An Overview of the Options Available for Practical Activities in Distributed Computing

Camille Coti, Jean-Vincent Loddo, Emmanuel Viennet

IUT de Villetaneuse, Université Paris 13, Sorbonne Paris Cité

July 11th, 2017
ITHET 2017

Roadmap

1. Introduction
   - Distributed systems?
   - Expectations
2. Use-cases
3. Solutions compared
4. Conclusion

Teaching distributed systems? Examples

Client-server programming

- Need to processes to interact through the network
- Sometimes, they really need to be located on two different machines

Distributed computing (MPI...)

- Run several processes
- On a set of resources: can be the cores of a multi-core machine, of a set of machines...

System administration

- Administrate a set of machines, use distributed algorithms (spanning tree, Bellmann-Ford...)
- Use servers, network components...
- Actually need to have different resources

→ Using a distributed system

→ Programming on a distributed system

What is expected?

Realistic feedback

- Practical activities: look how things happen
- Let the students experiment with the system
- Realistic procedures

Cost

- Well...

Performance

- Must be realistic too
- Not too slow!

Undergrad: network programming

IUT: vocational program, 2 years, 1 optional third year

- Labs need to be **realistic** , skills need to be transposable in the real world
- Department: networks and telecommunications
  - Signal transmission
  - Network configuration and administration
  - System administration

Some classes and topics covered :

- **Local networks** : configure local equipment, spanning tree protocol...
- **Network services** : configure and administrate servers (DHCP, NIS, LDAP, SMTP/IMAP...)
- **Routing** : configuration of static routers, dynamic routing (RIP, OSPF, BGP, IS-IS), distributed algorithms (Bellmann-ford, Dijkstra)
- **Network programming** : client-server

Graduate class: HPC

Masters's degree in computer science: class on high performance computing

- Distributed programming techniques: MPI, MapReduce...
- Distributed algorithms
- Computation center (cluster) usage
- Cluster administration

Two needs identified :

- Using distributed resources
- Knowing procedures implemented on clusters (batch scheduler, automatic node deployment...)

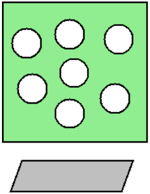
Running multiple processes on a single machine

Running **multiple processes on a single machine**

- Every student has one computer in from of him
- everything is done here

Very basic

- Simple and inexpensive
- **Not very realistic** , not always possible



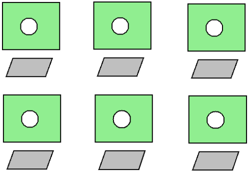
Using the computers of the lab room

Using the **computers of the lab room**

- Many resources available
- Linked together by a network
- Already there!

Very convenient , (almost) nothing to do

- shared file system
- performance vary with usage
- not always root



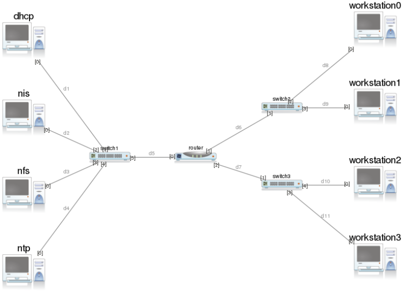
Virtualization

Virtualization : for instance, Marionnet

- Goal: provide a **virtual network laboratory** to each student
- No specific hardware required
- Each students works on his/her own network
- Configure all the network elements like real hardware
- Virtualized: User-Mode Linux, VDE
- Developed at Univ. Paris 13

Fine for network labs

- **Realistic**
- Performance issues



Using a cluster

Using a cluster

- For instance, the USPC computation center
- 50 40-core notes, SLURM batch scheduler
- Users from all the organizations of the COMUE: 14 universities, schools and research organizations

Shared resources, with usage procedures

- **Most realistic solution for HPC** , not the other ones
- **Queuing issue** : what if the queue is full?
- Asynchronous submissions: not sure when the result will arrive
- Non-interactive

Virtual, teaching cluster

Improve the previous solution with dedicated resources: **virtual, teaching cluster**

- Reserve a few nodes for the class
- Deploy several virtual nodes on each physical node

Realistic cluster experience

- better than the previous solution
- performance issue, depends on the virtualization solution
- cost!

Renting resources on a cloud

Renting resources on a cloud

- Infrastructure as a Service:
- rent “machines”, deploy whatever you want on them
- EC2, Oracle Cloud...

Attractive...

- realistic
- performant
- flexible
- cost
- **administrative mayhem** !!!

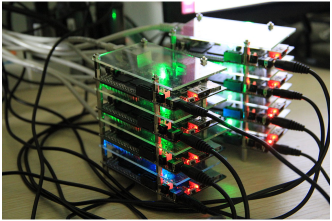
Cluster of cheap computers

Using a **cluster of cheap, single board computers**

- Inexpensive computers: Raspberry Pi...
- Very flexible: users can reconfigure them, a system can be copied...

Each student can have its own!

- **Inexpensive** , but still has a cost
- Flexible
- Not very powerful: not realistic for HPC
- Excellent for system administration



Conclusion

No perfect solution

- Depends on the needs

Network programming :

- Machines in the room

Network administration :

- Virtual laboratory, cluster of Raspberry Pi

High performance computing :

- Virtual cluster, machines in the lab room