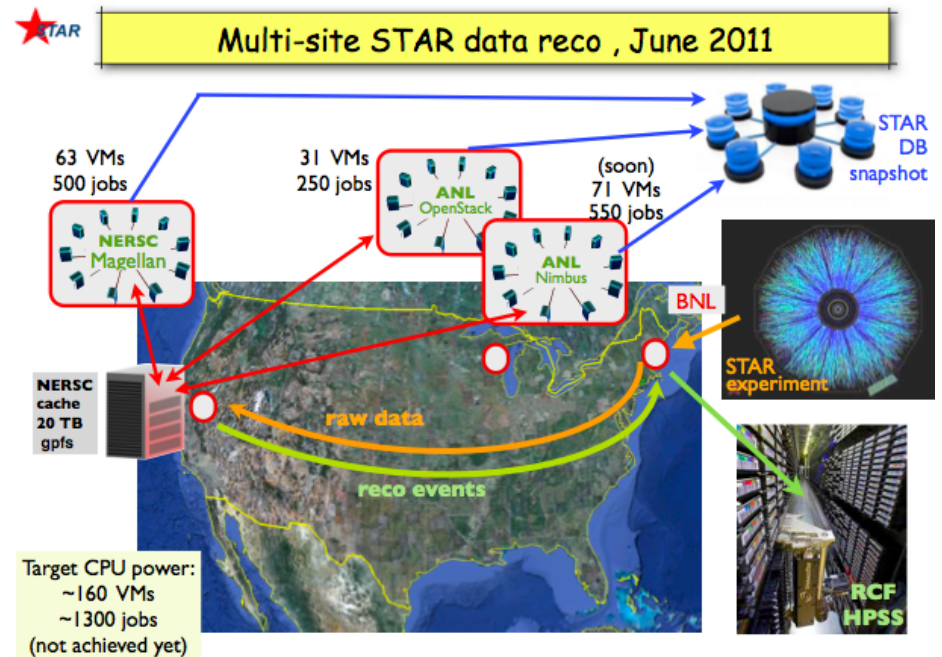
Patrick Armstrong, David LaBissoniere, Pierre Riteau

Argonne National Laboratory

Computation Institute, University of Chicago

The STAR Experiment

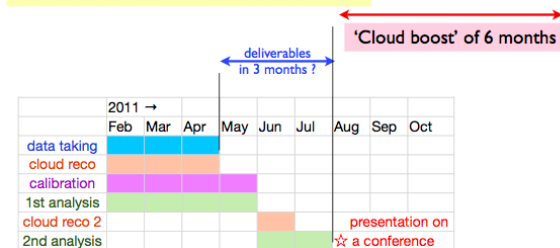
- Use case: cloud processing for W-boson reconstruction
- Benefits:
 - Near real-time processing
 - Reduce “time to science”
- Overall achievement
 - “10 months became 3 months”



