

Flying Low: Simple Leases with the Workspace Pilot

Euro-Par 2008, Las Palmas de Gran Canaria

Tim Freeman (tfreeman@mcs.anl.gov) *

Kate Keahey (keahey@mcs.anl.gov)

University of Chicago

Argonne National Laboratory

Resource Leasing

- Should be a “first class citizen”
 - not a side effect of job submission
 - emerging as a fundamental abstraction
- Cluster partitions
- Pilot jobs
 - Condor glide-ins, MyCluster, Falkon, PanDA Pilot
- Virtual Machines

Virtualization

- There is no silver bullet but virtualization is an attractive technology
- Overhyped period is fast becoming history
 - years of collective experience bearing fruits
- Maturing tools, maturing users

VM Advantages for Resource Providers

- **Isolation**
 - VMM abstraction provides a solid barrier between each user's activity and the physical resource
 - easily kill
- **Enforcement**
 - control/measure consumption
- **Minimal intervention**
 - application environment is fully configured already
 - no miscellaneous/conflicting site integrations

VM Advantages for Clients

- **Portability**
 - minimal effort to run on a new site
- **Consistency**
 - same software, compilers, behaviors
- **Resource options**
 - more resources will be available if less administrator time is needed
 - “lulls” in site usage can be more quickly reacted to
 - migration is possible
- **Cost/benefit**
 - performance may suffer but is often tolerable (*cf. Xen*), advantages can often outweigh concerns

Integration Challenges

- Virtualization adoption from the main Grid stakeholders has been relatively slow
 - though many “statically” deployed VMs
- Commitments
 - won't reduce current client throughput
- Modes of operation
 - preserve major client functionality such as batch jobs – even for the full cluster if necessary
 - preserve existing technology (GPFS, PVFS, etc)
- Small deltas
 - complexity is error-generating and time-consuming

A VM can not be a normal pilot job

- Launching VMs is a privileged operation
 - Workspace-control component, special account
- Network allocations and security
- Contextualization needs
 - other patching/customizations
- Monitoring / Adjustment
 - remote status/reboot
 - console output
- File management
 - caching/scheduling
 - free space creation

Workspace Pilot

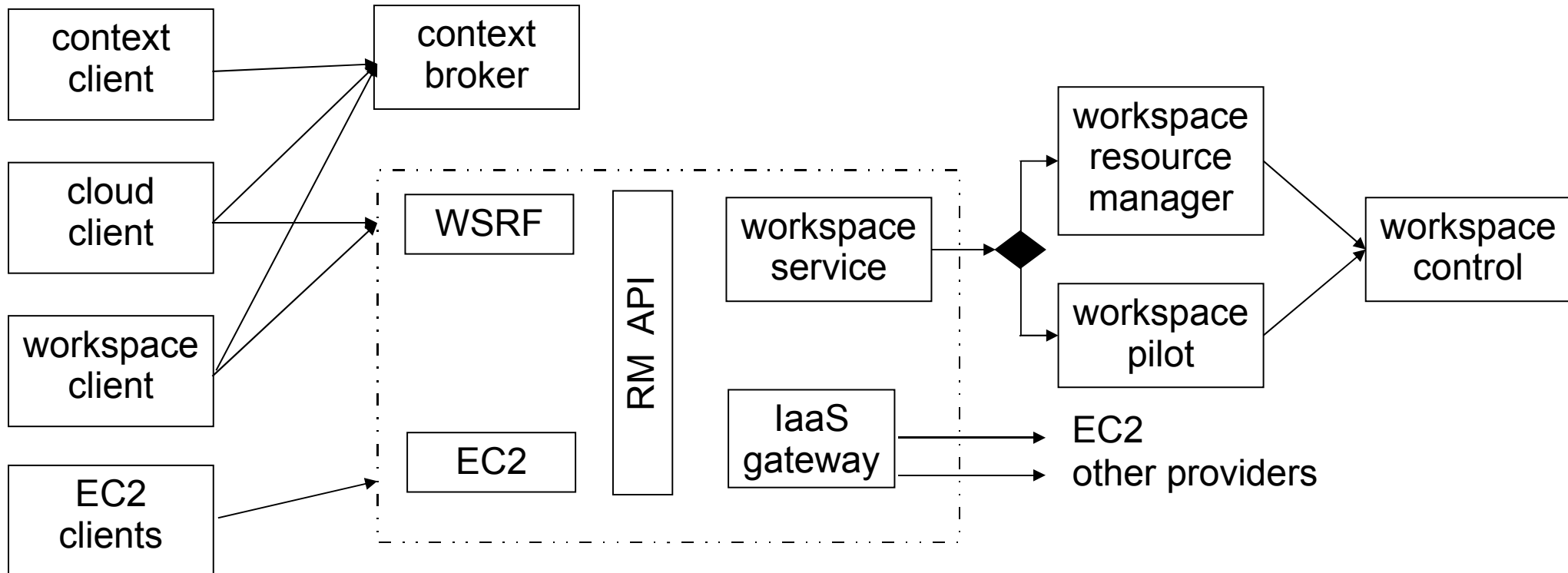
- “Glide-in” approach: submit a pilot program that claims a resource slot
- Works with popular LRMs
 - currently tested with PBS (*Torque*)
 - adapter for submission operation (*qsub/qdel*)
- Fault tolerant
 - VM service can be offline and pilot will still clean up
- Administrator tools
 - “kill-all” utility
- Standalone
 - Nimbus component



