

Tycoon User's Manual

Kevin Lai (klai@hp.com)

2/14/2004

Abstract

This manual describes the usage of the Tycoon market-based distributed resource allocation system. Tycoon allocates computer resources in a distributed cluster like PlanetLab, the Grid, or a Utility Data Center (UDC) to users. To do this, Tycoon uses a *market-based* mechanism where users have a limited amount of currency that they use to bid on resources. Users receive resources in proportion to their bids.

Contents

1	Introduction	1
1.1	Quick Start / Tutorial	2
2	Configuration	2
3	Users: Basic Agent	3
3.1	General Commands	4
3.1.1	get_host_list	4
3.2	Bank Commands	4
3.2.1	bank get_balance	4
3.2.2	bank create_account	4
3.2.3	bank transfer	4
3.2.4	bank get_history	4
3.2.5	bank delete_account	5
3.3	Host Commands	5
3.3.1	host create_account	5
3.3.2	host fund	5
3.3.3	host set_interval	5
3.3.4	host refund	5
3.3.5	host get_status	5
3.3.6	host delete_account	5
3.3.7	Error Messages	5
4	Frequently Asked Questions	6

1 Introduction

This manual describes the usage of the Tycoon market-based distributed resource allocation system. The User's Manual is for people who just want to run jobs on a Tycoon system. The Administrator's Manual is for people who want to contribute hosts to a Tycoon system or want to administrate an entire Tycoon system.

1.1 Quick Start / Tutorial

The Tycoon packages must have been installed on your Linux machine before you can use this Quick Start guide (see Administering a Client in the Administrator's Guide). The following steps walk through a new user setting up her bank account and one host account and starting to run on that host.

1. You must have an ssh public key to use Tycoon. If you already have one, skip to step 2. Otherwise, you must create one. For example, Alice Smith running on her workstation "host" would type the following and then hit return at all the prompts:

```
asmith@host> ssh-keygen -t dsa
```
2. Setup your tycoon configuration with your email address, the account name that you would like to use on remote machines, and your ssh public key. For example, Alice would type:

```
asmith@host> tycoon user setup alice.smith@hp.com asmith ~/.ssh/id_dsa.pub
```
3. Send the email address you specified in step 2 and your bank account public key to your tycoon system administrator. Your bank account public key is in `~/.tycoon/<user_name>/bank_account_public_key`.
4. Get the list of machines with their speed and current total bids. Alice would type:

```
asmith@host> tycoon get_host_list
```
5. Create accounts on the machines you want. Alice would type the following command to create an account on host 1.2.3.4 and transfer 10 credits:

```
asmith@host> tycoon host create_account 1.2.3.4 10
```
6. Copy programs and log into your machine using your account name. Alice would type these commands to copy the directory `my_program` to 1.2.3.4 and log into that machine:

```
asmith@host> scp -r my_program 1.2.3.4:  
asmith@host> ssh asmith@1.2.3.4
```
7. You may find that you need to install software on your machine. You can use the "yum" program. Alice would log into 1.2.3.4 and would use these commands to install gcc:

```
asmith@1.2.3.4> su  
root@1.2.3.4> yum install gcc
```
8. You should periodically check your current credit balance and spending interval at your machines. Your balance divided by your spending interval is your spending rate, which determines how much of that machine you get. Alice would type these commands to get her balance and spending interval at 1.2.3.4:

```
asmith@host> tycoon host get_status 1.2.3.4 balance interval
```
9. At some point, you may run out of money on your machine or find that it is too slow. You can add more money and/or increase the spending rate. Alice would do type this to transfer 10 more credits and change the spending interval to 10000 seconds:

```
asmith@host> tycoon host fund 1.2.3.4 10 10000
```

2 Configuration

Tycoon programs are configured using a series of files containing Python code that sets various variables. Tycoon programs use specific variables to configure themselves. For example, the variable "BankHostName" contains the host name of the bank.

The configuration files can contain any Python code, including import statements. These files are executed to obtain the configuration variables. Files that are executed later can override the values set by files that are executed earlier. In some cases, a configuration variable that is set earlier requires a configuration value

set by a later file. The configuration system runs two passes to provide this ability. The earlier variable must be set to a string with a string formatting mapping key of the same name as the later variable. For example, this allows `NonceFileName` to use the value of `UserName`, even though `UserName` is set later:

```
NonceFileName = "~/tycoon/%(UserName)s/nonce"
```

In addition, the dictionary variable `SecondPassOptions` must have a key with the same name as the earlier variable set to `True`:

```
SecondPassOptions['NonceFileName'] = True
```

The specific set of configuration files used varies from program to program, but in general, there is a non-program-specific file name (“tycoon.conf”) and a program-specific file name (e.g., “tycoon_dist_auc.conf”) and there is a system-wide file system path (“/etc”) and a user-specific path (“~/tycoon”). The files are executed in the following order:

1. non-program-specific file name at the system-wide location
2. program-specific file name at the system-wide location
3. non-program-specific file name at the user-specific location
4. program-specific file name at the user-specific location

3 Users: Basic Agent

Users interact with Tycoon through a software *agent*. A simple agent simply allows a user to manually control how his currency and jobs are distributed across hosts. More sophisticated agents automate some of these tasks. The only agent currently available is a simple agent.

