Lightweight Emulation to Study Peer-to-Peer Systems

Lucas Nussbaum and Olivier Richard

Laboratoire
Informatique et
Distribution

Laboratoire ID-IMAG
MESCAL INRIA Project
Grenoble, France

Outline

1. Introduction : Study of Peer-to-Peer Systems
2. P2PLab
3. Conclusion

Outline

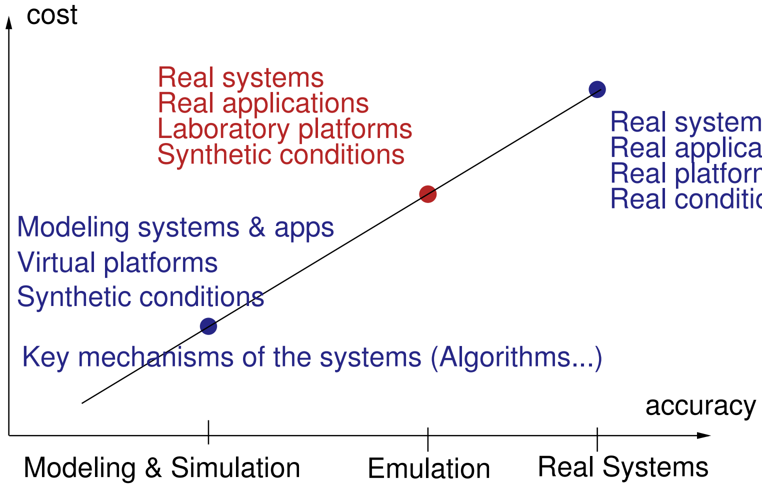
1. Introduction : Study of Peer-to-Peer Systems
   - Peer-to-peer systems
   - Solutions for P2P Study
   - Summary
2. P2PLab
3. Conclusion

Peer-to-Peer systems

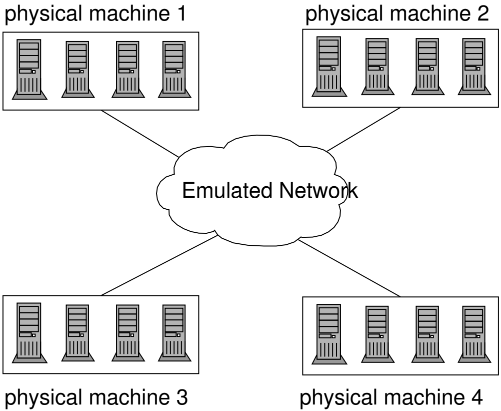
- Heterogeneous nodes (CPU, network)
- Non-persistent nodes
- Difficult (Impossible ?) to control

⇒ Hard to study (correctness, performance) and to *understand*

Solutions for P2P Study



Emulation



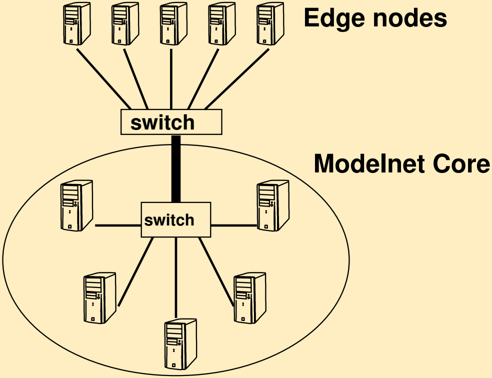
Virtualization on the nodes

Existing tools

Existing tools

Modelnet, NetBed/Emulab, MicroGrid, PlanetLab

Example : Modelnet



Existing tools (2)

Example : PlanetLab

700 distributed nodes available as a testbed for distributed systems research. But :

- 700 « millions
- Nodes hosted by universities, research centers, big corporations. "Real" Internet ?

Tools for emulation and virtualization

Network emulation

Change network link characteristics (bandwidth, latency, congestion/packet loss)

- NIST Net (Linux 2.4 and 2.6)
- TC and Netem (Linux 2.6)
- **Dummynet** (FreeBSD)

Virtualization

Execute several instances of an application on the same physical system

- User Mode Linux
- **VServer**
- **Xen**

Summary

- Solutions aim at very high accuracy
  - Efficiency ?
  - Scalability ?
- Virtualization of a full operating system
  - Really necessary for P2P systems ? (vertically integrated apps)
- Network emulation targeting the network core
  - Really important for P2P systems ? (applications running on the edge of the Internet)

Outline

1 Introduction : Study of Peer-to-Peer Systems

2 P2PLab

- Introduction
- P2PLab : process-level virtualization
- P2PLab : network emulation
- P2PLab : Evaluation
- Virtualization ratio
- Scalability
- Fairness of virtualization

