

Athena on AWS EC2/S3

Jan-Philip Gehrcke, Stefan Kluth, Stefan Stonjek
MPI für Physik

Amazon Web Services (AWS)

- Web Service access to
 - compute resources: Elastic Compute Cloud (EC2)
 - Simple Storage System (S3)
 - SimpleDB, Elastic Block Store (EBS), etc ...
- Web Service
 - ask for a service via http request
 - response http message is ack or error



[https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009\&InstanceCount=1\&InstanceType=c1.middle
...&...auth parameters...](https://ec2.amazonaws.com/?Action=RunInstances&ImageId=ami-60a54009\&InstanceCount=1\&InstanceType=c1.middle...&...auth parameters...)

AWS



- Pay as you go
 - EC2 CPU cost 10\$ct/h (1-1.2 GHz 2007 PC CPU)
 - S3 Storage 15\$ct/(GB*month) (first 50 TB)
 - 10\$ct/GB I, 17\$ct/GB O (first 10 TB/month)
- Security is built-in
 - private/public ssh key encryption of images and data
- Just give credit card details and start

AWS

Xen virtualisation, replace kernel (think chroot), provide e.g. 1.7 GB RAM,
ca. 1.7 GHz 2006 PC CPU (~1.8 kSI2k), 160 GB disk

Hardware (AWS business secret)

Linux virtualization host (Xen dom0)