Achieved timeline of W measurement in 2009



Intended timeline of W measurement in 2011



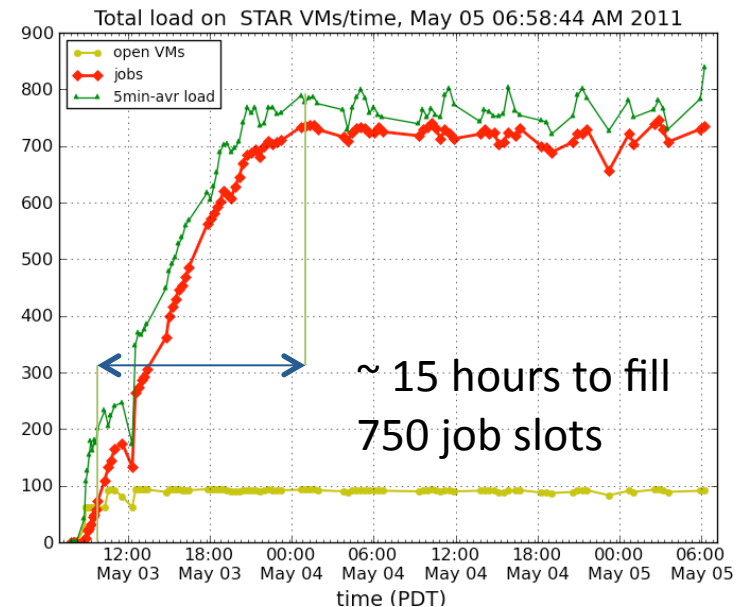
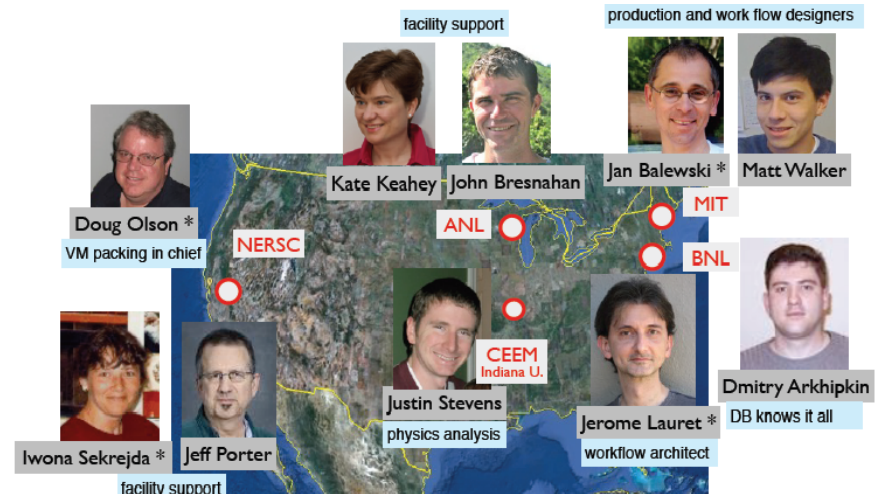
11/20/12



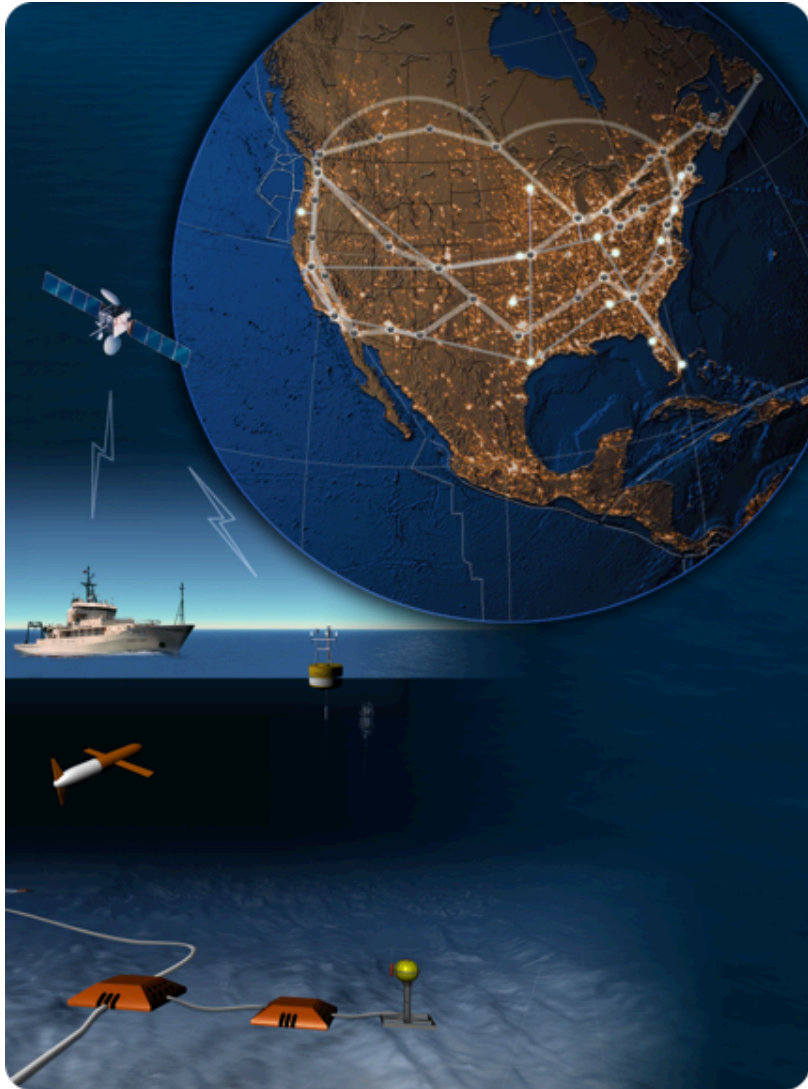
www.nimbusproject.org

... and Its Challenges

- Cloud-related issues:
 - Complexity: involving 10 people
 - Not very on-demand
 - Hard to coordinate several clouds – manual and ad hoc
 - Lack of consistency, scalability, bad reaction time
 - No autoscaling, streamlined fault management
 - No unified way to move/store data
 - Can't just define an appliance and have it work with several clouds
 - And many others...