3 Conclusion

P2PLab : Introduction

Main idea

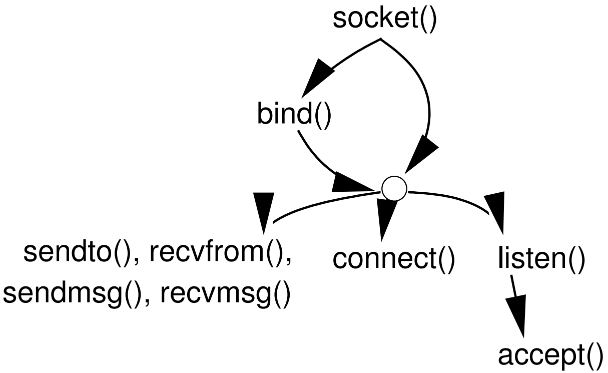
- light and efficient emulation system (good virtualization ratio)
- Targeting peer-to-peer systems specifically

Key facts

- Uses FreeBSD 5 (for Dummynet)
- Process-level virtualization (by virtualizing the network identity)
- Decentralized network emulation for good scalability

P2PLab : process-level virtualization

- Affect an IP address to each process, for both *clients* and *servers*
- Modification of the Libc (`bind()`, `connect()` and `listen()`) to always issue a `bind()` on the address specified by `$BINDIP`


socket()

bind()

sendto(), recvfrom(), sendmsg(), recvmsg()

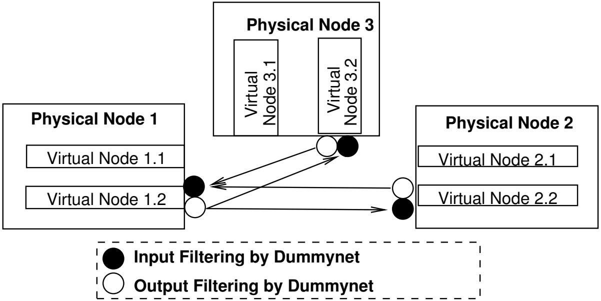
connect()

listen()

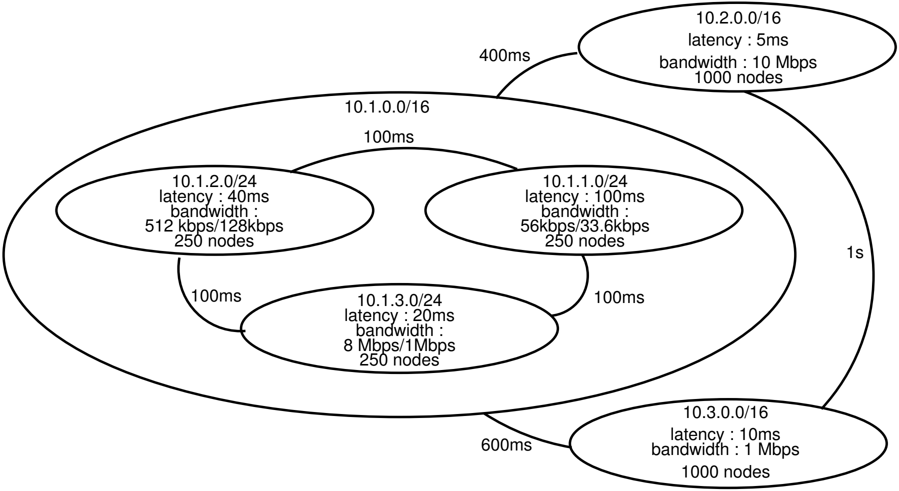
accept()

P2PLab : network emulation

- Uses interface aliases (minimal overhead)
- Manages network emulation on the nodes (input & output)
  - Limits bandwidth and adds latency for each virtual node
  - Adds latency between groups of virtual nodes



Emulation of network topologies



P2PLab : Evaluation

- Virtualization ratio
  - How many virtual machines on each physical machine ?
- Scalability ?
  - Experiments with a *large* number of nodes ?
- Fairness of virtualization ?
  - Which level of fairness between two processes from the same physical node ?

P2PLab : Evaluation (2)

BitTorrent

- popular file-sharing protocol
- each client gets a list of *peers* from the *tracker*, then connects to them directly and exchanges pieces of a file
- Largely studied through modelling, simulations and runs on real systems
- very complex, difficult to study it accurately

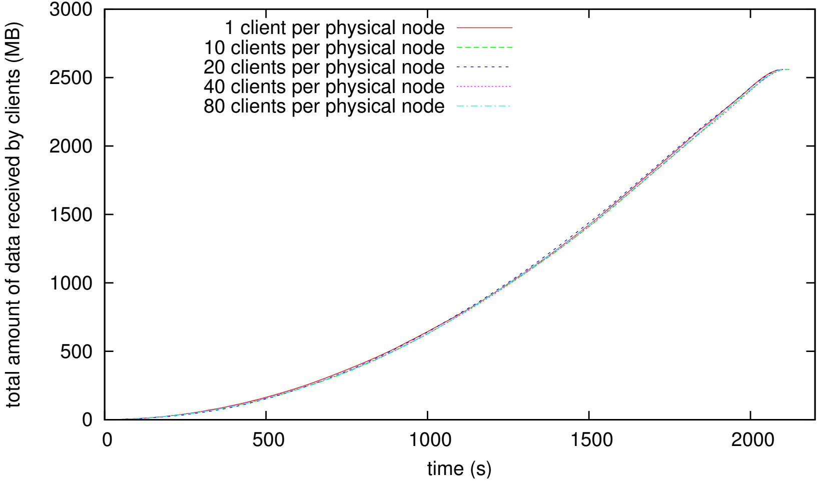
Evaluation system

- GridExplorer cluster (part of the Grid'5000 project)
- ˜200 bi-Opteron 2 Ghz, 2GB of RAM, gigabit ethernet network

Virtualization ratio

- Download of a 16 MB file with BitTorrent
- 160 downloaders
- Started every 10 seconds
- Network settings : 2 mbps down, 128 kbps up, latency 30 ms
- On 160 physical nodes, then 16, then 8, then 4, then 2.

