Experiences Using Cloud Computing for A Scientific Workflow Application

Jens Vöckler, Gideon Juve, Ewa Deelman,
Mats Rynge, G. Bruce Berriman

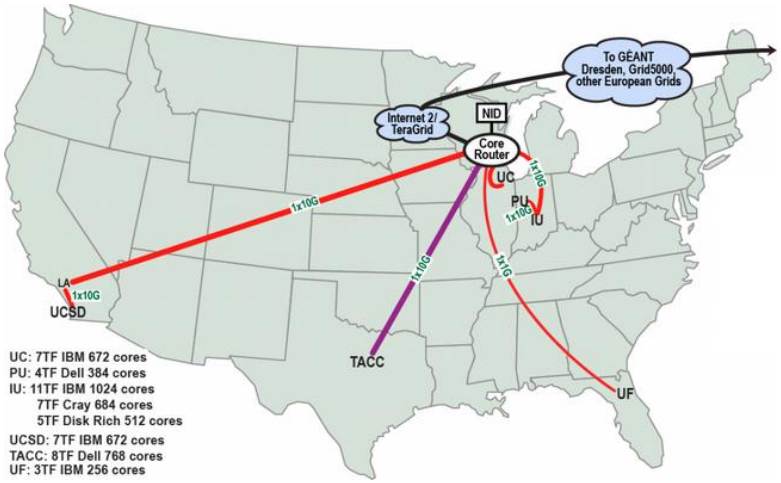
Funded by NSF grant OC 0910812

This Talk

- **Experience** in cloud computing talk
  - FutureGrid:
    - Hardware
    - Middlewares
  - Pegasus-WMS
  - Periodograms
  - Experiments
    - Periodogram I
    - Comparison of clouds using periodograms
    - Periodogram II

What is FutureGrid

- Something Different For Everyone
  - Test bed for ***Cloud*** Computing (this talk).
- 6 centers across the nation
  - Nimbus
  - Eucalyptus
  - Moab “bare metal”
- Start here:
  - http://www.futuregrid.org/



What Comprises FutureGrid

Resource	Type	Cores	Host CPU + RAM
IU *india*	iDataPlex	1,416	8 x 2.9 GHz Xeon @ 24 GB
IU *xray*	Cray XT5m	672	2 x 2.6 GHz 285 (E6) @ 8 GB
UofC *hotel*	iDataPlex	672	8 x 2.9 GHz Xeon @ 24 GB
UCSD *sierra*	iDataPlex	672	8 x 2.5 GHz Xeon @ 32 GB
UFl *foxtrot*	iDataPlex	256	8 x 2.3 GHz Xeon @ 24 GB
TACC *alamo*	PowerEdge	1,016	8 x 2.6 GHz Xeon @ 12 GB
TOTALS		4,704	

- Proposed:
  - 16 x (192 GB + 12 TB / node) cluster
  - 8 node GPU-enhanced cluster

Middlewares in FG

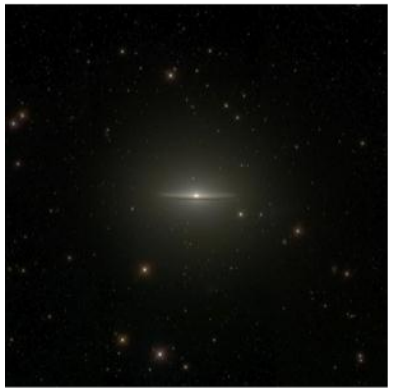
Resource	Type	Cores	Moab	Eucalyptus	Nimbus
IU *india*	iDataPlex	1,416	832 (59%)	400 (28%)	-
IU *xray*	Cray XT5m	672	672 (100%)	-	-
UofC *hotel*	iDataPlex	672	336 (50%)	-	336 (50%)
UCSD *sierra*	iDataPlex	672	312 (46%)	120 (18%)	160 (24%)
UFl *foxtrot*	iDataPlex	256	-	-	248 (97%)
TACC *alamo*	PowerEdge	1,016	896 (88%)	-	120 (12%)
TOTALS		4,704	3,048 (65%)	520 (11%)	744 (18%)

Available resources as of 2011-06-06

Pegasus WMS I

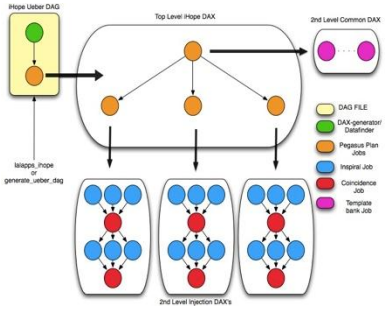
Automating Computational Pipelines

- Funded by NSF/OCI, is a collaboration with the Condor group at UW Madison
- Automates data management
- Captures provenance information
- Used by a number of domains
  - Across a variety of applications
- Scalability
  - Handle large data (kB…TB), and
  - Many computations (1…10^6 tasks)



