

# FALKON: A FAST AND LIGHT-WEIGHT TASK EXECUTION FRAMEWORK

---

## *User Guide*

---

*Web Site:* <http://dev.globus.org/wiki/Incubator/Falcon>

*People:* Ian Foster, Mike Wilde, Yong Zhao, Catalin Dumitrescu, Zhao Zhang

*Institution:* Distributed Systems Laboratory, Computer Science Department, University of Chicago

## **1 Abstract**

Falcon was designed and implemented to enable the rapid and efficient execution of many independent jobs on large compute clusters. Falcon combines three techniques to achieve this goal: (1) multi-level scheduling techniques to enable separate treatments of resource provisioning and the dispatch of user tasks to those resources; (2) a streamlined task dispatcher able to achieve order-of-magnitude higher task dispatch rates than conventional schedulers; and (3) performs data caching and uses a data-aware scheduler to leverage the co-located computational and storage resources to minimize the use of shared storage infrastructure.

## **2 Short User Guide**

We summarize the entire user guide in this section. For a quick overview, please use this section. For a more detailed write-up, please see sections 3+.

### **2.1 Installation Instructions**

To get the entire Falcon source tree, type:

```
svn co https://svn.globus.org/repos/falcon
```

Double check that you have the core pre-requisites:

```
echo $ANT_HOME
echo $JAVA_HOME
```

Both of these environment variables should be set. We have been testing with Java 1.4+ and ANT 1.6+, but earlier and later versions might work as well.

To compile the entire Falkon source tree, type:

```
cd falkon
source falkon.env
falkon-build.sh
```

If it is successful, then you are done with the installation of Falkon. If not, then please see the sections below for more details. Note that “source falkon.env” sets up various environment variables, including the \$FALKON\_HOME which we refer to in the next sections; you must be in the \$FALKON\_HOME directory when you invoke “source falkon.env” as it uses `pwd` to set various environment variables.

A complete SVN checkout can take between 80 seconds and 400 seconds to complete (depending on you system speed and network connectivity), and will occupy about 185 MB. If you want to only get essential Falkon components (most people only need these essential components, which will only take 50~300 seconds to checkout and occupy only 38MB), follow these instructions:

```
svn co https://svn.globus.org/repos/falkon -N
cd falkon
svn co https://svn.globus.org/repos/falkon/bin
source falkon.env
falkon-checkout-minimal.sh
```

If you don't have access to SVN, the source tree archive (equivalent to the minimal checkout from above) can be downloaded from:

...

## 2.2 Configuration Details

The following files are used to configure Falkon.

- ***`\${FALKON\_PATH}/config/Falkon.config`***: configures the Falkon service settings
- ***`\${FALKON\_PATH}/config/Provisioner.config`***: configures the Provisioner settings
- ***`\${FALKON\_PATH}/config/client-security-config.xml`***: configure the command line client security settings; the default settings are set to no security
- ***`\${FALKON\_PATH}/config/global\_security\_descriptor.xml`***: configure the GT4 container security settings; the default settings are set to no security
- ***`\${FALKON\_PATH}/config/security-config-submit.xml`***: configure the Falkon service security settings; the default settings are set to enable GSISecureConversation with authorization and encryption on the task submit function with the rest of the service functionality running without security

- **`${FALKON_PATH}/config/worker-security-config.xml`**: configure the worker security settings; the default settings are set to no security
- **`${FALKON_PATH}/config/security-config.xml`**: configure the Falkon service security settings; the default settings are set to no security
- **`${FALKON_PATH}/config/grid-mapfile`**: configures the authorization for accessing the Falkon service

The default Falkon.config should be adequate for most installs. For more details on modifying the Falkon configuration, please see section 4.

The default Provisioner.config has some default values, but most users will have to modify some of these values to make sure it matches their environment. If you just want to test Falkon by manually starting workers on compute resources, you can skip the Provisioner configuration. Here are the minimal configuration options that are likely to have to be changed from the default values:

- ***MinNumExecutors***: the minimum number of executors to be allocated; there is usually a 1-1 mapping between processors and executors
- ***MaxNumExecutors***: the maximum number of executors to be allocated; there is usually a 1-1 mapping between processors and executors
- ***ExecutorsPerHost***: the number of executors per host; this is usually the number of processors per host
- ***MinResourceAllocationTime\_min***: the minimum allocation time
- ***MaxResourceAllocationTime\_min***: the maximum allocation time
- ***DeAllocationIdleTime\_sec***: the time in milliseconds that executors are allowed to be idle before they are de-allocated
- ***FalkonServiceURI***: the location of the Falkon GT4 service
- ***GRAM4\_Location***: the location of the GRAM4 service
- ***GRAM4\_FactoryType***: the type of LRM behind GRAM4 (i.e. PBS, LSF, CONDOR, FORK)
- ***Project***: project number for accounting purposes; this is optional, but some grid sites will not work unless this is specified

By default, security is disabled in Falkon, so any of the security configuration files do not need to be edited unless security needs to be enabled. For more details on enabling security, please see section 4.

## 2.3 How to Run Falkon

Lets try each component individually, just for testing. Start 3 different shells. To avoid the first 2 lines in each shell "`cd ${FALKON_HOME} ; source falkon.env`", you could place this in the `.bashrc` (or equivalent for other shells) to automatically setup the environment when the shell starts (keep in mind that you will have to replace the `${FALKON_HOME}` variable with the actual installed path of Falkon).

