Implementation of User Level SLA for gLite Using Tycoon

Tycoon and gLite Integration

Amir H. Payberah
payberah@kth.se
Kungliga Tekniska Högskolan
Outline

- Introduction
- System Description
- Installation
Purpose of the BalticGrid JRA1 task:

- User-level service level agreement.
- The possibility for users to acquire more (or less) resources in a well defined manner.
Motivation

- To avoid disturbing the users of the BalticGrid unnecessarily.

- Providing a heterogeneous gLite system.
  - Some nodes can run the unmodified gLite software, and other nodes can run on Tycoon.
Motivation

- **Tycoon** as a market-based utility computing infrastructure.
- Creates and manages a SLA-capable environment, hosting *unmodified gLite*. 
Market-Based Resource Allocation

- Makes **efficient** use of resources.

- Allocates more resources to more important tasks.
  - Tasks are funded according to importance.

- Service level of tasks is determined by its **level of fundings**.
Outline

- Introduction
- System Description
- Installation
Tycoon
Tycoon

- **Market-based** system

- Managing resources in distributed clusters.
Service Model Abstraction

- Provides users to specify resource demand.
- The resources are allocated to the users who value them most.
- Provides a mechanism to encourage users to truthfully reveal those values.
  - Users use a limited budget of credit to bid for resources.
The user submits this bid \((h, r, b, t)\) to the auctioneer on host \(h\).

The auctioneer: \(b/t\) for each bid \(i\) and resource \(r\).

Auctioneer allocates its resources in proportion to the bids.
Architecture

- Agent
- Auctioneer
- Service Location Service (SLS)
- Bank
Agent

Client programs help users set up and manage accounts and also find and manage resources.
Auctioneer

- Management of local resources
- Collection of bids from users
- Allocation of resources to users
- Advertising of the availability of local resources
Auctioneer

- Users use their **agent** to contact auctioneers to query the availability and current prices of resources, bid on resources, etc.

- The auctioneer uses **Xen** for **virtualization**.
Service Location Service

- Keeps track of which auctioneers are up and what their status is.

- An Internet-wide SLS:
  - tycoon-sls.hpl.hp.com
Bank

- Keeps track of the amount of currency that different users have in their bank accounts.

- An Internet-wide bank:
  - tycoon-bank.hpl.hp.com
How does it work?

1. register
2a. query
2b. list of auctioneers
3a. transfer
3b. receipt
3c. fund (w/ receipt)
4. set_interval
5. Allocate resources based on funds transferred
Prototype User Interface

- Create an account on a host
  - tycoon create_account host0 10

- Run
  - tycoon_ssh amir@host0 my_program

- Transfer more credits into account
  - tycoon fund host0 cpu 10 1000

- Change bidding interval
  - tycoon set_interval host0 cpu 2000

- Determine current balance, resources allocated, etc.
  - tycoon get_status host0
Setting bids

Two methods of changing bid:

- **fund**
  - Transfers money from the user's bank account to the auctioneer's bank account.
  - High latency

- **set_interval**
  - Sets the bidding interval at the auctioneer.
  - Low latency
gLite
The components of the gLite system can be divided into two categories:

- Site level components
- Grid level components
Site Level

- **Job management system**
  - Computing Element (CE)
    - Acts as interface to the Grid level
    - Delegating jobs to the WNs
  - Worker Node (WN)
    - Performs the actual work

- **Data management system**
  - Storing data
  - Map globally unique ID to local file names

- **Information and monitoring service**
  - Publishing information of the state of CEs or jobs
Grid Level

- Grid security

- Information services
  - Binding information publishers at the site level and information consumers.

- Data management systems
  - Keeps track of where particular files are stored.

- Workload Management System (WMS)
  - Matching jobs with available CEs.
VOMS

- Repository for information of user authorization.

- Components such as CEs and WMSs, use a VOMS sever's SOAP interface to receive such information.
Tycoon-gLite Integration
Tycoon-gLite Integration
Tycoon-gLite Integration

- The gLite WMS is used for submitting jobs.
- Tycoon SLSs are used for discovering resources for creation of virtual clusters.
  - One SLS for each LAN
- VO managers can request the Auctioneers to create several VMs that together constitute a virtual gLite compute cluster.
  - One CE in each virtual cluster
  - An arbitrary number of WNs
- Only the member of the VO owning a virtual cluster are able to submit jobs to it.
Advantages

- “Easy” to implement
- Unmodified gLite
- Dynamic and secure system useful to trade resources
- Fully transparent for Grid users
Limitation

- Virtual clusters are created on VO level

- VO agent has no information on the priority of individual jobs

- All the nodes of a virtual cluster have to be in the same LAN (for easier communication between CE and WNs and between different WNs)
Outline