Guest 1
SLC4
athena

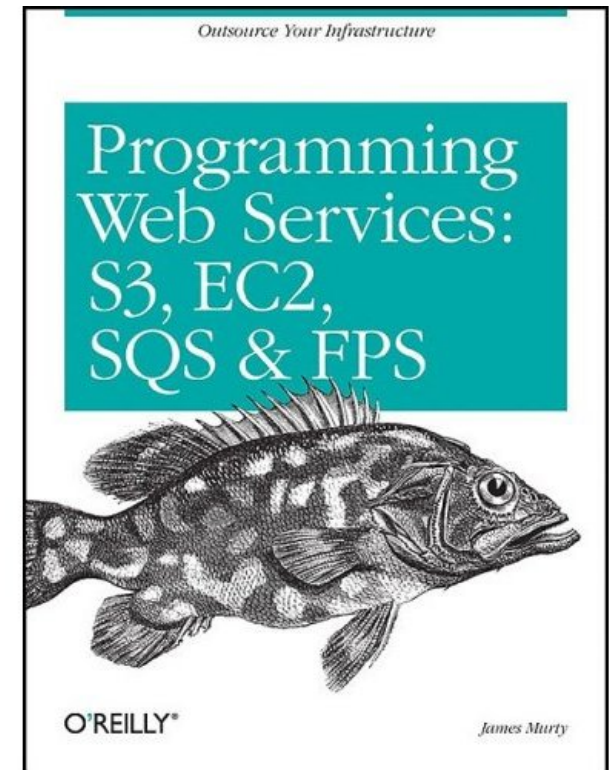
Guest 2
etch

Guest 4
FC9

Guest 4
SLES10

Using AWS

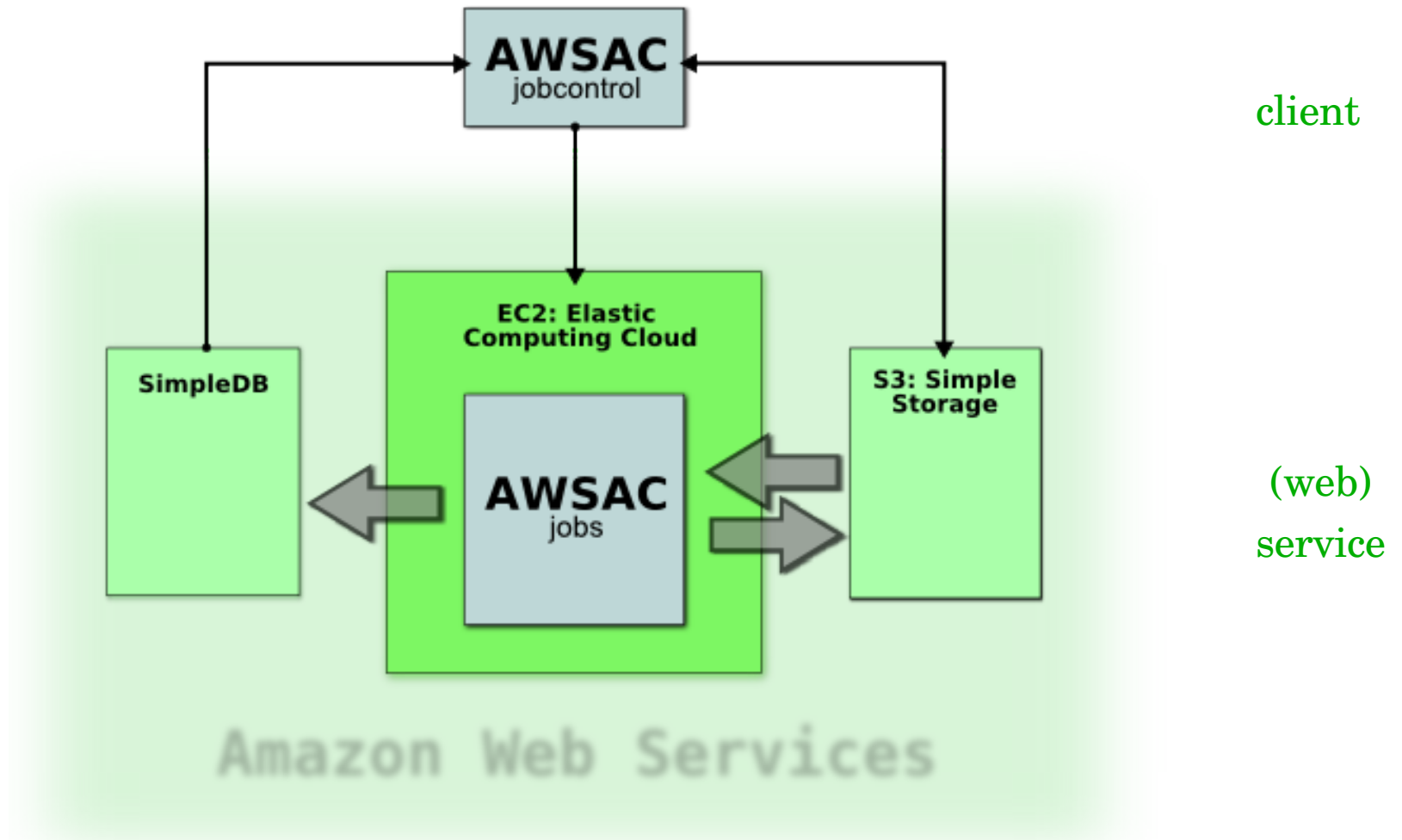
- Choose “Amazon Machine Image” (AMI)
 - official AWS | | contributed | | your own
 - stored encrypted on S3
- Start your AMI instanc(es)
 - choose instance type and number
 - log on, attach EBS disks
 - run your application
 - export results to S3
 - shutdown -h now your instance



AWSAC

[gehrcke.de/awsac]

AWSAC = Amazon Web Services for ATLAS Computing



AWS API available in Java, python, perl, chose python (boto)

AWSACtools

- Client
 - awsac-session (-start | -check | -checkresults)
 - start, monitor (via SimpleDB), stop sessions
 - retrieve results from S3
- SLC4 AMI
 - awsac-autorun
 - receive local configurations and start processing
 - awsac-processjobs
 - manage jobs (athena), EBS, SimpleDB
 - export results to S3

SLC4 AMI Preparation

- athena -> platform
 - head against wall, head looses
- platform -> athena
 - the AWS way, but SLC4 is aging
 - new versions of ruby and tar, disable hardware probing
 - not trivial, but documented, scripting possible
 - SLC4 AMI does not contain releases
 - universal image for all SLC4 compatible releases
 - size restrictions (10 GB)

Releases on AWS

- Install

- start SLC4 AMI instance
- logon, mount EBS under /atlas
- pacman e.g. 14.2.20 under /atlas/14.2.20 (O(h))
- archive contents of /atlas/14.2.20 on S3 (O(min))

- Use

- start SLC4 AMI instance
- mount EBS, get /atlas/14.2.20 from S3 (O(min))
- configure cmt, start job

Tests

- KitVal 14.2.20 successful on EC2
 - athena depends on OS tools, not kernel
 - would be nice if KitVal ran outside of release \Rightarrow smaller archives
- Single π simulation chain
 - “job transform”, complete chain RDO \rightarrow AOD
 - small output volume at AOD level
 - ran successfully (few small jobs)!