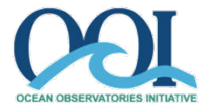
Ocean Observatory Initiative



- Towards Observatory Science
- Sensor-driven processing
 - An “always-on” service
 - Real-time event-based data stream processing capabilities
 - Highly volatile need for data distribution and processing
- Nimbus team building platform services for integrated, repeatable support for on-demand science
 - High-availability
 - Auto-scaling
- From regional Nimbus clouds to commercial clouds



www.nimbusproject.org

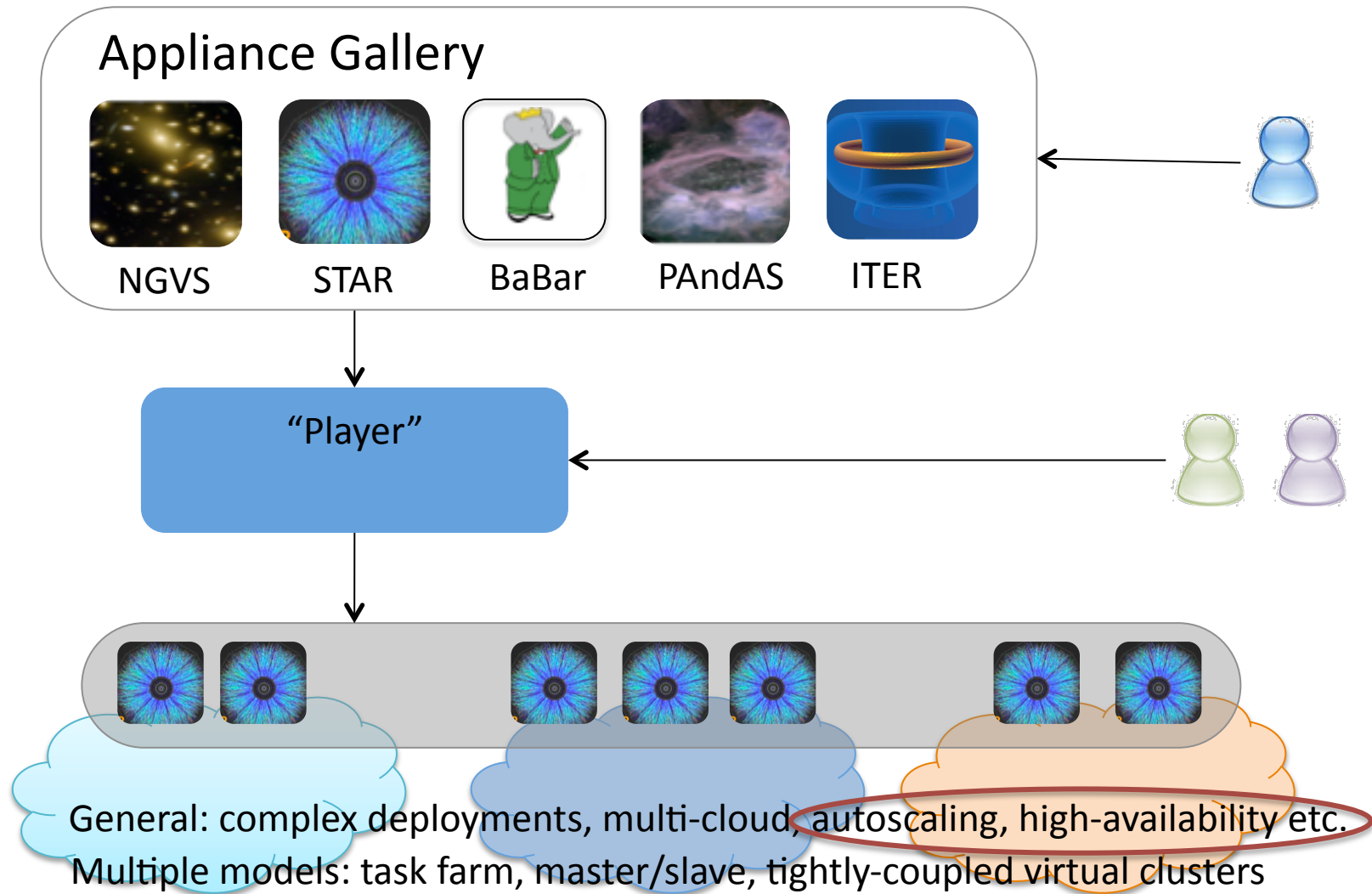


... and Its Challenges

- **Complexity Management:** complex deployment patterns, live upgrades and management, concurrent deployments, longevity, etc.
- **High Availability:** always-on service model, replenishable resources
- **Auto-scaling:** managing peaks and valleys, flexible resources
- **Multi-cloud:**
 - Escalation pattern: from private to commercial clouds