Pegasus WMS II

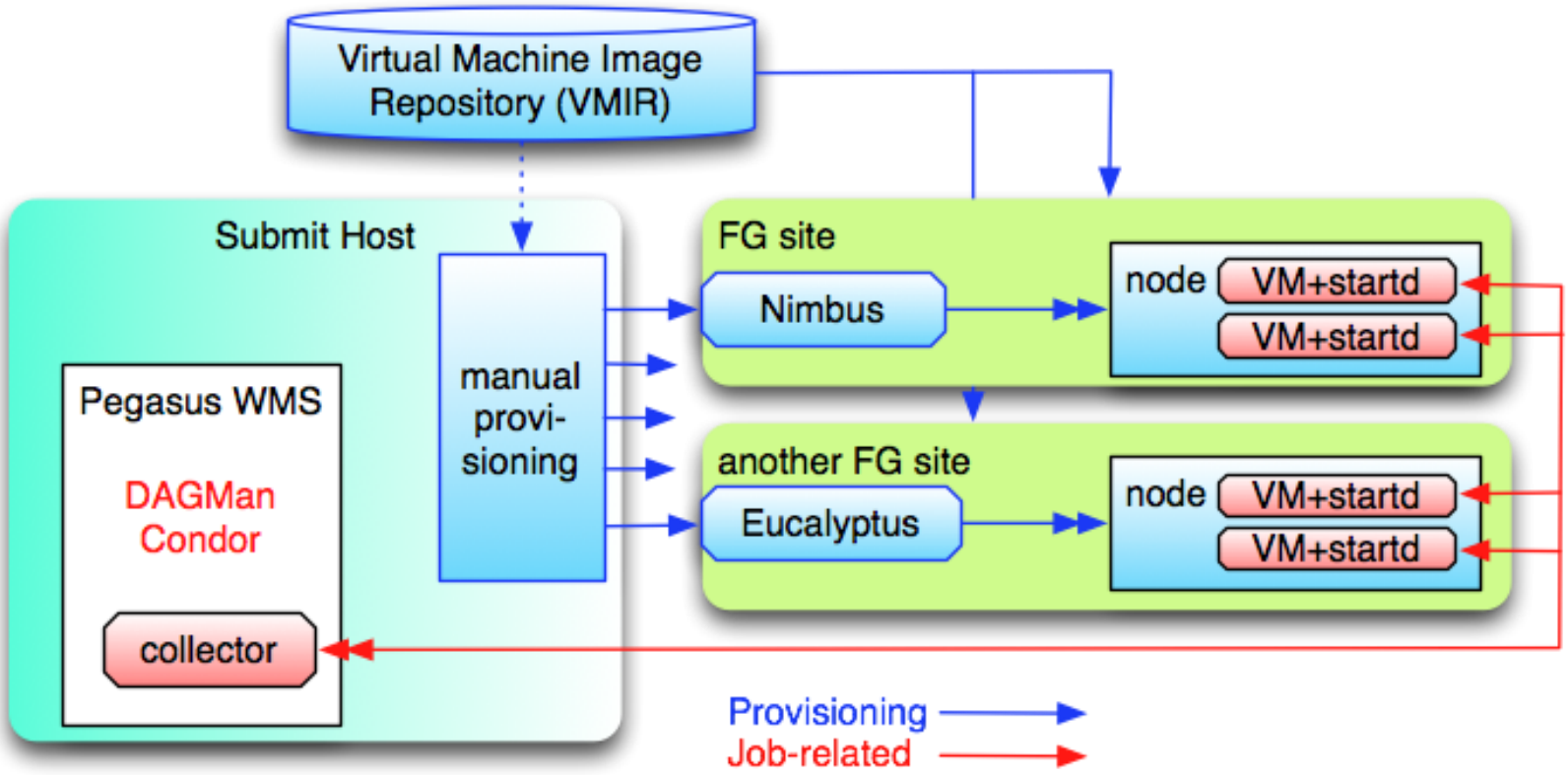
- Reliability
- Retry computations from point of failure
- Construction of complex workflows
  - Based on computational blocks
  - Portable, reusable WF descr.
- Can run pure locally, or
- Distributed among institutions
  - Laptop, campus cluster, grid, cloud