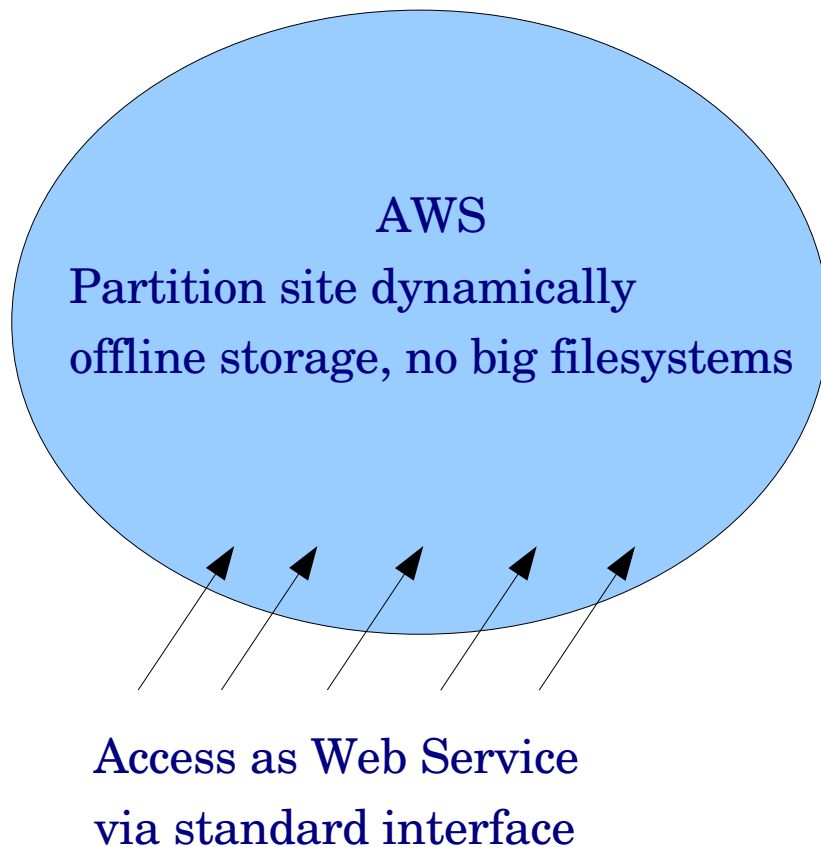
AWSAC Status

- AWSAC code
 - working(!) well written prototype
- AWSAC documentation
 - in progress (gehrcke.de/awsac)
- AWSAC support?
 - Jan-Philip still interested and working

Cloud vs Grid Computing

Cloud computing is
true economy of scale

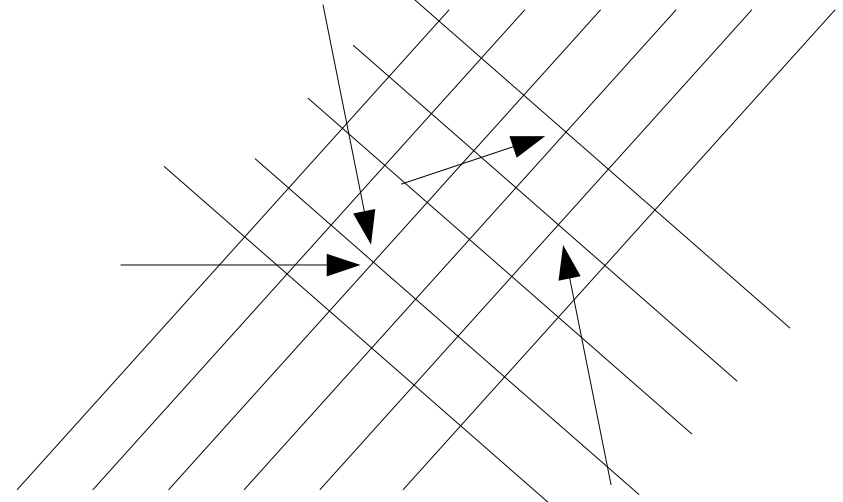
Few huge sites (redundancy, disaster recovery)



Grid computing is technical
solution to political problem

Many sites, tiny to large

Partition sites statically
(pseudo-) online storage



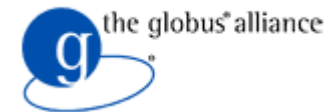
Access via custom (glite) protocol

“demand” platform on sites

Grid fragmented into many incompatible
community specific grids

Should we all do cloud computing?

- Naïve answer: no
 - cost of AWS prohibitive vs. efficient cluster
- Real answer: yes
 - (unexpected) peak loads, unconventional projects
 - allows us to plan for base load < peak load
- Run our own clouds? Yes!
 - Nimbus toolkit provides EC2 API
 - EC2 without the AWS price



Related Work

- Virtual Workspaces (Foster, Keahey, ...)
 - Part of globus toolkit 4 (workspace.globus.org)
- CernVM
 - SLC4 VM + GrowFS/Fuse to import releases
 - limited usefulness on multicore WNs
 - multiple caching
 - single reads slow (no cache yet)
 - CernVM for EC2/Nimbus w/o new filesystem?
 - nfs, afs, lustre, S3, ...

Related Work

- Xen and the Art of Cluster Scheduling
 - SGE frontend (Uni Marburg CS dep.)
 - dynamically create VMs for scheduled jobs
 - manipulate batch queues to reflect “new” cluster
 - submit job to specific queue
- STAR (sc06)
 - virtualized OSG site on physical cluster

Cost Considerations

- AWS
 - ECU ~ 1.8 kSI2k, 10\$ct/h $\Rightarrow 4$ €ct/(kSI2k \cdot h)
- IBM BladeCenter HS21 cluster (RZG)
 - 14x8x2.375 kSI2k, 34 k€, 5 kW, 16 €ct kWh, 3a $\Rightarrow 0.9$ €ct/(kSI2k \cdot h)
- Ratios (cluster efficiency)
 - 4.4 (1) 3.5 (0.8) 2.2 (0.5) 1.1 (0.25)

Summary

- athena on AWS EC2
 - slc4, rel 14.2.20
- AWSAC
 - working prototype EC2 client for ATLAS
- AWS expensive
 - run our own (Nimbus) clouds?
- Cloud computing
 - future of Grid computing?