Nimbus

- Using straightforward clients
 - cloud client
 - EC2 clients
- ... employing programmatic, remote interfaces
 - WSRF
 - EC2 WSDL
- ... users can launch custom VMs
 - per-deployment allocations / custom environments
 - administrator privileges (*site policy decision*)
 - coordinated virtual clusters (*contextualization*)

Nimbus Components

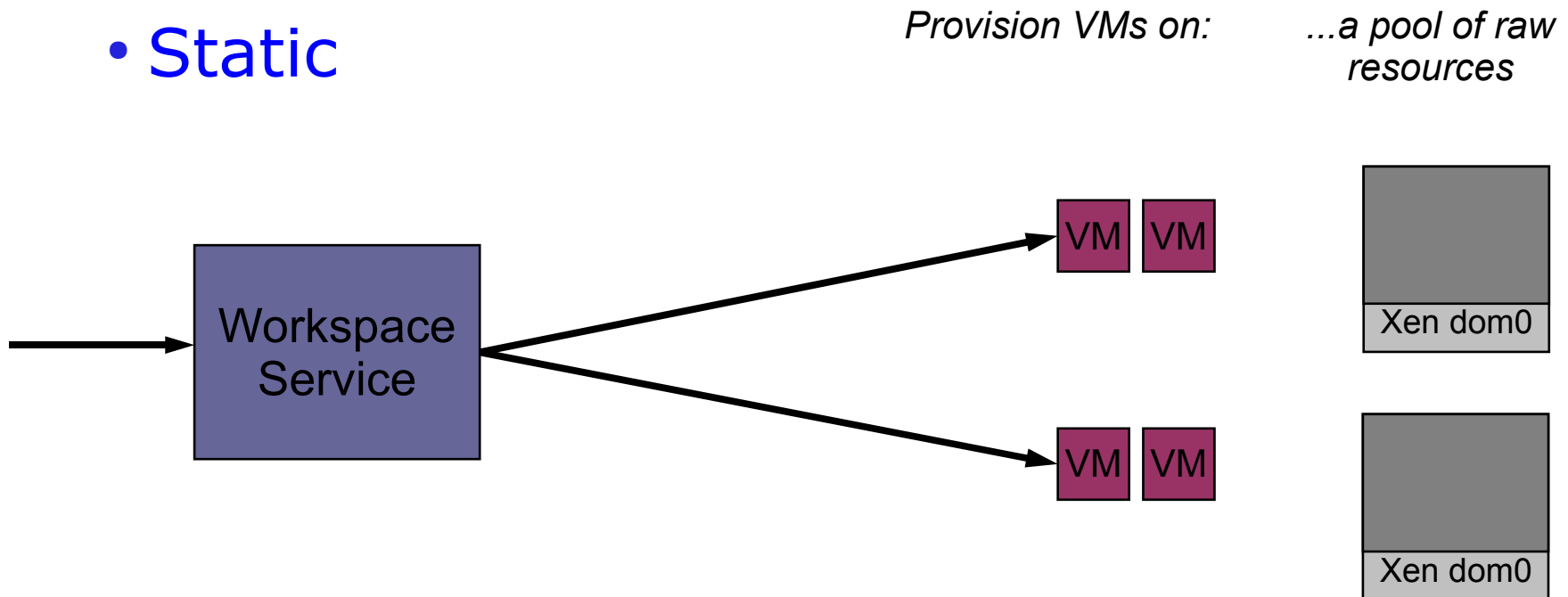


- use something different

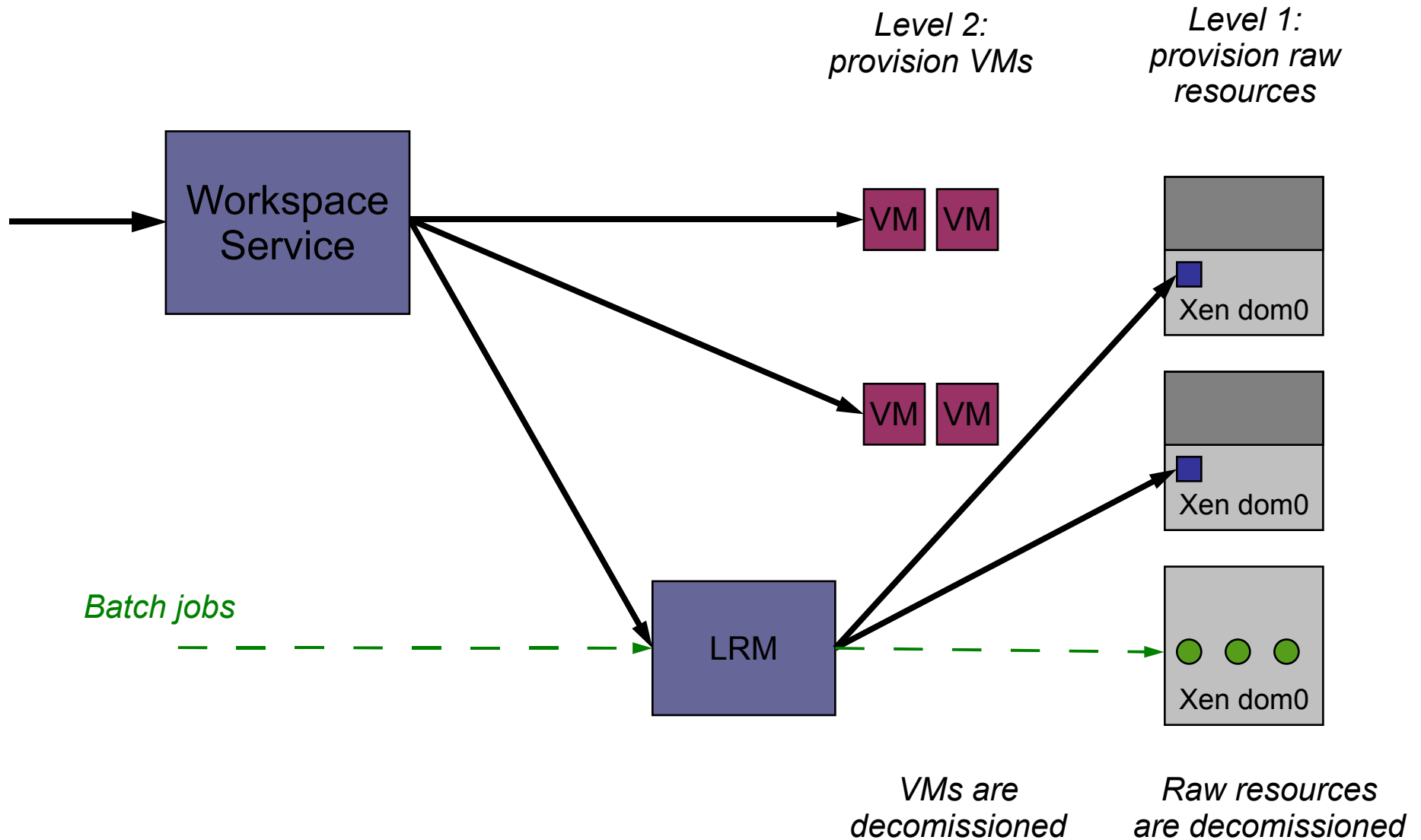
- use pieces elsewhere

Default Resource Selection

- Greedy
- Static



Pilot Based Resource Selection



The Pilot Program

- Reduces and restores Xen dom0 memory
 - uses balloon driver, other mechanisms easy to add
- Communicates with the Workspace Service
 - HTTP, with digest access authentication
 - timestamped notifications
- Requires sudo, privileged account
 - only other dependency is Python (and VMM)
- SIGTERM: allotted time exceeded
 - service is notified, small grace period for cleanup
 - if nothing happens in allotted time, the pilot terminates “without prejudice”

