an introduction to

Distem

Distem – http://distem.gforge.inria.fr/
Imagine you want to build a race car
But you want it to work well on both **dry** tracks and **wet** tracks.
You could use a **simulation**:

model the **car** and the **track**

and **compute** how they would interact
Or you could use a real world experiment: build a car, and try it on a real circuit
Or you could use a real world experiment: build a car, and try it on a real circuit.

But sometimes, rain is unlikely.
You could move to another race track . . .
But what if you could change the weather?
You could move to another race track . . .
But what if you could **change the weather**?

That’s what **Distem** does:
Take a platform, and alter it to suit your experimentation needs
An emulator for distributed systems

Take your real application

Run it on a cluster

And use Distem to alter the platform
so it matches the experimental conditions you need

Distem – http://distem.gforge.inria.fr/
Ideal complement to other tools

**SimGrid**

Toolkit for the simulation of distributed applications in heterogeneous environments

**Grid’5000**

Large-scale platform for experiments on distributed systems

1700 nodes, 7000 cores

Fully reconfigurable
Ideal complement to other tools

SimGrid

Toolkit for the simulation of distributed applications in heterogeneous environments

Grid’5000

Large-scale platform for experiments on distributed systems

1700 nodes, 7000 cores
Fully reconfigurable

Distem can run on Grid’5000
What can Distem do for you?

- Introduce heterogeneity in an homogeneous cluster
  How does your app perform when some nodes are slower?

- Emulate complex network topologies
  How does your app perform on a Grid? on a slower network?

- Inject faults and performance variations during the experiment
  How does your app behave when a node crashes?
  When the available CPU time decreases?
  When the available network bandwidth increases?
Introducing heterogeneity

- Distem splits real nodes into several virtual nodes
  - with a different number of cores
  - with different CPU performance
- Each virtual node can be used as a real Linux system

Emulating network topologies

- Emulate several local networks linked together
- Control available bandwidth and latency on each interface
Distem internals

Distem uses modern Linux features to *steal* resources from applications:
LXC, cgroups, cpufreq,iptables, traffic control

Distem – http://distem.gforge.inria.fr/
User interfaces

- **Command-line interface** – `distem` command
  - 🙂 Easy to use
  - 😞 Harder to automate
  - 😞 No access to more advanced features

- **Ruby library**
  - 🙂 Easy to automate
  - 🙂 Access to all features
  - 😐/🙂 Easy to use if you known Ruby

- **REST API**
  - 🙂 Can be used from any language
  - 😞 Requires REST knowledge
Overview of a Distem instance

User’s machine

Uses the command line interface, the Ruby client library or a REST client

Coordinator & Pnode 1

 Starts and controls other Pnodes
 Keeps the global state of the platform
 Is the gateway to Pnodes and Vnodes

Pnode 2

distemd

Pnode 3

distemd

Vnodes

Distem – http://distem.gforge.inria.fr/
Example experiment: SCP vs Rsync

- Transfer 50 files (total: 5 MB)
- Available bandwidth: 10 Mbps
- Varying network latency

![Graph showing transfer time vs emulated latency for SCP and Rsync]
Distem scales to thousands of virtual nodes

- Executing a command with ClusterShell and TakTuk

![Graph showing execution time vs. virtual nodes for ClusterShell and TakTuk.](http://distem.gforge.inria.fr/)
More information?

http://distem.gforge.inria.fr/