

Synthesizing generic experimental environments for simulation

Martin Quinson, Laurent Bobelin and Frédéric Suter

LORIA, CC IN2P3/CNRS

5th of November

Plan

- 1 Background
- 2 Experimental environment
- 3 Environment generation
- 4 Conclusion and future works

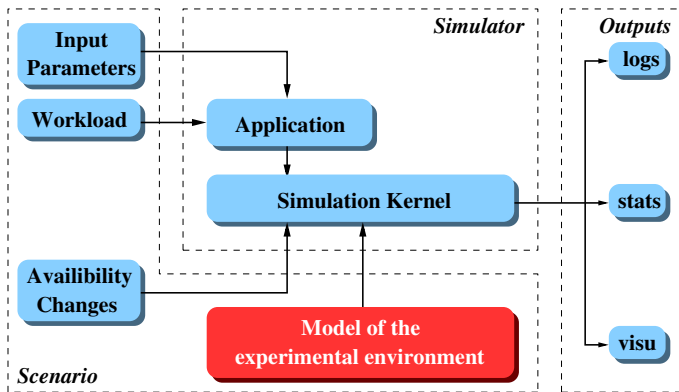
Distributed system simulation : overview

Simulation goals

- Test behaviour of an application,
- Performances of new algorithms,
- New mechanisms validation,
- Stress-tests,

Without the cost of effective testing on a real platform, and with the total control of platform topology, external workload on it, application parameters and so on.

Components of a simulation run and their interaction



We focus only on the **model of the experimental platform**.

Plan

- 1 Background
- 2 Experimental environment**
- 3 Environment generation
- 4 Conclusion and future works

Different levels of simulation from the experimental environment point of view

- Packet-level : Simulate low level effects of interaction between flows, dynamic (realistic) routing : ns-2, SSFnet...
- **High-level** : Simulate complex application behaviour. Considering flow-level network transfers, more (or less) static routing : **SimGrid**, GridSim, OptorSim ...
- Large-scale level : Simulate mass effects. No interaction between flows and transfers. No real network description because of simulation cost.

Experimental environment : different needs

- Packet level : complete description of the network. Network interface of each router and computer on the platform, bit rate, latency, per-packet external workload description (possibly with patterns).
- High-level : description of links in terms of flow-level significant parameters (latency and bandwidth), and platform and route descriptions.
- Large-scale level : Considering network with simpler parameters (complete graph with sometimes latency and bandwidth).

Plan

- 1 Background
- 2 Experimental environment
- 3 Environment generation**
- 4 Conclusion and future works

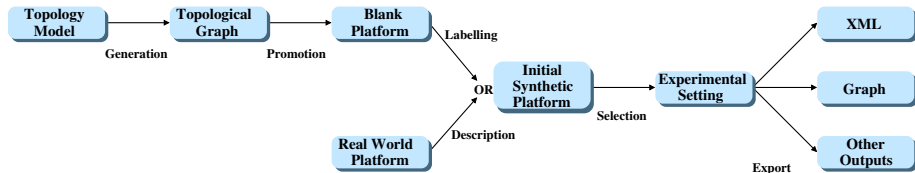
High-level simulation needs

- logical topology :
 - Graph generation,
 - Choice of which nodes to **promote** to the status of end-hosts,
 - Choice of a mechanism to label realistically links.
- Routing : description of routes for each possible paths

At the end of those steps : edges representing networks equipments and physical links, unlabelled routers and some nodes representing end hosts.

Simulacrum does this.

Simulacrum : environment generation flowchart



Simulacrum : From topology model to an blank platform

Topology model

- "Classic" topologies : ring, star, clique
- Random graph generation : uniform, exponential, Zegura, Barabasi and Albert

Node promotion

Two steps : **choice of the nodes to promote** and **generators for managing the transformation from a node to a resource**.

- Promoters : Defined by filters associated by AND or OR. Rules are based on node degree and/or platform properties (total number of resources for example).
- Generators : single host/homogenous cluster ? Compute speed ? ...

Edge labelling is done in a manner similar to node promotion.

Simulacrum final step : subset selection

Why?

At the end of this step, platform description can be imported from real life platform description OR from the previously described process :

- Random generation : maybe some generators have given undesired properties for some hosts and links.
- Real life platform description : possibly huge. Maybe only a subset of the platform described is usefull.

How?

- Filters to accept/discard nodes.
- rules based on structural properties (number of nodes, number of hosts, diameters) and/or statistical properties of computing power or user-defined properties.

At the end, an experimental environment is obtained.

Plan

- 1 Background
- 2 Experimental environment
- 3 Environment generation
- 4 Conclusion and future works

Conclusion and future works

- Simulacrum produces platform for SimGrid simulators, **by following a well-defined process**.
- More random generation model, including mainly hierarchical ones, are needed, as it reflects better the nowadays network topology.
- More parameters could be passed to the node promoters/edge labellers, as distance to end hosts for example.
- More interactive way for user to choose and design its experimental environment : need of an interactive visualisation of the platform produced.

Thanks for your attention

Questions ?