Lease Semantics

- Limited by LRM
 - “best effort”
 - advanced reservations uncommon or not allowed by site policies (hoping this will change)
- Co-scheduled requests still work
 - users want co-scheduled homogeneous and heterogeneous groups of VMs
- Overlaid policies
 - Workspace Service gets raw resources, not VMs
- Current policy
 - one VM per physical node

Accounting and Authorization

- LRM records only show the nimbus account
- Detailed usage recording
 - all use recorded by the accounting module (plugin)
 - service logs show nodes used, times etc.
- Detailed policies
 - arbitrary plugin, but Nimbus provides defaults
 - all-time duration based authorization
 - per-deployment bounds
 - image authorization
 - create policy templates and assign users to one
 - changes effective immediately (no restarts)

Evaluation

- Concentrated on mechanism overhead
 - job performance out of scope
 - many performance studies, results application specific
- Standup and teardown times
 - from client perspective
 - breakdowns for LRM, Workspace Service, Workspace Pilot, and some internals
- Methodology
 - local timestamps, collected afterwards
 - impossible to always use the same clock (but all nodes run NTP)
 - no other LRM jobs in this queue

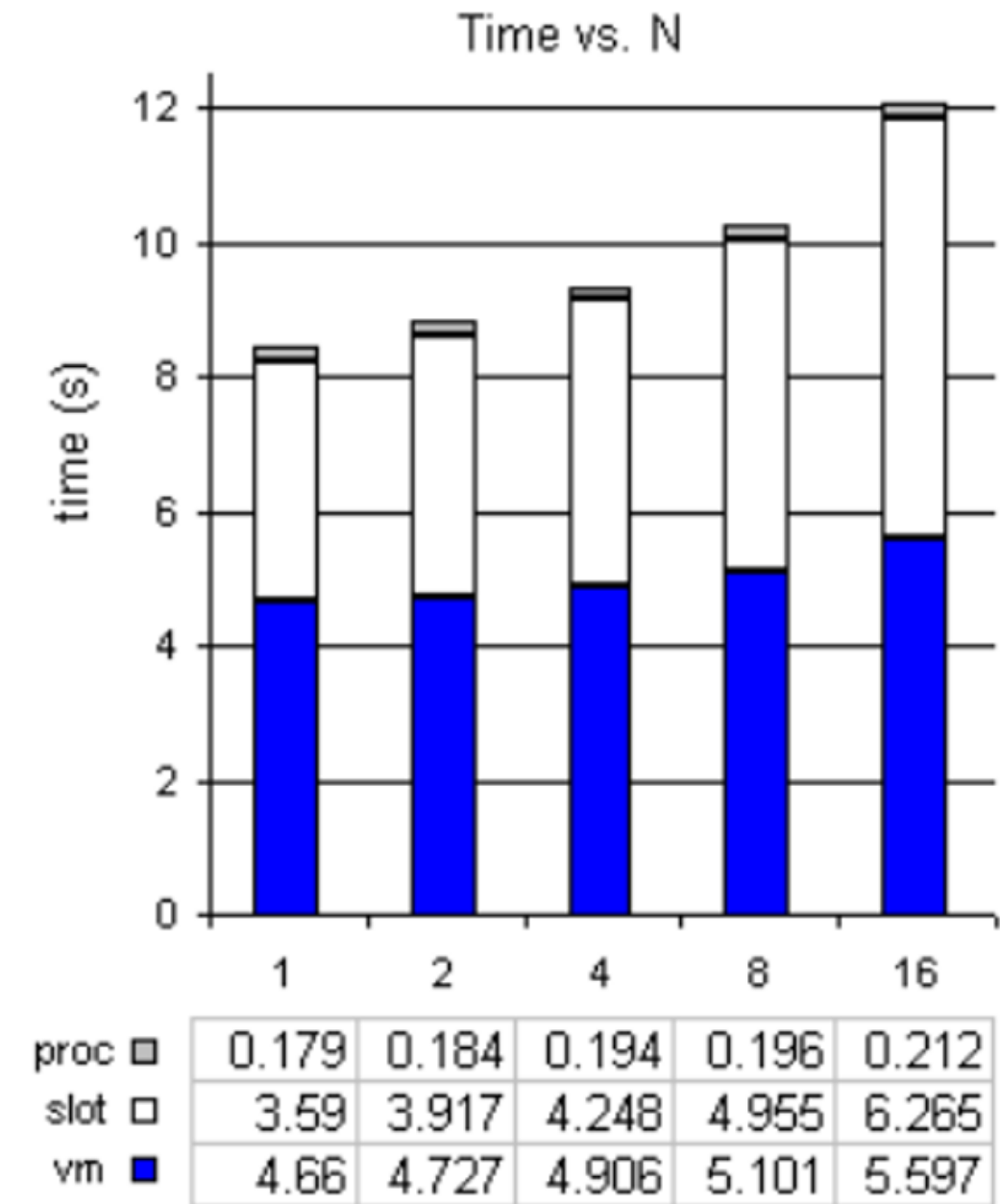
Memory Adjustment

- No correlation for memory size
- 100 trials, 1GB memory
 - no correlation for memory size
- Domain 0 memory reduction
 - 743.6ms (SD 100.5)
 - almost all cost is from Xen mem-set operation
- Domain 0 memory restoration
 - 1317.2ms (SD 96.5)
 - includes system checks like examining running VMs
- Kill-all / emergency backout
 - one running VM
 - 2336.27ms (SD 236.7)

Client end-to-end

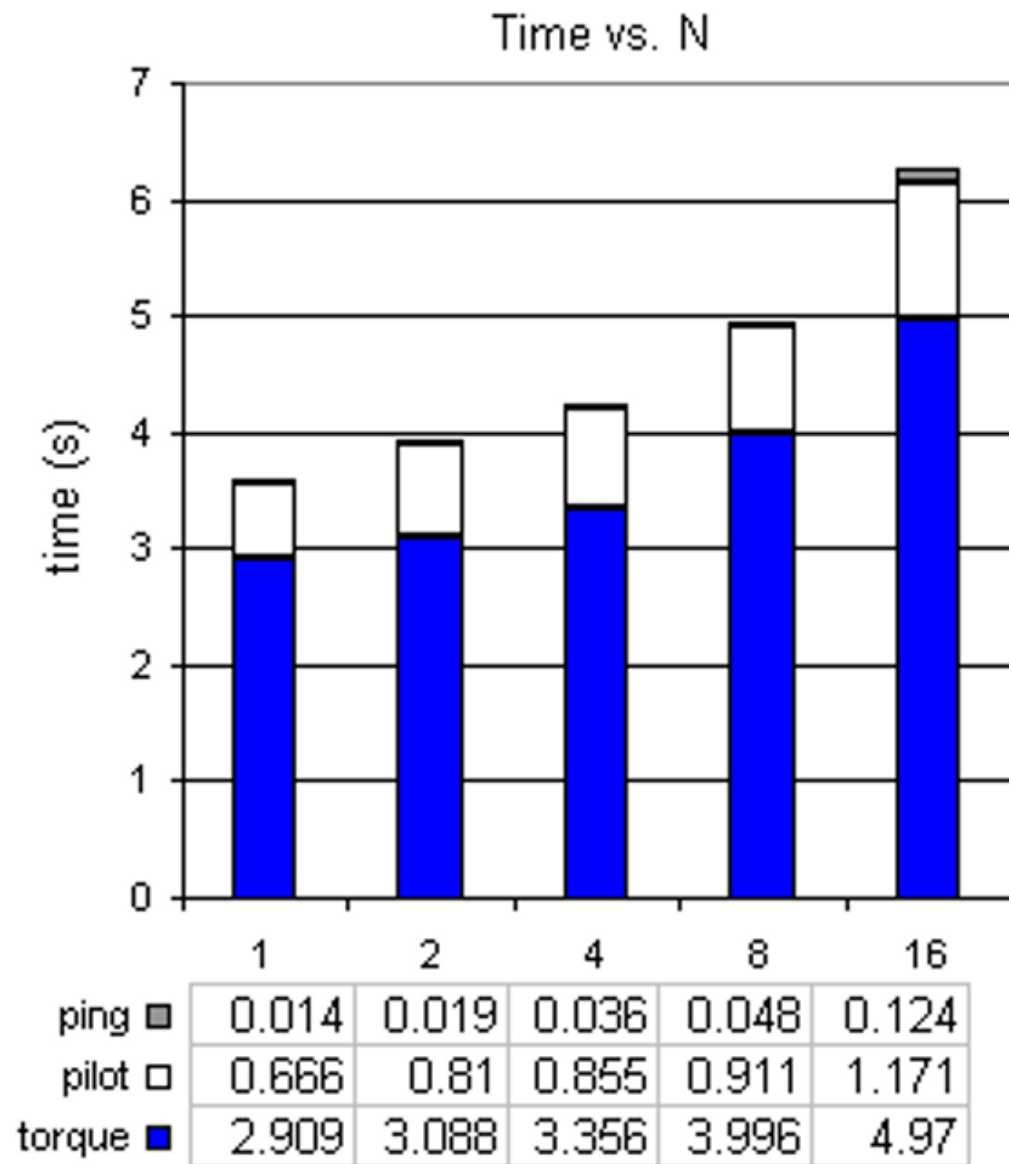
- **proc**: client-side processing
- **slot**: securing the slot
- **vm**: provisioning the VM, LRM independent