 - Risk mitigation: not enough cycles, failure, market factors
- **Configurability:**
 - Application and system sensors
 - Modifiable policies



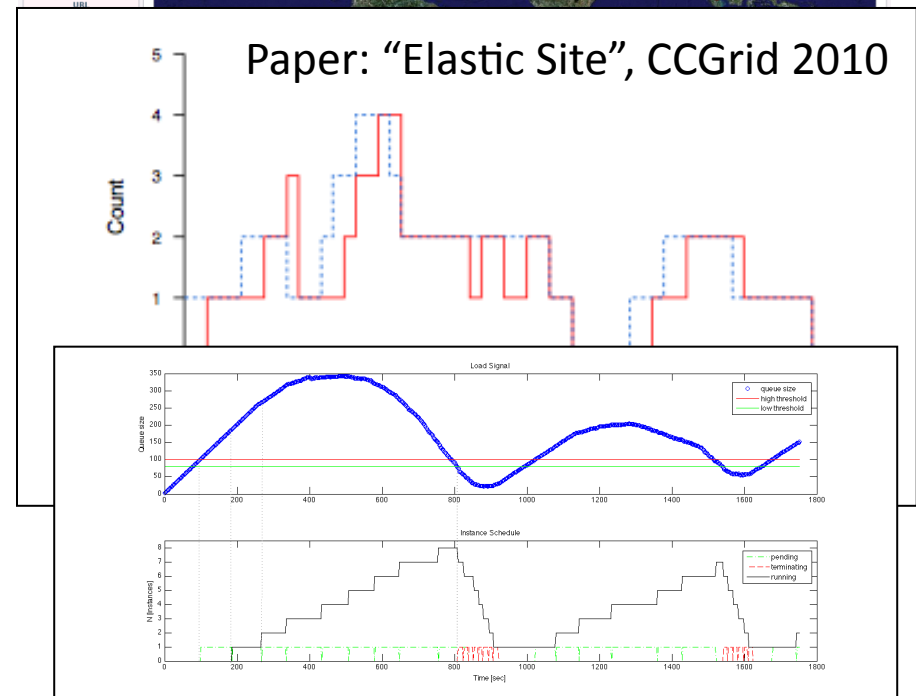
Infrastructure Outsourcing for Science



Autoscaling: Early Examples

- 2008: The ALICE proof-of-concept
- 2009: ElasticSite prototype
- 2009: OOI pilot

*Challenge:
a generic HA
elastic service model*



Starting Simple: the Task Farm Model



Domain Management

*Monitor and regulate domain properties
based on system-specific and application-specific metrics*



Domain
(configuration and security)

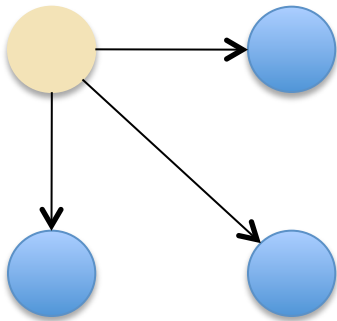


- **Assumptions**

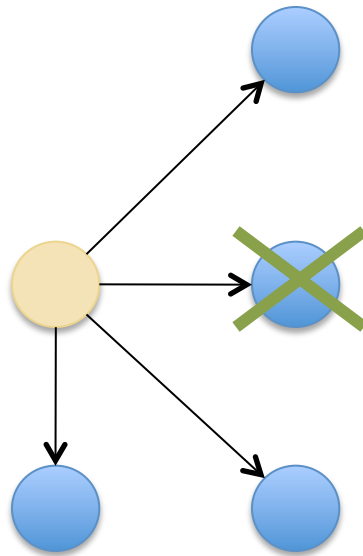
- Resources are available on-demand, but any particular resource may fail at any time
- Applications can absorb new resources
- Applications can tolerate failures

