



eTICS2
The Grid Quality Process



INFSOM-RI-1234567

ETICS and EC2

EGEE 08, Istanbul
September 2008

Valerio Venturi
INFN ETICS

Introduction

For those who don't know what ETICS or wants to know more, join us later at the Integrating ETICS in the European Grid Infrastructure session (4PM Malazgirt Hall)



Problem statement

Our problem

- On demand provisioning of nodes to be used for running build and test jobs
- The node must have the required architecture, operating system, and compiler

The generic problem

- On demand provisioning of nodes that meet given requirements to be used to perform a task
 - Minimize infrastructure costs and maintenance



Management of on demand resource

Once started, a bunch of EC2 instances are not that different from a bunch of nodes in a farm

- It's pretty easy to make a Condor pool out of them
 - <http://www-rcf.usc.edu/~juve/condor-ec2/>

But so where the deal would be

- You still pay the costs, as an EC2 core costs 876 euro per year
- You still need the maintenance , as if the instance life cycle is not short, you have to maintain it more or less as you would do for a physical node

Fundamental is the ability to fire up and tear down nodes according to the needs, to scale the pool size according to demand.



Management of on demand resources

Existing solutions

Currently ETICS uses Condor for managing jobs, so it's natural to look around there first

Red Hat has an EC2 integrated solution based on Condor, developed in collaboration with the Condor team at UoW

- Just google condor ec2, a talk at the last Red Hat summit will show up in the first results
- User creates a job AMI and instruments Condor to run it
 - An entry `amazon_ami_id = ami-a1b2c3d4` in the job file

That could be an easy start, but quite not optimal

- Need more flexibility in describing the job, creating a virtual image is more complicated than writing a JDL
- Doesn't optimize resources usage



Management of on demand resources

The cloud architecture approach

Design software application to use internet accessible on-demand services, so that the underlying computing infrastructure is used only when it is needed (for example to process a user request), draw the necessary resources on-demand (like compute servers or storage), perform a specific job, then relinquish the unneeded resources and often dispose themselves after the job is done.

Cloud Architectures white paper

The paper describe an approach at building software applications using Amazon provided services

- SQS for reliably queuing requests
- EC2 to provide the computing resources
- S3, SimpleDB and ESB to provide storage resources



Management of on demand resources

Building on the cloud architecture approach

- Creates a set of AMI, representing the supported platform
- Request for builds or tests are queued to SQS as soon as they arrives
- A controller fire up EC2 instances according to the number of requests in a queue
 - Scaling algorithm according to SLA with customers
- Instances dequeues requests from SQS and execute them
 - Software that do that could either be previously bundled to the image or downloaded later
- The controller tear down EC2 instances as the number of job in the queue decreases

