Computing Resources

- First Compute Nodes – 1996
  - goal: being able to run Autocad and Radiance

- Standalone Compute Nodes – 1997
  - goal: offload rendering to standalone hardware

- HPC Clustering
  - goal: batch multiple jobs over cluster, run and forget
    - OpenMosix - 2003
    - OpenSSI - 2004
    - Warewulf - 2006
      - http://warewulf.lbl.gov
      - http://metacluster.lbl.gov

- AWS EC2 – 2008
  - goal: cost, power, availability, scalability, don't want to own hardware
Why use AWS?

- **cost**
  - fixed rate hourly use
    - biddable (spot) rate use
  - fixed rate for storage GB

- **flexibility**
  - power – wide range of instance types
  - availability – when you need them
  - scalability – how many do you need

- **Radiance on AWS**
  - cost and flexibility
What are the key components to AWS?

- **EC2 – Elastic Compute Cloud**
  - AMI – Amazon Machine Image
  - Types – hardware specs
    - http://aws.amazon.com/ec2/pricing/#on-demand
    - http://aws.amazon.com/ec2/pricing/#spot
  - Instance – a running instance of an AMI on a selected type

- **EBS – Elastic Block Store**
  - block level storage - $0.10/GB-month

- **Security**
  - public key encryption
  - security group firewall
How to setup Radiance and AWS

- Setup an AWS account
- Launch a default AMI
- Install Radiance and any other desired tools
- Create a new custom AMI from the running instance
Creating a Custom AMI

- launch an instance based on a default AMI (ex)
  - select default ami
    - Ubuntu 13.04 Server
    - Ubuntu 13.04 Server for HVM Instances
  - select a low cost instance type
    - t1.micro
    - m3.xlarge
  - add an EBS volume for storage of simulation/visualization data
  - login and setup a default user
  - configure ssh
  - setup the EBS volume for use
Install Radiance

- install prerequisites for compiling and running Radiance
  - csh emacs libX11-dev g++ tk make
  - optional - imagemagick
- setup directories (Rlib format)
- install source
- edit pcomb.c for file open size at 4096
- copy additional cal file to radiance lib dir – ray/src/cal/cal
- setup .profile for radiance
  - PATH and RAYPATH
  - ulimit -n 4096
- create an AMI from the running instance
  - creates snapshot that captures all the changes customizations to the default instance
How to run Radiance jobs on AWS

• basic approach
  • ssh to connect
  • scp to transfer data
  • run jobs with nohup
  • terminate vs stop

• fancy approach
  • automated – cli
    – instance startup
    – data transfer in
    – job run
    – data transfer out
    – instance terminate
Closing

- things to watch for
  - leaving volumes hanging around

- benchmark
  - [http://markjstock.org/pages/rad_bench.html#smp_results](http://markjstock.org/pages/rad_bench.html#smp_results)
  - seem to be hitting limit of processing spawning
  - would be interesting to test rtrace with this
  - what would be a suitable benchmark for rcontrib

- further work
  - using a ramdisk on instance with large memory
  - using Dropbox, Ubuntu One for xfer in/out automation
  - clustering on AWS
Selected Works
Selected Works
Selected Works
Selected Works
Selected Works
Selected Works

- IBL sources
- LED sources at selected spandrel conditions
- Interior down lights at corner condition
- Office lights (randomized) at “punched” condition
- Lighting at atrium/lobby condition
- Randomized office lighting at typical conditions
- “background” HDRI
- Ambient term
- Tone mapping for human visual sensitivity
Selected Works
Selected Works
Thanks!