Actor/Reactor Model

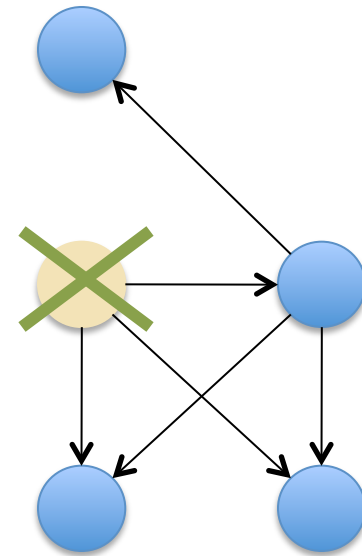
Auto-scale



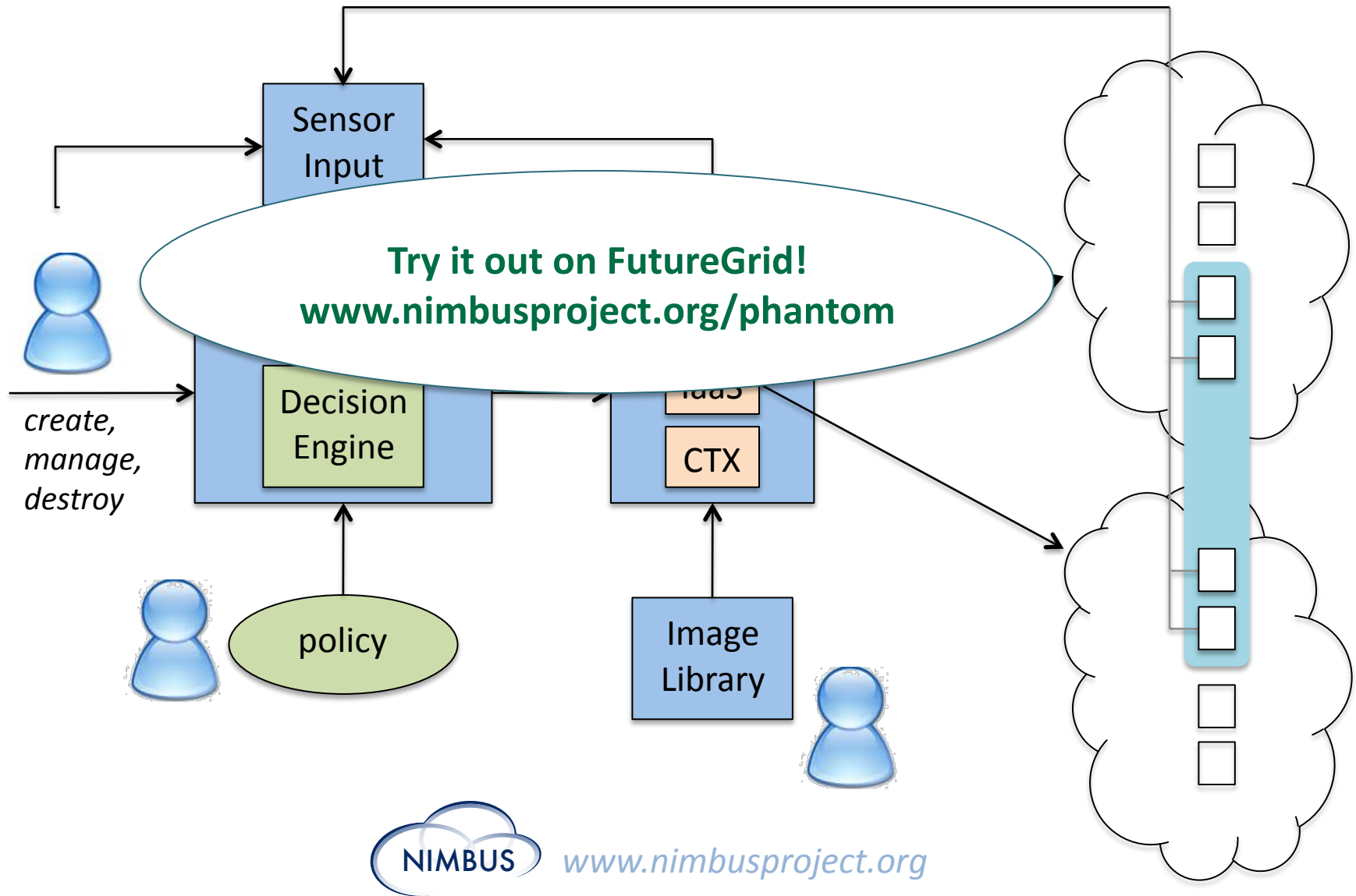