- Introduction
- System Description
- Installation
System Platform
Platform

- Fedora Core filesystem
- Xen virtualization software
- Tycoon Auctioneer
- ScientificLinux filesystem images for use by the Tycoon-managed CEs and WNs
Installing Tycoon
Tycoon Client

- **Install packages**
  - `yum -y -c http://tycoon.hpl.hp.com/~tycoon/dl/yum/tycoon.repo install tycoon_client`

- **Leave setup to use the standard SLS and Bank (recommended)**
  - `tycoon-sls.hpl.hp.com`
  - `tycoon-bank.hpl.hp.com`

- **Or install and configure our own SLS and Bank.**

- **Configure firewall**
  - SLS: out TCP port (25955)
  - Bank: out TCP port (8899)
  - Auctioneer: out TCP port (24571)
Create Bank Account

- Generate a ssh public key and configure to log into Tycoon machines
  - `$ ssh-keygen -t dsa`
  - `$ cat .ssh/id_dsa.pub >> .ssh/authorized_keys`
  - `$ chmod 600 .ssh/authorized_keys`

- Setup a Tycoon configuration
  - `$ tycoon user setup amir@kth.se amir ~/.ssh/id_dsa.pub`

- Use online form and send your public key
  - [http://tycoon.hpl.hp.com/wiki/TycoonAccountForm](http://tycoon.hpl.hp.com/wiki/TycoonAccountForm)
Create Bank Account

- Verify Bank
  - $ tycoon bank get_balance
  - Account balance: 100

- Verify SLS
  - $ tycoon sls query_cost_efficiency CPU
  - IP Address | GHz\*Hour/Price | GHz  | Price/Hour
  - 128.142.134.113 | 4.301e+13  | 3.600 | 8.371e-14
  - 128.142.134.111 | 4.271e+13  | 3.600 | 8.430e-14
Tycoon Auctioneer

- Install packages
  - `yum -y -c http://tycoon.hpl.hp.com/~tycoon/dl/yum/tycoon.repo install tycoon_aucd_xen3`

- Finish the installation
  - `iptables -A INPUT -s ! 127.0.0.1 -p tcp --dport 8001:8002 -j REJECT`
  - `iptables -A INPUT -s ! 127.0.0.1 -p tcp --dport 9601:9699 -j REJECT`
  - `service iptables save`
  - `echo 1 > /proc/sys/net/ipv4/ip_forward`

- Reboot with new Xen Kernel

- Configure SLS and Bank (same as client)
Configure Tycoon Auctioneer

- Copy the owner's bank account key pair to /etc/tycoon
  - amir@kth.se_bank_private_key
  - amir@kth.se_bank_public_key

- Copy the owner's public key to /etc/tycoon/admin_public_key
  - amir@kth.se_bank_public_key

- Change (or add) the UserName option in /etc/tycoon/tycoon_aucd.conf
  - UserName = "amir@kth.se"

- Configure firewall(s) (Linux and/or external) to open ports auctioneer
  - SLS: out TCP port (25955)
  - Bank: out TCP port (8899)
  - Auctioneer: out TCP port (24571)
Create Account on Host

- Create host on Auctioneer (It boots VM as well)
  - `tycoon host create_account 192.168.0.1 10`

- Verify the host created VM
  - `tycoon_ssh amir@192.168.0.1`
  - `tycoon_scp a.out amir@192.168.0.1`
Auctioneer uses Xen for virtualization.

It uses configuration file in /var/lib/tycoon/aucd/Xen3/accounts

```plaintext
kernel = "/home/amir/kernel/vmlinuz"
disk = ['file:/home/amir/slc3.img,sda1,w']
name = "amir"
root = "/dev/sda1 ro"
memory = 856
vif = [ "mac=aa:00:00:77:66:d4, ip=10.106.212.209/30", "mac=aa:00:00:2b:ca:15" ]
ip = "10.106.212.210"
netmask = "255.255.255.252"
gateway = "10.106.212.209"
hostname = "amir-boogieman.pdc.kth.se"
```
Creating gLite Filesystem Image
gLite Filesystem Image

- Installing Scientific (SL) Linux on QEMU or one partition
  - gLite uses **yaim** as a configuration tool.
- Install CE and WN on SL
  - The CE and WN on image are preinstalled, but not configured.
  - /opt/glite/yaim/scripts/install_node site-info.def glite-CE
  - /opt/glite/yaim/scripts/install_node site-info.def glite-WN
- Copying required keys and certificates to image.
Summary

- Providing a heterogeneous gLite system.

- Tycoon as distributed market-based resource allocation system.

- Each site has:
  - Its SLS
  - Set of nodes with Auctioneer on them
  - gLite prepared filesystem image (CE and WN)