SGE & Amazon EC2

Chris Dagdigian
2008 OSGC Conference

Putting my $$ where my mouth is

- I'm about to pay $.30 for the privilege of showing you this demo
  - … by firing up a 3-node cluster within EC2
  - … Using cheap 32bit server images that cost $.10 per instance/hour

chris@bioteam.net

Live demo

- Lets start the SGE deployment script
- While it runs I'll keep talking …

chris@bioteam.net

First Principals - 1

- Impossible to cover this topic in any detail with a 30min timeslot
- If *anything* in this talk interests you …
  - Invest 1 day of your time in going through the Amazon EC2 tutorials
  - http://aws.amazon.com

chris@bioteam.net

First Principals - 2

- In ~1 day of EC2 experimentation …
- You will have had enough exposure to
  1. Understand the strengths & weaknesses
  2. Have a decent sense of how your particular needs may or may not be suitable
  3. Be in a position to decide if further investment in time/effort/money is worthwhile
- It will probably cost you about $2.00 USD

chris@bioteam.net

Amazon Web Services

http://aws.amazon.com

Amazon EC2

- Basically a set of APIs you can invoke to manipulate remote VM instances
- Easy to launch existing images
- Easy to build your own custom server images
- Xen instances on-demand
  - Starting at .10/hour for 32bit system
  - 64bit systems start at $.40/hour
  - Fire up as many as you need, whenever you need them
    - Many interfaces/control points
      - Mozilla plugins, CLI, Java, Perl, etc.

chris@bioteam.net

Amazon EC2

- Why it works
  - Smart pricing
    - Server instance pricing is reasonable
    - Traffic to/from S3 storage cloud is free
    - Experimenting is dirt cheap
      - 1 week of messing around == invoice for $9 USD
      - Weeklong SGE training on big machines == $79 USD
  - Easy to use

chris@bioteam.net

Amazon EC2

- Why it works, continued
  - Clever people can make money
    - Amazon allows reselling AMI instance images
      - I can build a specialized workflow engine and charge a small fee on top of the Amazon costs
      - All financial transactions handled by Amazon
        - I don't need to see/know/store your credit card info
  - Limitations are pretty obvious
    - Pretty easy to know what workflows are/are-not EC2 friendly

chris@bioteam.net

Amazon EC2 “Aha! Moment”

- Consider: 100 CPU hour research problem
  - EC2: 10 large servers @ .40/hr for 10 hours
    - Work done in 10 hours at cost of $40 USD
  - EC2: 100 large servers @ .40/hr for 1 hour
    - Work done in 1 hour at a cost of $40 USD
- Can you do THAT in your datacenter today?

chris@bioteam.net

Amazon S3

- Storage service
  - Add and remove stuff into “buckets”
  - Popular with web 2.0 outfits
  - Required component of EC2 usage
    - All EC2 AMI (server images) are stored in S3
  - Cheap to move data in/out
  - Reasonable monthly fee for persistent storage
  - Free to move data within Amazon services
  - Lots of interfaces
    - Mozilla plugins, “s3sync.pl”, FUSE mounts …

chris@bioteam.net

S3 Quick Tour

- *i can haz infinite buckets*
- *let me show you them*

- (Ok, seriously now)
  - Quick demo of Firefox S3 interface

chris@bioteam.net

Amazon SQS

- Simple Queuing Services
  - Message passing between AMI instances
  - Cheap, flexible, reliable
  - Can add new message at any time
    - 8KB size; any format
  - Messages are locked while being processed
  - If read fails, lock is removed
    - Message free to be re-read

chris@bioteam.net

Why I drank the EC2 Kool-Aid

Remember … I'm the "anti Grid" guy

EC2 True Story

- BioTeam is an independent consulting shop
  - Business model
    - 100% objective, vendor & technology agnostic
    - Hire smart people with a known track record
    - Set them loose to solve customer problems
    - Lots of independence
      - Author of the SOW/proposal is the same person who does the actual work for the client

chris@bioteam.net

EC2 True Story, continued

- Since start of 2008
  - *Every single BioTeam consultant has independently deployed one or more EC2 solutions*
    - No corporate mandate
    - Days, not weeks of development time
    - It just made sense
    - Satisfied many diverse use cases and deliverables
- That is a big deal to me

chris@bioteam.net

BioTeam & EC2 (since Jan 08)

