Methodology Convergence for Distributed Applications

Where do we stand?

Maximiliano Geier

AlGorille Group
INRIA Lorraine
Nancy, France

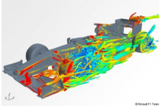
March 12, 2013

Introduction

- Understanding a distributed application is not easy
  - Parallelism
  - Complex, dynamic system
  - Multiple subsystems involved
    - Network
    - Disk
    - CPU
- Experiments help us understand their behavior

Motivation

- Three different paradigms

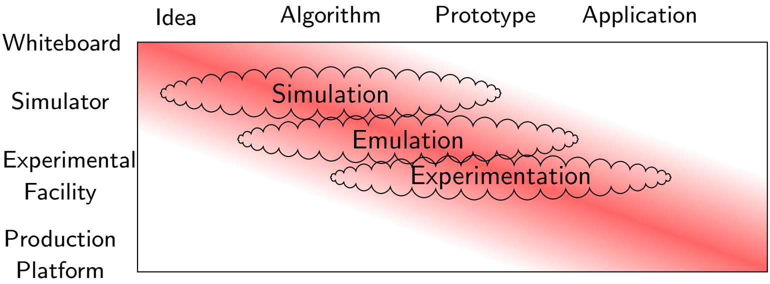


1. Simulation: prototypes of applications on system models
2. Emulation: real applications on synthetic platforms
3. Experimentation: real applications on a real platform

- Tools in each paradigm evolve independently
- We need an unified approach

Our Dream

Methodologies must be combined



Idea

Algorithm

Prototype

Application

Whiteboard

Simulator

Experimental Facility

Production Platform

Simulation

Emulation

Experimentation

- Share experiments description, DoE and visualization tools
- Be able to switch seamlessly to the most apt tool

Thanks to Martin for this slide!

Coming up next . . .

- Tool choice
- Methodology description
- Issues arised from using the tools
- Potential improvements
- Future work
- Conclusions

The Tools

1. Simulation
   - Simgrid: more than 100 papers built on it
2. Emulation
   - Distem: new platform based on Wreakavoc
3. Experimentation
   - Grid'5000: proven platform, 11 sites all over France and Luxembourg

- All of them are good for the job
- We can concentrate on the methodology

The Methodology

Usual approach

- Questions are the driver
- Methodology comes afterwards
- Result: interesting question, lacking methodology

Our approach

- Trivial question
- Focus on methodology convergence

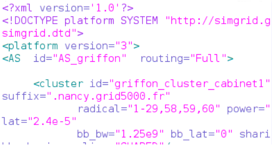
Question:
How do chain propagation algorithms perform to broadcast files in local area networks?

- We (kind of) already know the answer
- Kastafior

The Traps

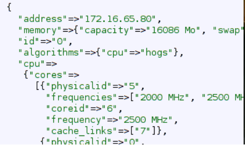
Trap: It's difficult to get the platform right

- Information accuracy
- User mistakes
- Mapping among the tools
  - Distem can load Simgrid v2 platforms, not there yet



```
<?xml version='1.0'?>
<!DOCTYPE platform SYSTEM "http://simgrid.g
simgrid.dtd">
<platform version="3">
<AS  id="AS_griffon"  routing="Full">

     <cluster id="griffon_cluster_cabinet1"
suffix=".nancy.grid5000.fr"
             radical="1-29,58,59,60" power="
lat="2.4e-5"
             bb_bw="1.25e9" bb_lat="0" shari
```

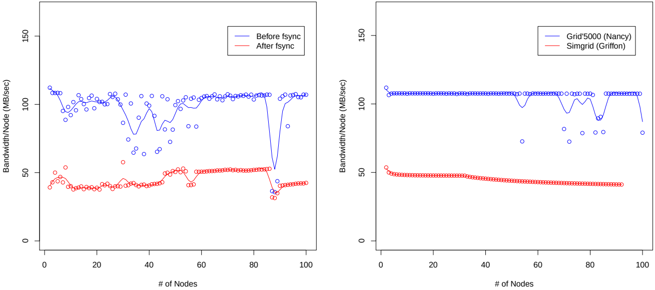


```
{
  "address"=>"172.16.65.80",
  "memory"=>{"capacity"=>"16086 Mo", "swap"
  "id"=>"0",
  "algorithms"=>{"cpu"=>"hogs"},
  "cpu"=>
   {"cores"=>
     [{"physicalid"=>"5",
       "frequencies"=>["2000 MHz", "2500 MH
       "coreid"=>"6",
       "frequency"=>"2500 MHz",
       "cache_links"=>["7"]},
      {"physicalid"=>"0",
```

The Traps (cont.)

Trap: Working with abstractions

- Simgrid
  - Transport protocol
  - Connection flow
  - Model parameters that effect flow
  - Simulated hardware
- Distem
  - Node mapping
  - Model parameters that depend on the mapping

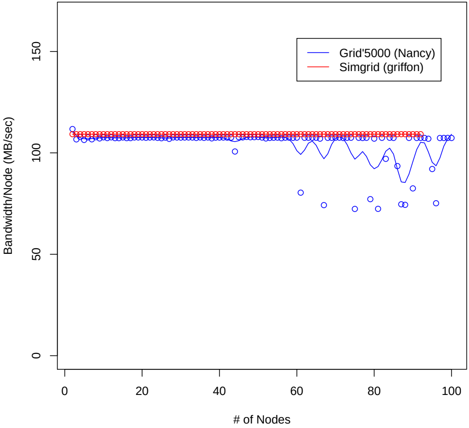


The Traps (cont.)

Trap: Model complexity

- Impossible to take everything into account
- Work application around abstractions
- Simplify aspects that are not interesting to analyze
    - Code bloat
    - Bugs

Working it out



- Very similar results in Simgrid and Grid'5000
- Distem work is still ongoing

Potential Improvements

Working with different platforms

- Client input is usually a list of hosts
- Converge name handling
    - Automatic DNS or `/etc/hosts` setup
        - Simgrid: it's already there
        - Distem: virtual node names
        - Grid'5000: OAR nodes
    - Doable by hand, but better if the tool does it for you

Potential Improvements (cont.)

Platform convergence

- Reduce error-prone work duplication
    - Extensible platform description for Simgrid
        - Already using XML, but not really extensible
    - Converters
        - JSON (Distem) to XML (Simgrid)
        - XML to JSON

Towards an integrated framework

- Understand power and limitations of the tools
- Use tools to manage experiment workflow
  - Easier to work with several experimental frameworks
- Identify application traits
  - Type of traffic
  - Transport protocol
  - Bottlenecks
  - etc.
- Corroborate models whenever possible

Future Work

- Check platforms: *linktest*
  - Latency
  - Bandwidth
  - Tests go three-way
- Converge model descriptions
- Get Distem into the picture
- Work on visualization aspects to improve analysis

Conclusions

- Each paradigm shows a different perspective
- All of them are useful in different ways
- We should try to make it easier to use all approaches
  - Tool designers
  - Users
- Tools to manage experiments
- Abstraction and modularization

The End

Questions?