Main increase is from slot provisioning



Slot detail (1)

- **ping**: notification time
- **pilot**: from pilot executing to notification
- **torque**: from LRM request to pilot executing (Maui scheduler used)

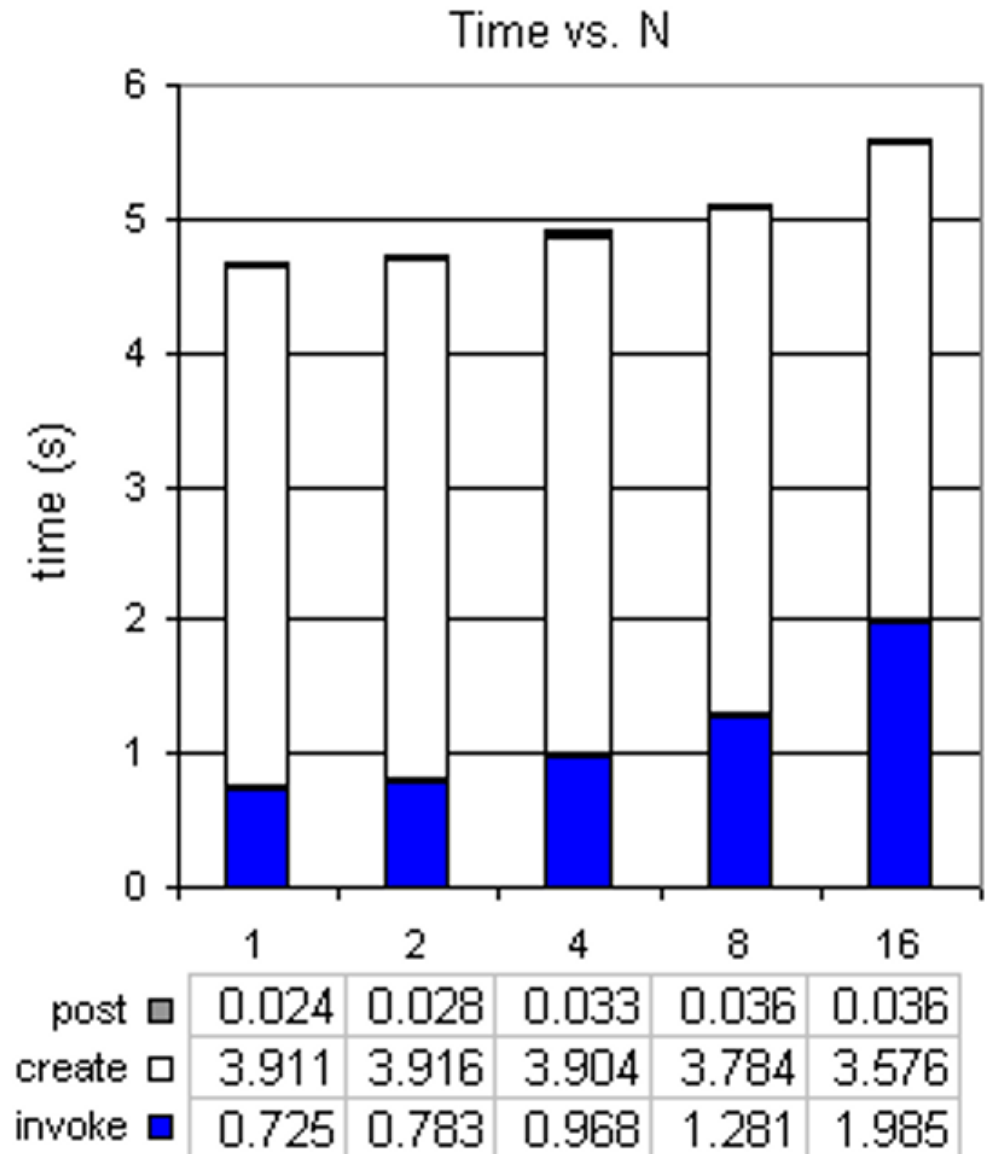


Slot detail (2)