- Deploy individual apps in custom AMIs
  - Cross-platform python-based client GUIs
  - Mpiblast, mrbayes, etc.
- Deployed iNquiry product to EC2
  - Scales to arbitrary size for production use
  - Easy to fire off dedicated demo/eval systems
- iNquiry data service moving to EC2/S3
  - Scales bigger, saves thousands of dollars in colocation fees

chris@bioteam.net

BioTeam & EC2 (since Jan 08)

- Grid Engine Training Clusters
  - Virtual classroom & training lab on demand
  - When I teach SGE I want all attendees to be root on their own dedicated cluster
    - *Instead of $20,000 in startup costs I pay ~$79 each time I run a class*
- Full blown Grid Engine clusters
  - Destined for production use
  - Spun up when needed
    - Per user, per-developer, per-workgroup

chris@bioteam.net

EC2 Limitations

- Personally not happy with 64 bit pricing
  - .40/hr is a big jump from the .10/hr 32 bit pricing
  - Would like a “small” 64 bit AMI instance type
- No promises on latency & location
  - Location: select your preferred *continent*
  - AMI instances can be on different subnets
    - OpenMPI had issues with this …
  - Data movement of obvious concern
- Good news
  - Amazon adds features rapidly
    - Within last 1.5 months:
      - Elastic IP addresses
      - Availability zones (request US or European hosting)
      - Cheaper storage transfer rates
      - Support & service contracts

chris@bioteam.net

Sanity Check

SGE use within EC2 may be a dumb idea

SGE/EC2 Sanity Check

- SGE is fantastic for managing *shared* resources
- EC2 can provide *dedicated* resources
  - Per user, per workflow, per application
- Why use SGE at all?
  - Spin up dedicated systems as needed …

chris@bioteam.net

SGE/EC2 Sanity Check

- AWS workflow suggestion:
  - Dedicated management system
    - 2 SQS queues (“incoming”, “outgoing”)
    - Data movement via S3 buckets
    - Workflow indexed via Amazon Simple DB service
  - Fire up worker systems as needed
    - Based on size of incoming message queue
  - Each worker system
    - Grab task off of incoming queue
    - Pull data from S3; work; push results to S3
    - Send task-complete message to outgoing queue
    - Rinse, repeat

chris@bioteam.net

SGE/EC2 Sanity Check

- This is probably obvious:
  - Perform your own due-diligence
  - Deploy what makes sense for you

chris@bioteam.net

EC2 Deployment Suggestions

Key Recommendation

- Changing and re-bundling an AMI takes time
  - I learned this the hard way
    - 15 minutes to bundle the AMI; 1 minute to upload/register
    - Painful when making incremental changes
- Follow the rightscale.com model instead
  - Lightweight AMI images
  - Pull data and sync files from S3 at boot time
  - This is a much faster method of keeping your AMIs synchronized and up to date
- www.rightscale.com plug
  - Nice folks - smart people who don't hoard knowledge
  - They publish their AMI build scripts - fantastic learning resource

chris@bioteam.net

SGE deployment walkthrough

This is one way to do it …

SGE autodeploy in EC2

- Single AMI Centos5 image
  - Must be able to learn all hostnames associated with our reservation
    - Perl module Net::Amazon::EC2 makes this trivial
  - Then decide who is master and who is client
    - Therefore: at least one semi-smart script is baked into the AMI

chris@bioteam.net

SGE autodeploy in EC2

- If we are a “Master”
  - Configure NFS server & /etc/exports
  - Configure ‘dsh’ for remote execution
    - So we can tell nodes to NFS mount the master
    - So we can run SGE client install script remotely

chris@bioteam.net

SGE autodeploy in EC2

- If we are a “node”
  - Query our reservation; learn hostnames
    - Create $SGE_ROOT/$CELL/act_qmaster file
    - Create /etc/fstab entry for NFS mount of master
    - Wait

chris@bioteam.net

SGE autodeploy in EC2

- Biggest issue
  - No control over order in which nodes are booted and when services (SSH) are available
  - This means we can't 100% auto-deploy on AMI server launch
    - Too many sensitive dependencies
- Workaround
  - Create host-specific files (/etc/fstab, /etc/exports, act_qmaster) on boot; then wait
  - Remainder of deployment happens the 1st time root user logs into the "master" node
    - NFS server and client started
    - SGE auto-installation triggered

chris@bioteam.net

Back to live demo

Lets hope this works!

End;

- Thanks!
- Plug
  - http://gridengine.info
  - June ‘08 Grid Engine Workshops
- Questions?
- Comments/feedback:
  - “chris@bioteam.net”