**Shell #1:** starts GT4 container (including Falkon service) on port 50001

```
cd ${FALKON_HOME}
source falkon.env
falkon-service-stdout.sh 50001 ${FALKON_CONFIG}/Falkon.config
```

Shell #2: starts 1 worker on local machine in interactive mode, terminate by simply typing any key and hit enter

```
cd ${FALKON_HOME}
source falkon.env
falkon-worker-stdout.sh localhost 50001
```

Shell #3: starts the command line client that submits 10 sleep 1 tasks

```
cd ${FALKON_HOME}
source falkon.env
falkon-client.sh localhost 50001 workloads/sleep/sleep_1x10
```

If everything went OK, here is the sample output of shell #3:

```
iraicu@vipер:~/falkon> falkon-client.sh localhost 50001 workloads/sleep/sleep_1x10
Starting Falkon Command Line Client v0.8.1...
Starting non-interactive mode....
Reading file: workloads/sleep/sleep_1...

null time 0.0 tasks_success 0 tasks_failed 0 tasks_sent 0 completed 0.0 tasks_tp 0.0 aver_tp 0.0 stdev_tp 0.0 ETA ?
null time 0.0010 tasks_success 0 tasks_failed 0 tasks_sent 0 completed 0.0 tasks_tp 0.0 aver_tp 0.0 stdev_tp 0.0 ETA ?
null time 1.006 tasks_success 0 tasks_failed 0 tasks_sent 10 completed 0.0 tasks_tp 0.0 aver_tp 0.0 stdev_tp 0.0 ETA ?
null time 2.01 tasks_success 1 tasks_failed 0 tasks_sent 10 completed 10.0 tasks_tp 1.0 aver_tp 1.0 stdev_tp 0.0 ETA 18.09
null time 3.014 tasks_success 2 tasks_failed 0 tasks_sent 10 completed 20.0 tasks_tp 1.0 aver_tp 1.0 stdev_tp 0.0 ETA 12.056
null time 4.017 tasks_success 3 tasks_failed 0 tasks_sent 10 completed 30.0 tasks_tp 1.0 aver_tp 1.0 stdev_tp 0.0 ETA 9.375
null time 5.021 tasks_success 4 tasks_failed 0 tasks_sent 10 completed 40.0 tasks_tp 1.0 aver_tp 1.0 stdev_tp 0.0 ETA 7.532
null time 6.024 tasks_success 5 tasks_failed 0 tasks_sent 10 completed 50.0 tasks_tp 1.0 aver_tp 1.0 stdev_tp 0.0 ETA 6.024
null time 7.032 tasks_success 6 tasks_failed 0 tasks_sent 10 completed 60.0 tasks_tp 0.99 aver_tp 1.0 stdev_tp 0.0010 ETA 4.688
null time 8.039 tasks_success 7 tasks_failed 0 tasks_sent 10 completed 70.0 tasks_tp 0.99 aver_tp 1.0 stdev_tp 0.0020 ETA 3.446
null time 9.043 tasks_success 7 tasks_failed 0 tasks_sent 10 completed 70.0 tasks_tp 0.0 aver_tp 0.87 stdev_tp 0.329 ETA 3.876
null time 10.047 tasks_success 8 tasks_failed 0 tasks_sent 10 completed 80.0 tasks_tp 1.0 aver_tp 0.89 stdev_tp 0.313 ETA 2.512
null time 11.051 tasks_success 9 tasks_failed 0 tasks_sent 10 completed 90.0 tasks_tp 1.0 aver_tp 0.9 stdev_tp 0.299 ETA 1.228
null time 12.054 tasks_success 10 tasks_failed 0 tasks_sent 10 completed 100.0 tasks_tp 1.0 aver_tp 0.91 stdev_tp 0.286 ETA 0.0
10 tasks completed in 12.235 sec
Successful tasks: 10
Failed tasks: 0
Notification Errors: 0
Overall Throughput (tasks/sec): 0.82
Overall Throughput Standard Deviation: 0.286
```

For more serious stuff, you'll have to run the Provisioner:

```
cd ${FALKON_HOME}
source falkon.env
falkon-provisioner-stdout.sh ${FALKON_CONFIG}/Provisioner.config 15
```

You can then try the command line client example from above again, to make sure it works. Keep in mind that Executors could take 1~2 minutes to get allocated (on remote resources) even if resources are available.

You will find various logs at `$FALKON_LOGS` separated into directories based on the start of each component.

There is also a monitor that allows you to see the state of Falkon without having to dig through the log files. There are 2 different monitors (text based and graphical based). To start the text mode monitor, type:

```
cd ${FALKON_HOME}
source falkon.env
falkon-top.sh localhost 50001 1000
```

To start the graphical mode monitor, type:

```
cd ${FALKON_HOME}
source falkon.env
falkon-monitor.sh
```

The rest of this document is a work in progress, please check back soon for updates. For questions and problems not addressed in this document, please see the main Wiki Falkon site at <http://dev.globus.org/wiki/Incubator/Falkon> and the mailing lists which can be found on the main Wiki page.