Redeploy



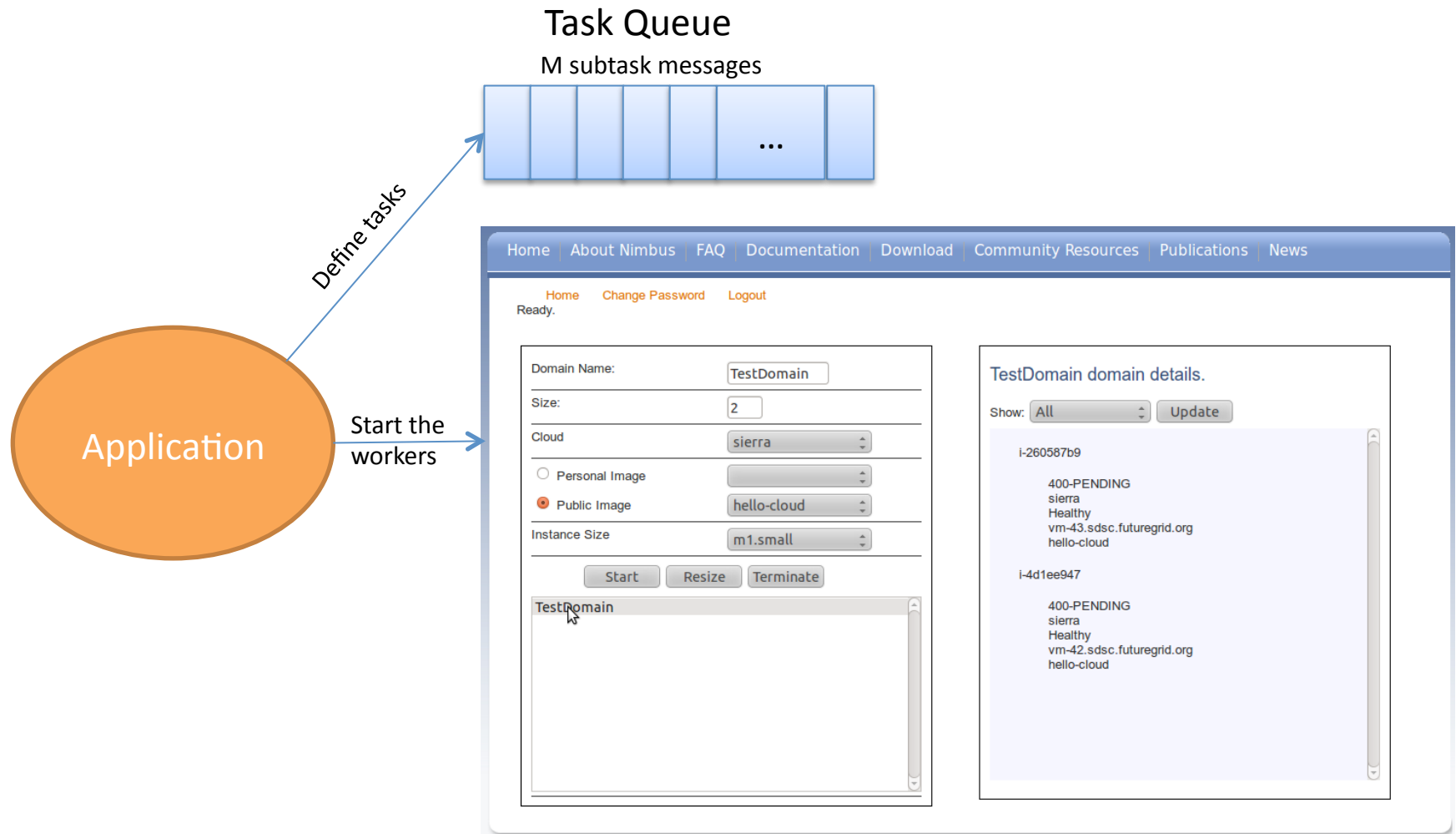
Role transfer + leader election