The basic agent allows users to manually set the amounts spent at different hosts. Users can then manually log into those hosts using SSH. Also, users must manually fund their host accounts again as the funds are depleted.

The basic agent is installed as “/usr/bin/tycoon”. Its system-wide configuration options are in “/etc/tycoon.conf”. User-specific configuration options are in “~/tycoon/tycoon.conf”. Described below are the user-specific options that users must set:

SLSHostName: This is the host name of the Service Location Service.

BankHostName: This is the host name of the bank.

UserName: This is the user’s bank account name.

RemoteAccountName: This is the user’s default host account name. This is the account name that the user uses to log into Tycoon hosts. This is set when the user creates accounts on Tycoon hosts (see below).

The default behavior is to apply the system-wide configuration options and then the user-specific configuration options. To change this behavior, use the `--config_search_path` option with a list containing the new search path. For example, to apply the `tycoon.conf` file in the current directory and then the user-specific file:

```
tycoon --config_path "('./', '~/tycoon')" ...
```

To run the basic agent:

```
tycoon [options] get_host_list | bank | host
```

This command will use the user-specific options described above.

The following sections describe the subcommands in more detail.

3.1 General Commands

3.1.1 get_host_list

```
tycoon get_host_list [fields]
```

This prints out a list of all the hosts in the system. If the fields are not specified, then a commonly useful set is printed. If fields is “all”, then all the fields are printed. You can reduce the amount of data printed by specifying fields. For example to only print out the machines’ Mhz, IP address, and price history, use

```
tycoon get_host_list "cpu_info[cpu MHz]" ip_address price_history
```

3.2 Bank Commands

All of the bank commands implicitly operate on the user’s bank account. The commands also have an administrative form that allows the administrator perform operations on user’s accounts. This form is

```
tycoon -oUserName=bank_admin bank admin <command> <command args> <account_name>
```

For example, to check Bob’s balance, do the following:

```
asmith@host> tycoon -oUserName=bank_admin bank admin get_balance bob.johnson@hp.com
```

The following are the bank commands.

3.2.1 bank get_balance

```
tycoon bank get_balance
```

This prints out the user’s current bank balance.

3.2.2 bank create_account

```
tycoon bank create_account <account_name> <initial_deposit> <account_public_key_file_name>
```

<initial_deposit> This is the initial amount to be deposited in the new account. These funds are transferred from the creating user’s account

<account_public_key> This is the file name of the new account’s public key.

This creates a new bank account. The new account’s public key should have been generated as described in Section 1.1. Although account creation is usually performed by the bank administrator using the administrative form described above, this is not required. In fact, any Tycoon user can create a new Tycoon user account.

The initial deposit is necessary in case the bank has reached a preset maximum number of accounts, n . If the initial deposit of the new account is more than the balance of at least one of the existing accounts, then the account with the smallest balance is evicted and the account is created. Otherwise, the creation of the new account is rejected.

3.2.3 bank transfer

```
tycoon bank transfer <recipient> <amount>
```

<recipient> This is the bank account name of the recipient of the transfer.

<amount> This is the amount to transfer.

This transfers funds from the user’s account to someone else’s account. This can be used for transactions outside the scope of the Tycoon markets.

3.2.4 bank get_history

```
tycoon bank get_history
```

This gets a list of a user’s recent transactions from the bank. This can be used to reconcile disputes about transfers.

3.2.5 bank delete_account

```
tycoon bank delete_account <account_name>
```

This deletes the specified bank account. The funds in the account disappear, so use this command with caution.

3.3 Host Commands

All of the host commands take a `host_names` parameter. This can be a single host name or IP address or a list of them enclosed in double quotes and separated by spaces (e.g. "1.2.3.4 w.x.y.z"), or simply a pair of double quotes, which means to use all of the machines in a cluster.

3.3.1 host create_account

```
tycoon host create_account <host_names> <initial_deposit>
```

<initial_deposit> This is the initial amount to be deposited in the new account.

This creates an account at the specified host. Note that this command can take a long time (~300s) because a file system must be created and populated. An initial deposit is necessary because hosts use the same policy as the bank to handle too many accounts (see above).

3.3.2 host fund

```
tycoon host fund <host_names> <fund_amount> <funding_interval>
```

This funds the user's account at the specified host. The specified funds are transferred to the host's account. The remote auctioneer is notified that all of the user's funds at the remote host are to be spent over the specified interval.

3.3.3 host set_interval

```
tycoon host set_interval <host_names> <interval>
```

This sets the spending interval for the user's account at the specified host. The remote auctioneer is notified that all of the user's funds at the remote host are to be spent over the specified interval. Note that this operation is faster than "host fund", so it is better to use this if you are not adding funds.

3.3.4 host refund

```
tycoon host refund <host_names> <fund_amount>
```

This refunds money from the user's account at the specified host. To refund all the funds, specify "all" as the fund amount.

3.3.5 host get_status

```
tycoon host get_status <host_names>
```

This gets the status of the user's account at the remote machine. This is mainly useful to see what funds are left there.

3.3.6 host delete_account

```
tycoon host delete_account <host_names>
```

This deletes the user's account on the remote machine. This deletes all of the user's files there and refunds the remaining funds.

3.3.7 Error Messages

XXX List error messages here

4 Frequently Asked Questions