How Pegasus Uses FutureGrid

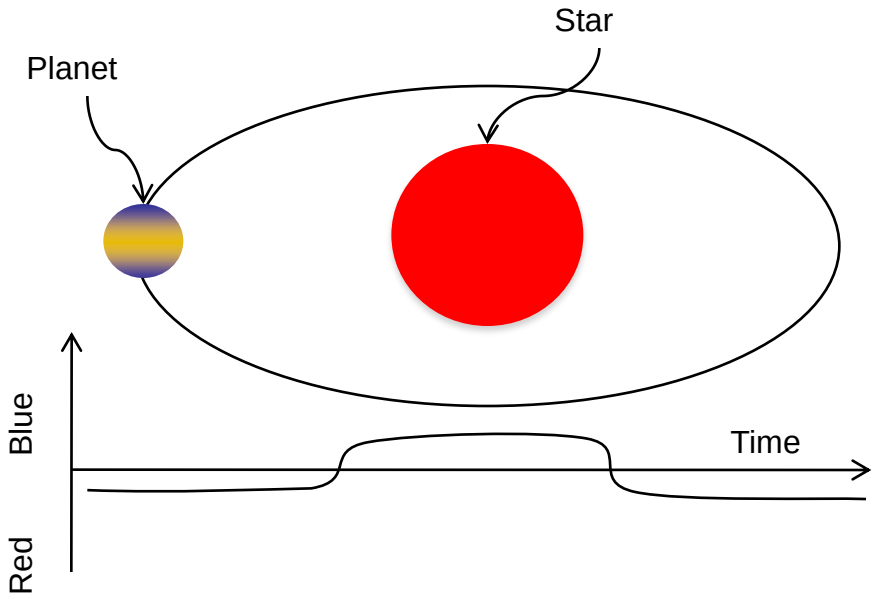
- Focus on *Eucalyptus* and *Nimbus*
  - No *Moab "bare metal"* at this point
- During Experiments in Nov' 2010
  - 544 Nimbus cores
  - 744 Eucalyptus cores
  - 1,288 total potential cores
    - across 4 clusters
    - in 5 clouds.
  - Actually used 300 physical cores (max).

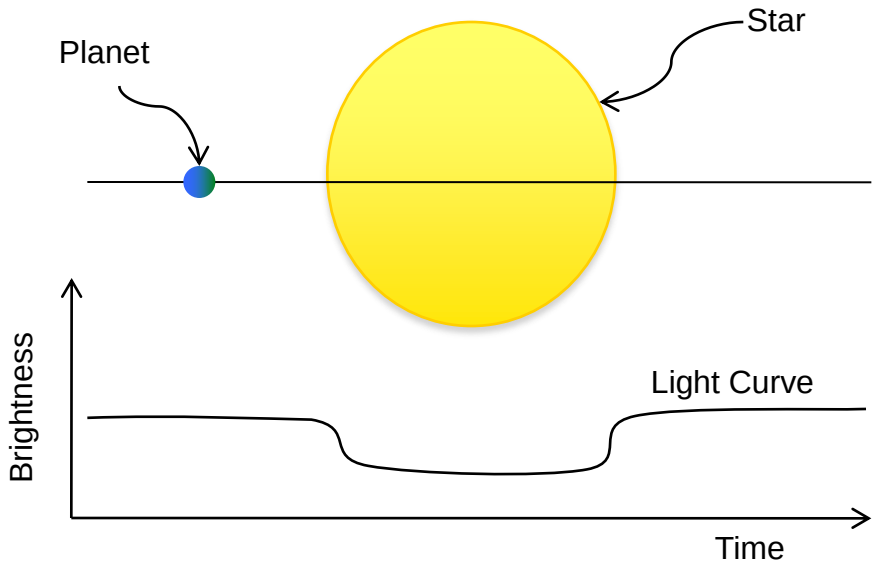
Pegasus FG Interaction



Periodograms

- Find extra-solar planets by
  - Wobbles in radial velocity of star, or
  - Dips in star's intensity