Virtualization ratio (2)



Virtualization ratio (3)

- Download of a 16 MB file with BitTorrent
- 160 downloaders
- Started every 10 seconds
- Network settings : 2 mbps down, 128 kbps up, latency 30 ms
- On 160 physical nodes, then 16, then 8, then 4, then 2.

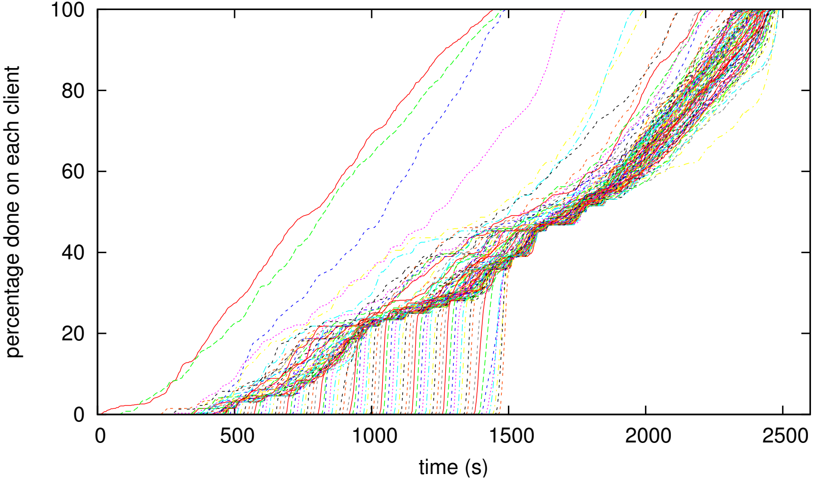
⇒ No visible overhead even with 80 clients per physical node
⇒ First limiting factor : performances of the underlying network

Scalability

- Download of a 16 MB file using BitTorrent
- Between 5760 peers (5755 downloaders, 4 seeders, 1 tracker)
- Started every 0.25s
- On 5760/32 = 180 nodes
- Network settings : 2 mbps down, 128 kbps up, latency 30 ms

Clients displayed on the graph : no 0, 50, 100 ... 5750.

Scalability (2)



Fairness of virtualization

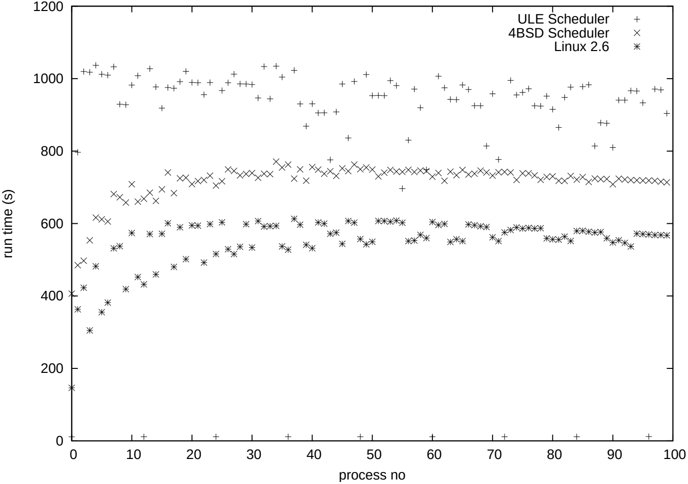
Question

Which level of fairness can be expected between two processes from the same physical machine ?

Experiment

- Starting a process every second
- When alone, needs 12 seconds to complete
- Result : time needed for each process to complete

Fairness of virtualization (2)



Outline

1 Introduction : Study of Peer-to-Peer Systems

2 P2PLab

3 Conclusion

Conclusion

- Emulation is an interesting tool when studying peer-to-peer systems
    - Heavy virtualization and emulation is not always needed
- ... and P2PLab is a useful emulation system, using a simple approach
- Still needs a lot of work :
    - Comparison with other tools (Modelnet, NetBed/Emulab, MicroGrid, PlanetLab)
    - Comparison with classical topology generators
        - What's the importance of congestion in the Internet core for P2P systems ?
    - Virtualization and fairness
    - Move to Linux ?
    - Realistic parameters for experiments ?
        - Node failure and departure, varying network conditions

Questions ?

- Q : Is P2PLab available ?
  - A : Currently, no
  - A : But I'm interested in collaborations : if you have a working prototype you would like to evaluate using P2PLab, contact me (`lucas.nussbaum@imag.fr`) !
- **Other questions ?**