User Guide for Jitter-tolerant Framework

1 Introduction

This brief user guide is for researchers and scientists to use our jitter-tolerant framework for time-stepped applications. The jitter-tolerant framework was proposed in our recent paper "Making Time-stepped Applications Tick in the Cloud " published in ACM Symposium on Cloud Computing 2011.

We have packed our runtime setup, as well as example applications, in a public Amazon Linux AMI which is visible to all EC2 users. In the following sections, methods to access such AMI, and to run user-defined applications will be described.

2 Start Instances

To get our public AMI ID, please execute

```
ec2-describe-images -a | grep Jitter
```

This command returns a single line in which the second string is the AMI ID.
Assume the ID shown is "ami-8e916fe7", to launch new instances, please execute

```
ec2-run-instances ami-8e916fe7 --instance-type m1.large
--availability-zone us-east-1a -k YourKeyPair
```

3 Upload Host File

First, choose any instance as the master, say the instance with external IP address "ec2-67-202-62-111.compute-1.amazonaws.com".
Then, use the following command to generate the host file.

```
ec2-describe-instances | grep running | awk '{print $16}' > hosts
```

Finally, please execute the following command to upload this host file.

```
scp -i YourKeyPair hosts
root@ec2-67-202-62-111.compute-1.amazonaws.com:host
```

4 Run Applications

Now, we can start to run applications using the jitter-tolerant framework.

4.1 Fish Simulation

Usage:

```
fishSim.exec infile [-t ticks] [-lx layout_x] [-ly layout_y]
[-e exectuion] [-d sch_d] [-rm rep_m] [-rk rep_k]
```

List of optional parameters:

parameter	meaning	default value
-t	number of ticks to execute	2000
-d	scheduling depth	5
-lx	processor layout x	int(sqrt(MPI_SIZE-1))
-ly	processor layout y	int(sqrt(MPI_SIZE-1))
-e	execution method (local/sch/rep/combine)	sch
-rm	replication m	2
-rk	replication k	1

To run fish simulation using 49 instances organized as a 7 by 7 grid, please execute

```
mpirun -hostfile ~/host/hosts -n 50 -x LD_LIBRARY_PATH
~/jitter/bin/fishSim.exec ~/jitter/data/fish/fish50K.dat
-t 1000 -e combine -lx 7 -ly 7 -d 10 -rm 5 -rk 3
```

4.2 Jacobi Solver

Usage:

```
jacobi.exec -s size [-t ticks] [-lx layout_x] [-ly layout_y]
[-e execution] [-d sch_d] [-rm rep_m] [-rk rep_k]
```

List of optional parameters:

parameter	meaning	default value
-t	number of ticks to execute	2000
-d	scheduling depth	5
-lx	processor layout x	int(sqrt(MPI_SIZE-1))
-ly	processor layout y	int(sqrt(MPI_SIZE-1))
-e	execution method (local/sch/rep/combine)	sch
-rm	replication m	2
-rk	replication k	1

To run 2D head diffusion solver using 49(7*7) instances with a 1000*1000 block on each instance, please execute

```
mpirun -hostfile ~/host/hosts -n 50 -x LD_LIBRARY_PATH
~/jitter/bin/jacobi.exec -s 1000 -t 2000 -e combine
-lx 7 -ly 7 -d 10 -rm 5 -rk 3
```

4.3 PageRank

Usage:

```
graph.exec infile [-t tickes] [-e execution] [-d sch_d]
```

List of optional parameters:

parameter	meaning	default value
-t	number of ticks to execute	100
-d	scheduling depth	5
-e	execution method (local/sch)	sch

To rum PageRank using 36 instances over a pre-partitioned U.S. Patent Citation Network, please execute

```
mpirun -hostfile ~/host/hosts -n 37 -x LD_LIBRARY_PATH
~/jitter/bin/graph.exec ~/jitter/data/graph/cit-Patents-36
-t 500 -d 11 -e sch
```