Kepler Workflow

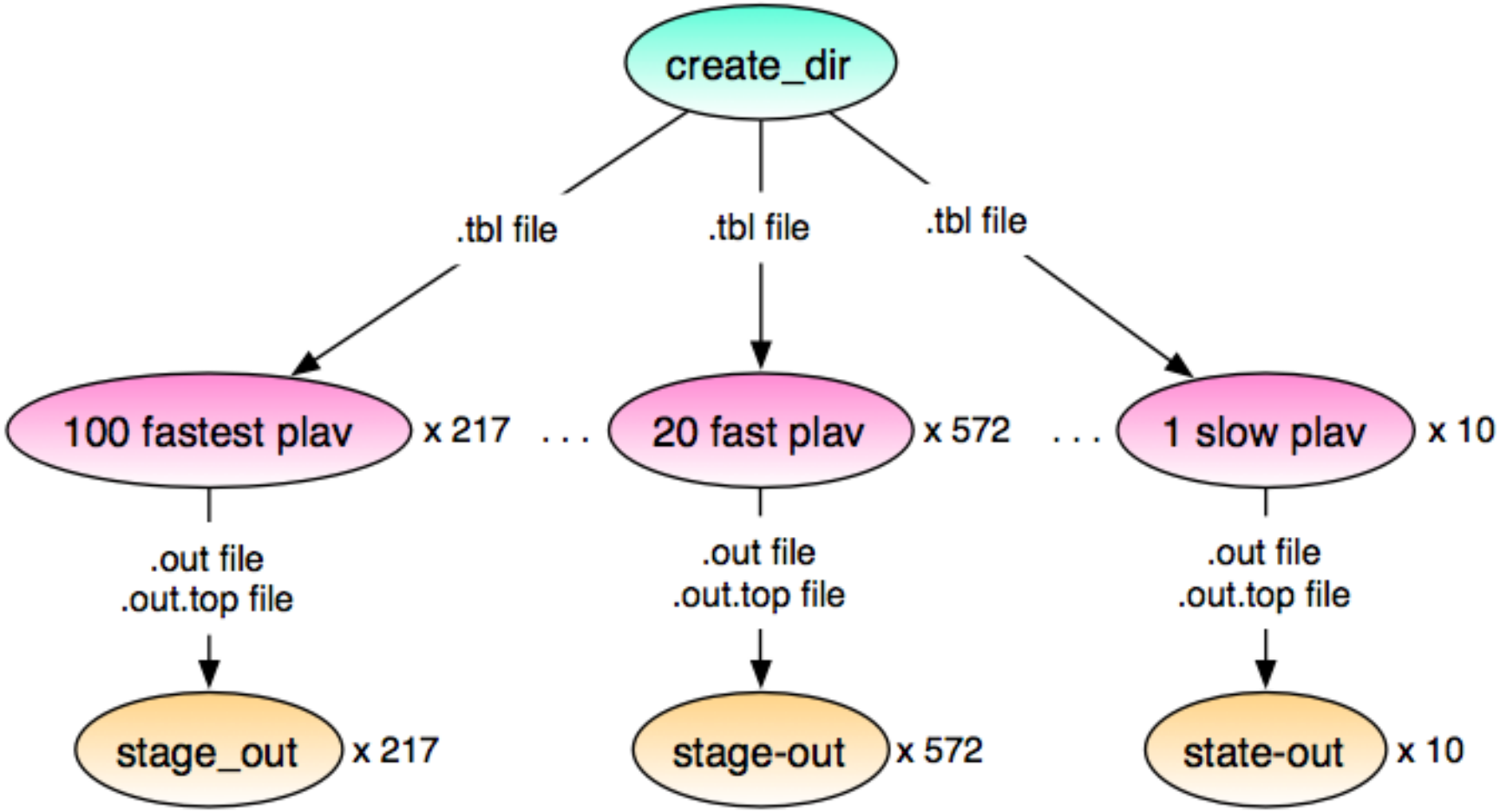
- 210k light-curves released in July 2010
- Apply 3 algorithms to each curve
- Run entire data-set
  - 3 times, with
  - 3 different parameter sets

- This talk's experiments:
  - 1 algorithm, 1 parameter set, 1 run
  - Either partial or full data-set

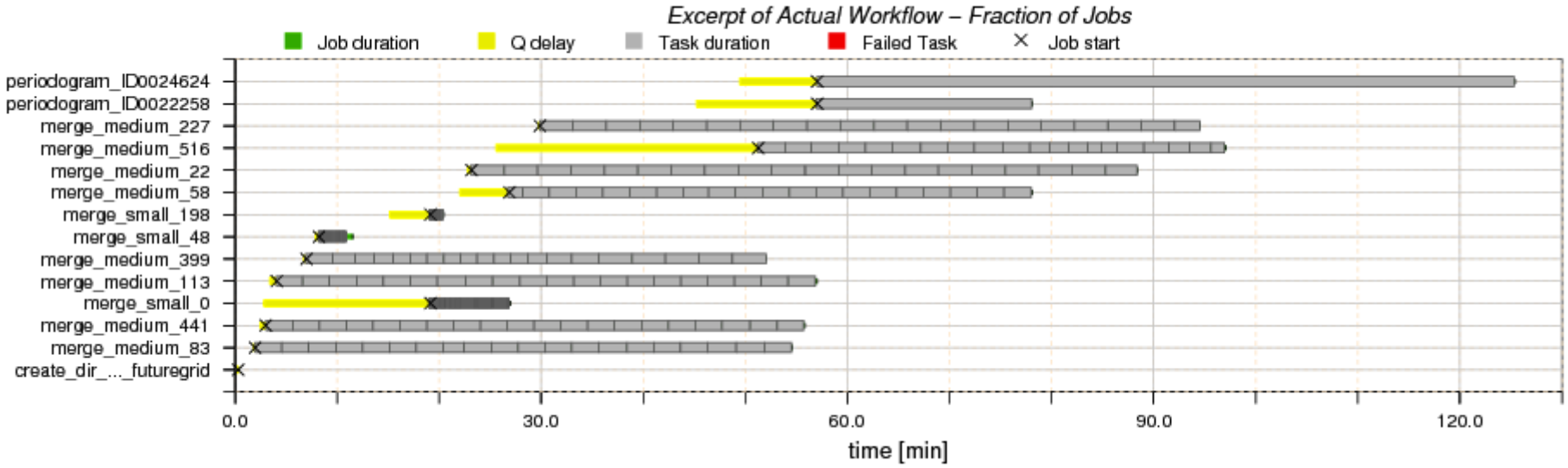
Pegasus Periodograms

- 1st experiment is a “ramp-up”
  - Try to see where things trip
    - 16k light curves
    - 33k computations (every light-curve twice)
  - Already found places needing adjustments
- 2nd experiment also 16k light curves
  - Across 3 comparable infrastructures
- 3rd experiment runs full set
  - Testing hypothesized tunings