- 'torque' responsible for most of the delay
 - the *pbsdsh* program runs on first node in reservation then loops to launch on the rest
- 'pilot' should not be increasing!
 - stumped us for a minute
 - *sudo* and account system is backed by LDAP which was getting N simultaneous requests
- 'ping' increase (0.124 ms for 16 nodes)
 - service processing N simultaneous requests
 - digest access authentication
 - examining more robust/efficient messaging systems (distributed message queues, etc)

Mechanisms with no LRM (1)

- **post**: client notification time
- **create**: starting VMs and connecting to them network (DHCP/security)
- **invoke**: invocation of backend 'workspace-control' service agent

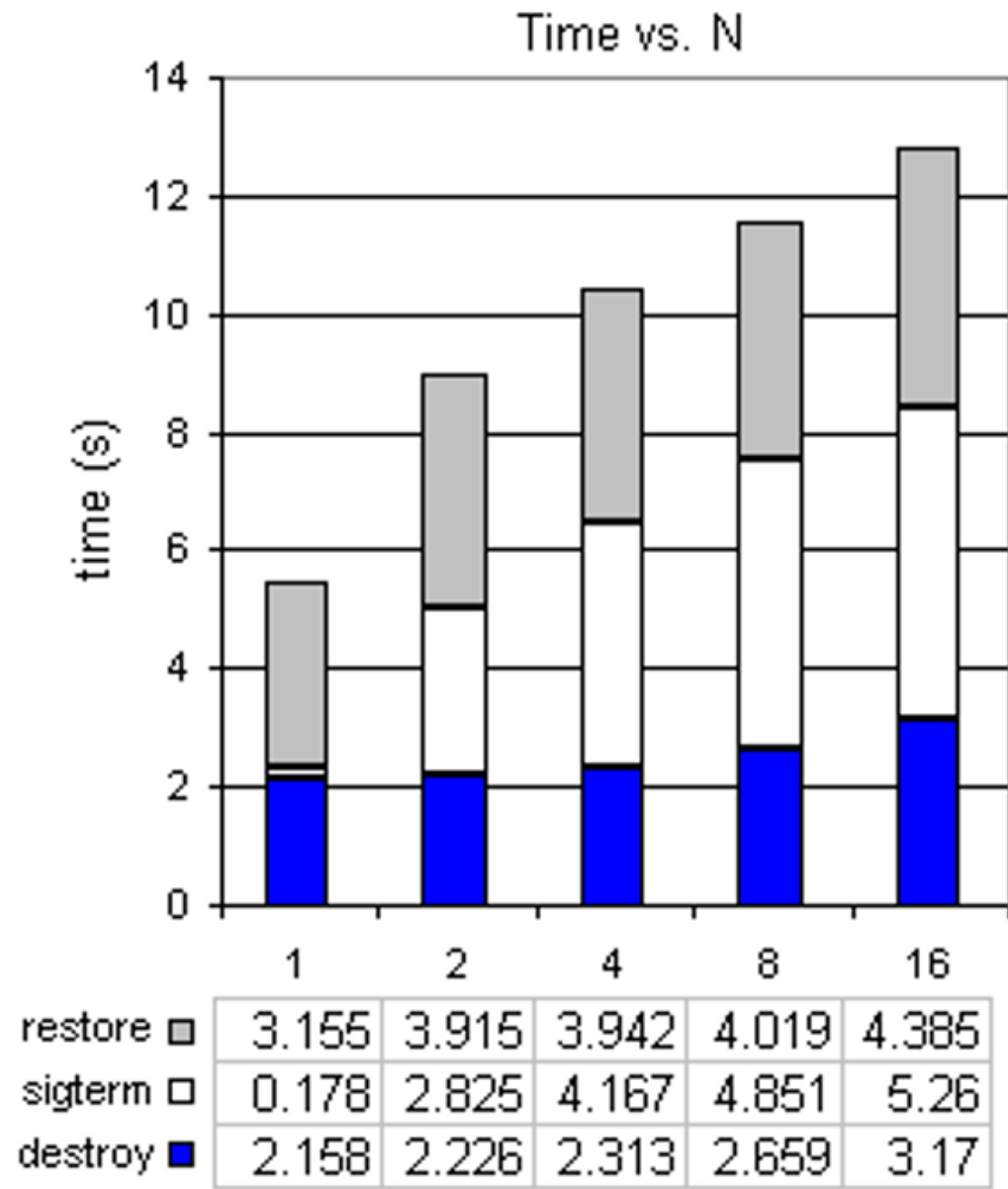


Mechanisms with no LRM (2)

