# ReGen v1.0 User Manual

Helgi Leifsson

November 27, 2014

## 1 Policy Options

### 1.1 Dispatch

No preemption used. If jobs are started they get to finish or their deadlines run out.

### 1.2 Natjam-R eviction

For MDF and MLF, jobs can be preempted for other jobs to run. They are checkpointed at the RM until resources become available. They are then restarted where they left off. A job can be preempted multiple times. If there are no checkpoints to restart, EDF is used to dispatch new jobs.

### 1.3 Dispatch policies

Multiple policies can be selected by holding down the CTRL key and selecting with the mouse.

**EDF** jobs are dispatched on an earliest deadline first basis.

**FIFO** jobs are dispatched on a first-in-first-out basis.

**MDF** jobs are dispatched on a maximum deadline first basis.

**Priority** jobs have a low or high priority. High priority jobs are dispatched as soon as resources are available. If two or more jobs have the same priority, EDF is used to select between them.

### 1.4 Natjam-R policies

**MDF** jobs are evicted based on a maximum deadline first policy

**MLF** jobs are evicted based on a maximum laxity first policy where $laxity = deadline - jobs\ projected\ completion\ time$.

### 1.5 Job arrival patterns

Lists the available job arrival patterns. Multiple job arrival patterns can be selected by holding down the CTRL key.

**Bursty** Jobs come in bursts with a fixed interval and a separate fixed amount.

**Nondet** The number of jobs arriving every timeunit is nondeterministic.

**Uniform** The number of jobs arriving every timeunit is uniform.

**Wave** Job arrival follows a wave pattern and goes systematically from a fixed lowest point to a fixed maximum point.

**Ascending** The number of job arrivals ascends from a lowest number to a highest number repeatedly.

**Descending** The number of job arrivals descends from a highest point to a lowest point. Number of jobs less than 0 is set as 0.

### 1.6 Job length patterns

Lists the available job length types. Multiple job length patterns can be selected by holding down the CTRL key.

**Exponential** Job length grows exponentially.

**Nondet** Job length is nondeterministic with a minimum and maximum length.

**Uniform** Job length is uniform.

**Wave** Job length follows a wave pattern from a lowest point to a highest point and back down again.

**Ascending** Job length ascends to a highest point and starts back from the lowest point.

**Descending** Job length descends from a highest point to a lowest point (minimum 0) and starts back from the highest point.

## 2 Common parameters

Parameters shared between all policies, job arrival patterns and job length patterns.

### 2.1 Max AppMasters

The maximum amount of AppMasters, or concurrent jobs.

### 2.2 Queue size

The size of the incoming and checkpoint queues.

### 2.3 Simulation traces

The number of simulations to run.

### 2.4 Simulation timeunits

The amount of timeunits for each simulation.

### 2.5 Epsilon

The deadline of each job is computed as $job\ length * (1 + epsilon)$.

## 3 Job arrival and length parameters

### 3.1 Burst interval

The amount of timeunits between bursts of jobs.

### 3.2 Burst size

The amount of jobs per burst.

### 3.3 Nondeterministic job arrival/job length

The number of new job arrivals and their lengths can be nondeterministic. Each element has the same probability of selection.

#### 3.3.1 Nondet minimum

The minimum amount.

#### 3.3.2 Nondet maximum

The maximum amount.

### 3.4 Uniform value for job arrival/length

The number of jobs arriving each timeunit and their lengths can be uniform. There are separate values for job arrival and job length.

## 3.5 Wave job arrival/job length

### 3.5.1 Wave jobs/length per timeunit

The increment or decrement on the wave depending on whether it's ascending or descending.

### 3.5.2 Wave minimum

The lowest point on the wave.

### 3.5.3 Wave points

The number of points on the wave. For example jobs/length per timeunit 2, minimum 3 and points 6 generates 3, 5, 7, 9, 7, 5 and repeats.

## 3.6 Ascending job arrival/job length

### 3.6.1 Ascending increment

Additional jobs/length for each point.

### 3.6.2 Ascending minimum

The lowest point.

### 3.6.3 Ascending points

The number of points. For example an increment of 2, minimum 3 and points 4 generates 3, 5, 7, 9 and repeats.

## 3.7 Descending job arrival/job length

### 3.7.1 Descending decrement

Jobs/length decrement each timeunit.

### 3.7.2 Descending maximum

The highest point.

### 3.7.3 Descending points

The number of points. For example a decrement of 2, maximum 9 and points 4 generates 9, 7, 5, 3 and repeats itself. Jobs/length does not go below 0.

### 3.8 Exponential multiplier

Job length is computed as *current time* $*$ *exponential multiplier*. For example an exponential multiplier of 2 will start at timeunit 0 and generate job lengths of 0, 2, 4, 8, 16...

## 4 High priority job options

If two or more jobs are of the same priority, EDF is used to select between them.

### 4.1 Probability %

The probability of a job being of high priority.

### 4.2 Length

The length of high priority jobs. Selecting 0 will give high priority jobs the same length pattern as the low priority jobs.

## 5 Natjam-R options

### 5.1 Checkpoint overhead

The penalty in timeunits for each context switch. This includes both preempting the job and restarting it. The penalty is added at the time of preemption.

## 6 Input/Output Options

### 6.1 Output path

Output directories and artifacts will be created here.

### 6.2 Prefix

A prefix for the name of directories created.

### 6.3 Compiler path

The path to the files needed for compiling the Rebeca code. November 27, 2014, these files are:

- rmc-2.5.0-SNAPSHOT.jar
- g++.exe

- cygiconv-2.dll
- cygintl-3.dll
- cygwin1.dll

### 6.4 Traces path

The traces from a simulation will be output here. Must be on the same hard drive as the Output path.

## 7 Results window

Messages from threads and other components. Threads are named by their Output directory.

## 8 Run button

Runs a test using the set parameters. Is disabled until at least one dispatch policy, one job arrival type and one job length type is selected.
Multiple runs using different parameters can be run simultaneously if there is enough RAM available. Change the output and traces paths to avoid overwriting other results if doing multiple simultaneous runs.