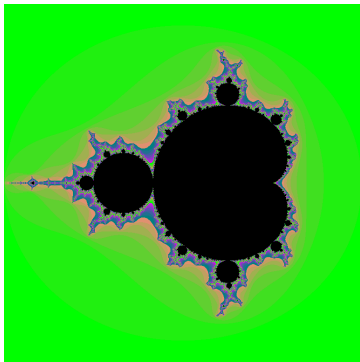
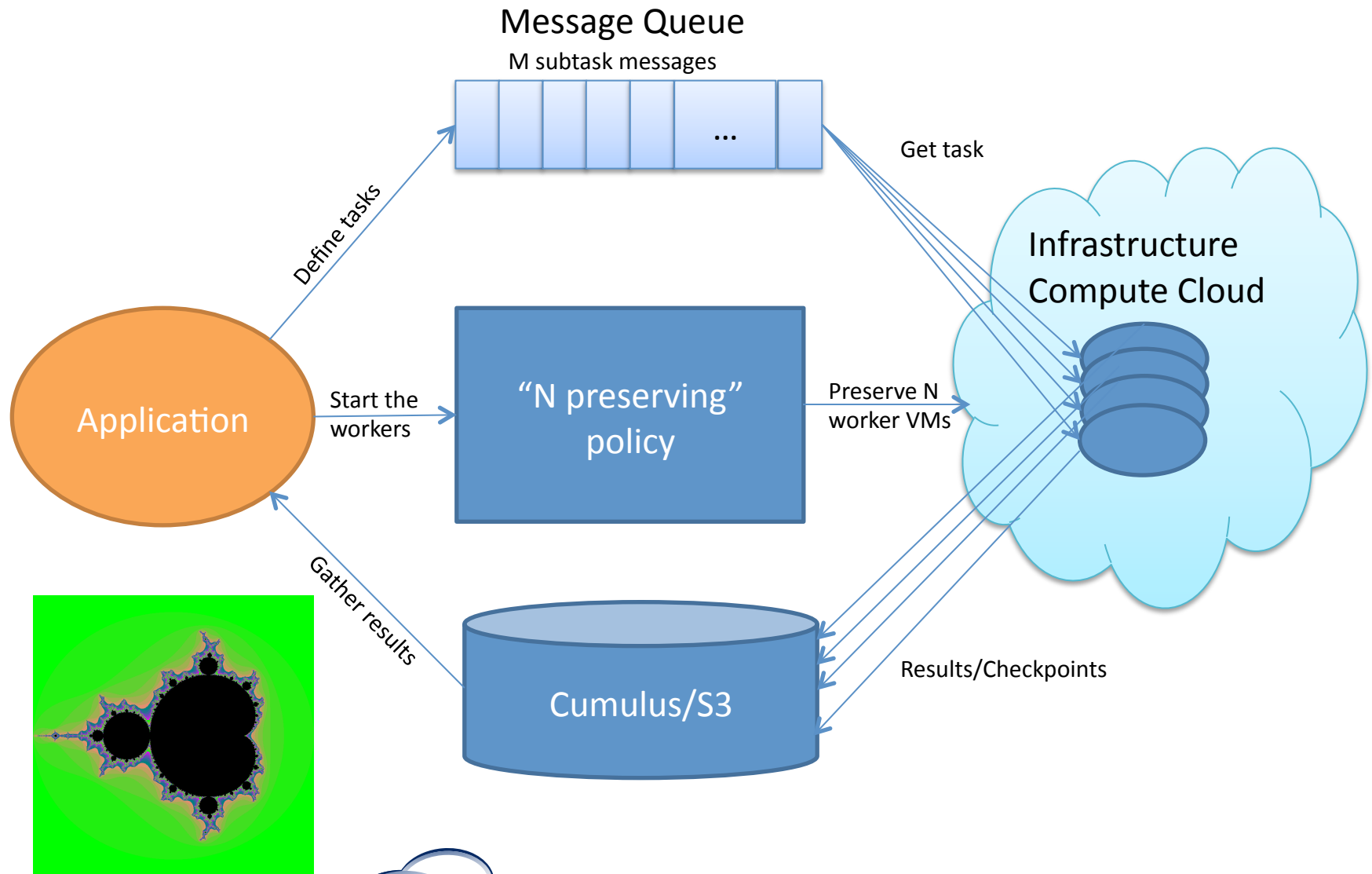
Core Components