- 'invoke' is the main problem
 - one SSH process per VMM node
 - *sudo*/LDAP issue plays a (smaller) role
- Again, examining robust/efficient messaging
 - especially systems that implement collective communication
 - confident this time can be reduced by at least an order of magnitude, perhaps even more
- Decreasing 'create' time is deceptive
 - should be stable since it is node-local
 - performance difference between nodes
 - since this is an empty queue, torque was assigning same set of nodes for each N value

Destruction (1)

- **restore**: memory adjustment and idling period
- **sigterm**: from qdel request until SIGTERM reaches the pilot(s)
- **destroy**: time for service to cause all VMs to be destroyed



Destruction (2)

- 'restore' includes idling period
 - idling is part of the design, allows for graceful cleanup if the service is running
 - recall earlier memory adjustment discussion
 - *sudo*/LDAP again a problem
 - notifications also, but a lesser role
- LRM ('sigterm') least scalable component
 - sudden jump from N=1 to N=2 is because *pbsdsh* is not used at all with N=1
 - when *pbsdsh* is used there is a delay between SIGTERM on node #1 and the signal propagation to the others

Issues

- Image caching assumed
- Did not cover contextualization costs
- Already discussed:
 - sudo/LDAP
 - messaging overhead, group communication will help
- Large N delays
 - Torque/Maui delays for $N = \{15, 16\}$
 - configuration out of control
 - measurements discarded

Status

- Part of the past four Nimbus releases
- Teraport deployment
 - GPFS issues, working with IBM/Leggett/Cross
- University of Victoria
 - UVIC / ATLAS
 - Gable, Bartle, Grundy, Penfold-Brown
- PanDA (ATLAS)
 - Omer Khalid (CERN) working with PanDA, workspace pilot and workspace control

Related Work

- Pilot systems: Condor glide-ins, MyCluster, Falkon, PanDA Pilot
- DVC, XGE
- Shirako, Vio-Cluster, Maestro-VS, Clusters on the fly

Nimbus

- Create compute clouds
 - new users get going quickly using the *cloud client*
- Use EC2 clients
- Deploy "one-click" auto-configuring virtual clusters
 - long ago hit the "now what?" problem with loose VMs
 - role based, VMs adapt to new networks and security
- Easily experiment with remote protocols and other plugins (networking, scheduling, authorization, VMMs)
- Free, open source
 - Apache2 license
 - active development, contributors

Nimbus

- **Start here:**
 - <http://workspace.globus.org>
- **Science Clouds**
 - growing pool of institutions running their own Nimbus clouds to aid science applications
 - <http://workspace.globus.org/clouds/>
- **Current Release**
 - EC2 client support
 - RM API
 - <http://workspace.globus.org/vm/TP2.0/>
- **Google Summer of Code 2008**
 - KVM

Credits

- **Workspace Team:**

- Kate Keahey (Project Lead)
- Tim Freeman (Senior Developer)

- **With thanks to many collaborators!**

- Jerome Lauret (STAR, BNL), Doug Olson (STAR, LBNL), Marty Wesley (rPath), Stu Gott (rPath), Ken Van Dine (rPath), Predrag Buncic (Alice, CERN), Haavard Bjerke (CERN), Rick Bradshaw (Bcfg2, ANL), Narayan Desai (Bcfg2, ANL), Duncan Penfold-Brown (Atlas, uvic), Ian Gable (Atlas, uvic), David Grundy (Atlas, uvic), Ti Legget (University of Chicago), Greg Cross (University of Chicago), Lizhe Wang (FZK), Marcel Kunze (FZK), Mauricio Tsugawa (UFL), Jose Fortes (UFL), Renato Figueiredo (UFL), Tina Hoffa (Indiana/ISI), Omer Khalid (CERN), Artem Harutyunyan (CERN), Michael Fenn (Clemson), Sebastien Goasguen (Clemson), Ewa Deelman (ISI), Josh Boverhof (LBNL), Leve Hajdu (STAR, BNL), Lidia Didenko (STAR, BNL), David Bartle (Atlas, uvic), Lee Liming (ANL), Frank Wuerthwein (OSG, SDSC), Abhishek Rana (OSG, SDSC), Jeff Chase (Duke), Marlon Pierce (Indiana), Samer Al-Kiswany (UBC), and many others.

- **Funding: NSF SDCI “missing links”**