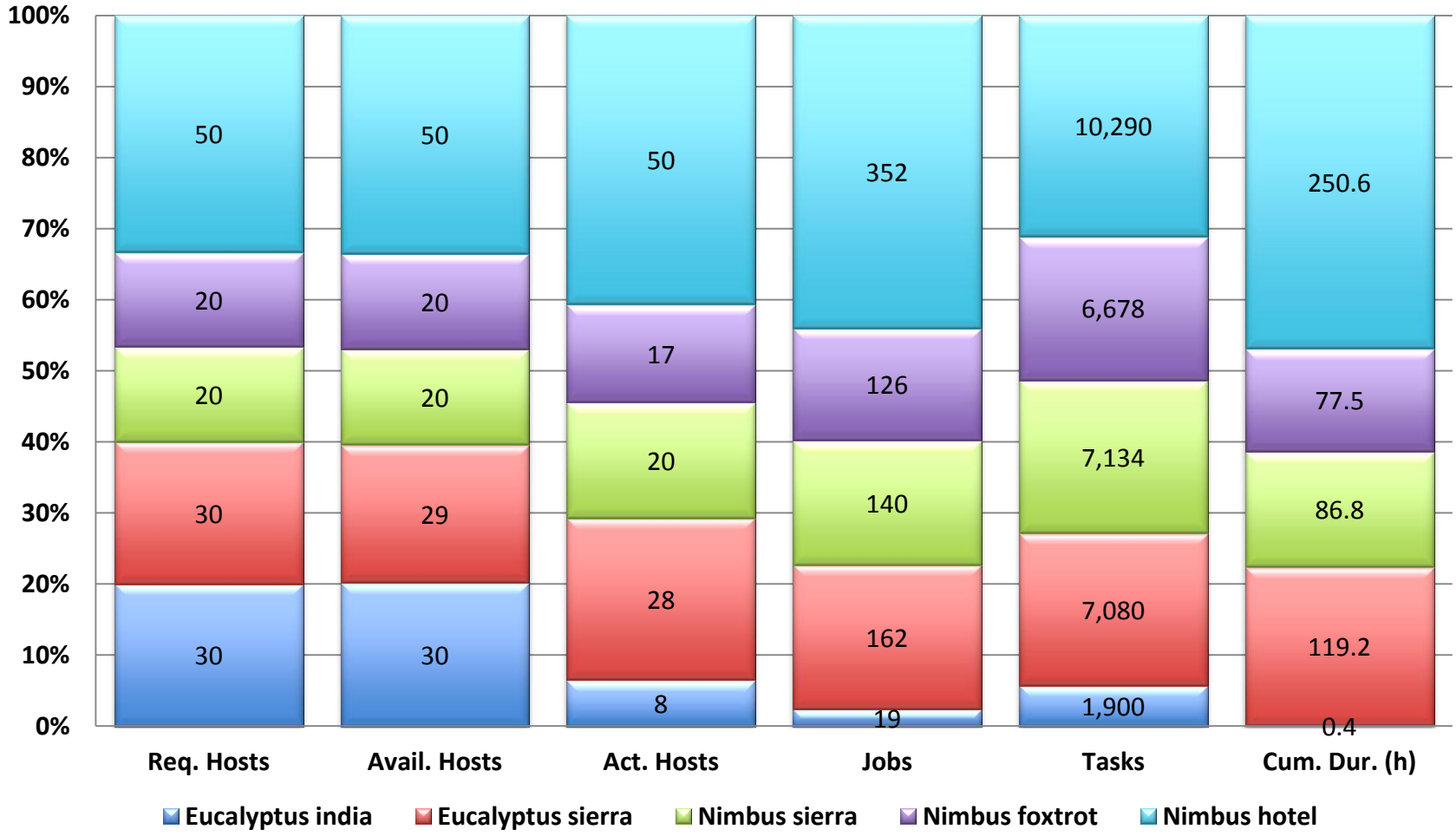
Periodogram Workflow



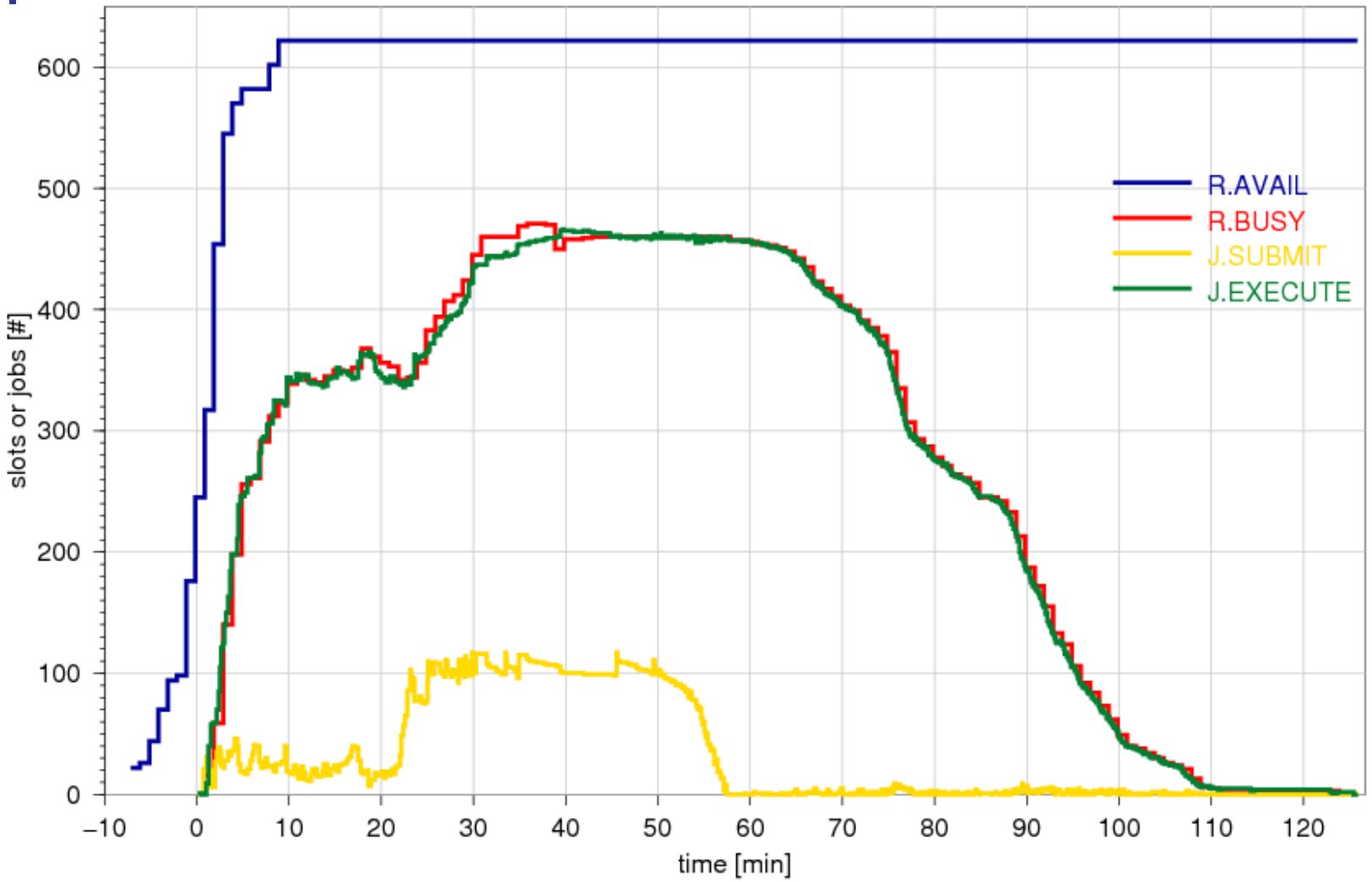
Excerpt: Jobs over Time



Hosts, Tasks, and Duration (I)



Resource- and Job States (I)



Cloud Comparison

- Compare academic and commercial clouds
  - NERSC's Magellan cloud (Eucalyptus)
  - Amazon's cloud (EC2), and
  - FutureGrid's *sierra* cloud (Eucalyptus)
- Constrained node- and core selection
  - Because AWS costs $$
  - 6 nodes, 8 cores each node
  - 1 Condor slot / physical CPU

Cloud Comparison II

Site	CPU	RAM (SW)	Walltime	Cum. Dur.	Speed-Up
Magellan	8 x 2.6 GHz	19 (0) GB	5.2 h	226.6 h	43.6
Amazon	8 x 2.3 GHz	7 (0) GB	7.2 h	295.8 h	41.1
FutureGrid	8 x 2.5 GHz	29 (½) GB	5.7 h	248.0 h	43.5

- Given 48 physical cores
  - Speed-up ≈ 43 considered *pretty good*
  - AWS cost ≈ $31
  - 7.2 h x 6 x c1.large ≈ $29
  - 1.8 GB in + 9.9 GB out ≈ $2