Usage Example



Usage Example



Scaling with System Factors

Domains

Add Domain

scienceapp

Configuration for scienceapp

Launch Configuration:

Decision Engine:

Metric:

Cooldown (s):

Minimum:

Maximum:

Scale Up Threshold:

Scale Up By:

Scale Down Threshold:

Scale Down By:

Details

Show:

i-05ec787a

600-RUNNING
ec2.us-east-1
Healthy
ec2-23-20-144-240.compute-1.amazonaws.com
ami-3324e75a
m1.small
ooi
CPUUtilization: Average: 00.45

i-cccd787a

800-TERMINATED
ec2.us-east-1
Healthy
ec2-23-20-144-242.compute-1.amazonaws.com

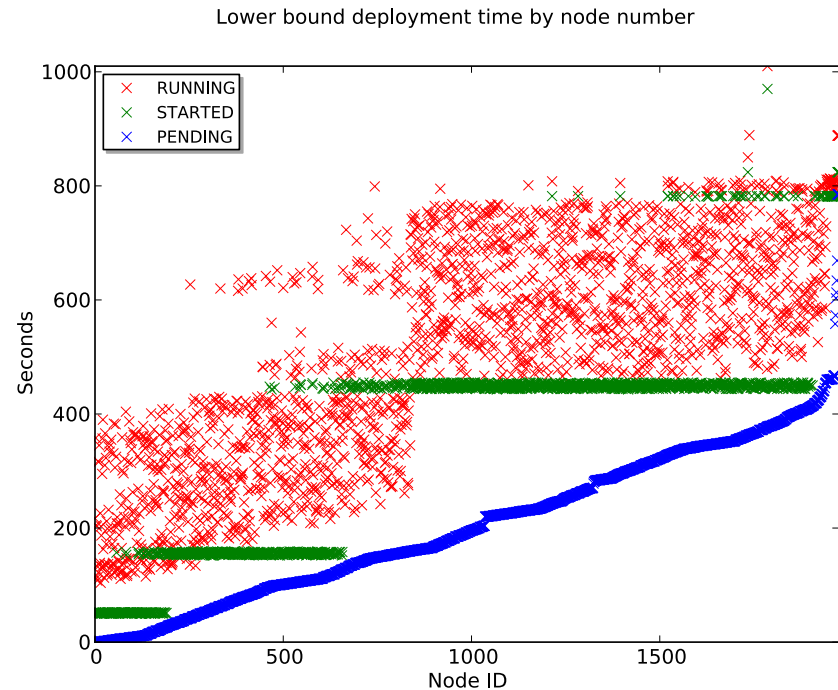
Adventures in Scalability

Mean time between failures

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

Mean time to repair

- Time to repair (TTR)
 - Diagnosis
 - Time to scale (TTS)
 - PENDING (request)
 - STARTED (deployment)
 - RUNNING (contextualization)



TTS: preliminary results for 2,000 VMs provisioned on AWS EC2

Parting Thoughts

- Science has unique outsourcing needs
 - Observatories, experiment support, conference deadlines, fluctuating workload, growth management and others
- Infrastructure clouds are a disruptive innovation
 - Opportunity: on-demand availability
 - But also many challenges!
- “Crossing the chasm”
 - We are living in the age of early adopters
 - Broadening the set:
 - Complexity and scale
 - New models
 - Performance and features
 - The challenges are in crossing the chasm
- Impact on Applications
 - The ability to absorb resources is critical
 - Fault-tolerance is increasingly a requirement
 - Emphasis on programming models that support them

The Nimbus Team



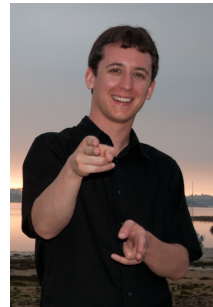
Patrick Armstrong
University of Chicago



John Bresnahan
Argonne National Lab



David LaBissoniere
University of Chicago



Pierre Riteau
University of Chicago



Kate Keahey
Argonne National Lab

...and looking for a postdoc!