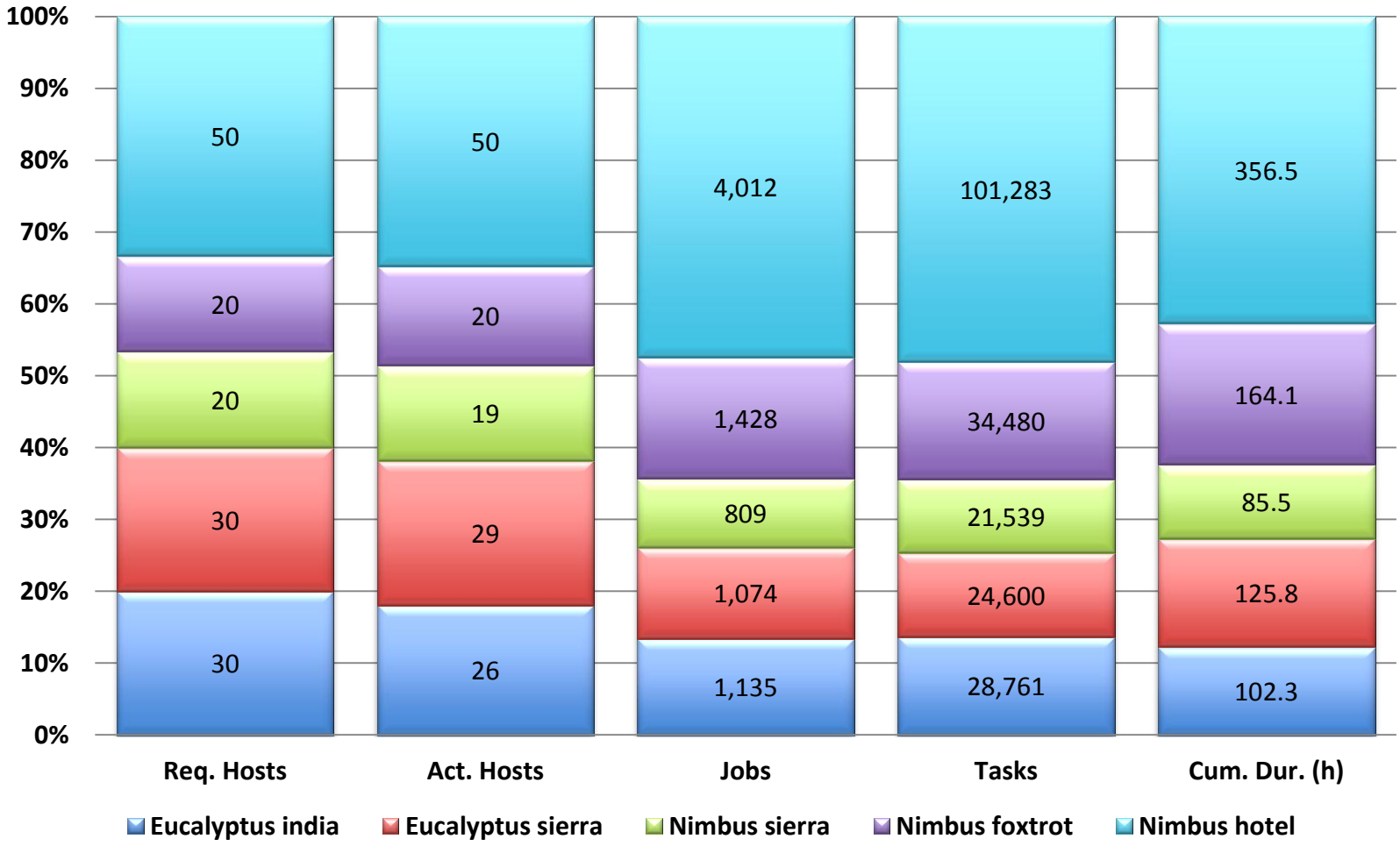
Scaling Up I

- Workflow optimizations
  - Pegasus clustering ✔
  - Compress file transfers
- Submit-host Unix settings
  - Increase open file-descriptors limit
  - Increase firewall's open port range
- Submit-host Condor DAGMan settings
  - Idle job limit ✔

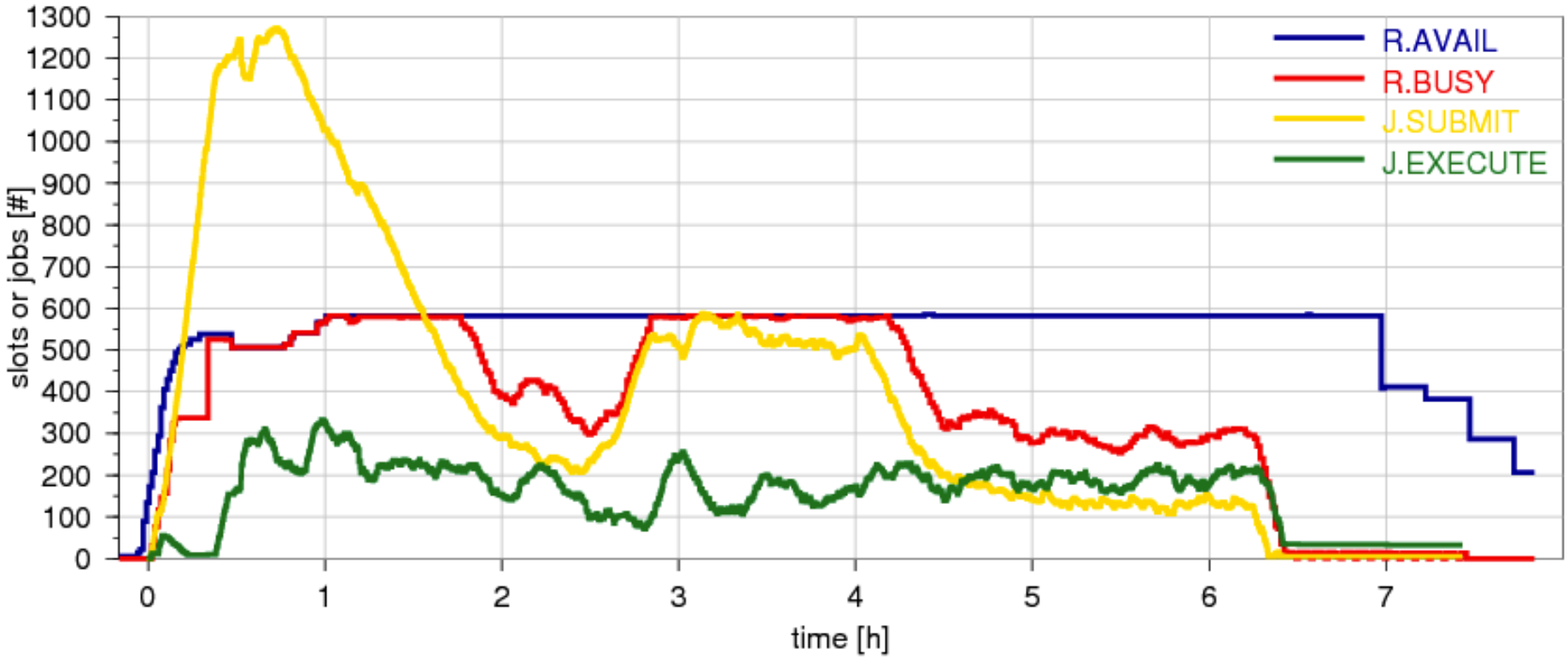
Scaling Up II

- Submit-host Condor settings
  - Socket cache size increase
  - File descriptors and ports per daemon
    - Using *condor_shared_port* daemon
- Remote VM Condor settings
  - Use CCB for private networks
  - Tune Condor job slots
  - TCP for *collector* call-backs

Hosts, Tasks, and Duration (II)

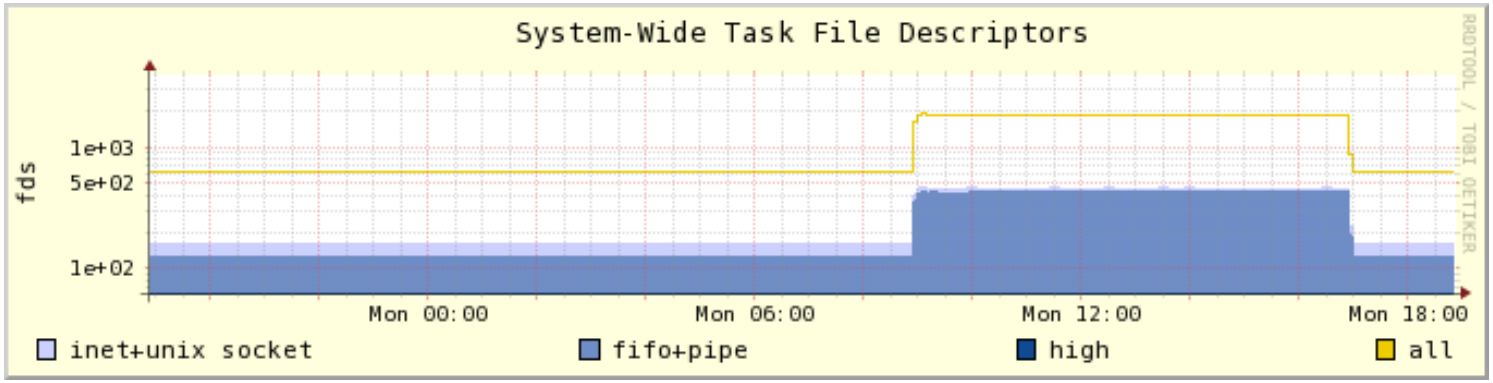


Resource- and Job States (II)



Lose Ends

- Saturate requested resources
- Clustering
- Better submit host tuning
  - Requires better monitoring ✔



- Better data staging

Acknowledgements

Funded by NSF grant OC 0910812

- Ewa Deelman,
- Gideon Juve,
- Mats Rynge,
- Bruce Berriman
- FG help desk ;-)

INFORMATION SCIENCES INSTITUTE

agent of innovation

USC Viterbi School of Engineering

ipac

http://pegasus.isi.edu/

Future Grid