

Cloud Computing

Prof. Tahar Kechadi

School of Computer Science & Informatics

Organisation

- **Lectures**

- Lectures will be made available on Mondays and Wednesdays during the semester.
- The lectures will be accompanied by tutorial/practical work.

- **Practical**

- Practicals will be submitted each Friday.
- Course material: **Moodle** (Virtual Learning Environment)

- **Practicals and Projects**

- Tutorial and practical sessions will be organised to help you in your project work and to
- Give some more details about certain topics
- Projects will be posted on Moodle
- The projects will be submitted by the end of the semester (the last Friday of the term)

Assessment

- **Continuous Examination**
 - Worth 30% of the overall mark
 - 3 MCQs of 30 minutes each
 - MCQ 1 : will take place on February 14th
 - MCQ 2 : will take place on March 7th
 - MCQ 3 : will take place on April 25th
- **Tutorial/Practical & Learning Journal**
 - Worth 30% of the overall mark
- **Final Exam**
 - Worth 40%

Course Outline

- Motivation: Why Cloud Computing?
- What is cloud computing?
- Virtualisation
- IaaS, PaaS, and SaaS
- Data storage in the Cloud
- Hadoop and MapReduce
- Case Studies: AWS, EC2, S3, etc.
- Migration in the cloud
- Cloud Security
- Advanced Topics

References

- Cloud Computing: SaaS, PaaS, IaaS, Virtualisation,...
 - Kris Jamsa, Jones & Barnett Learning, 1st Edition, 2013
- Cloud Computing Explained: Impl. Handbook for Enterprises
 - J. Rhoton, 2013 Edition
- Cloud Computing & SOA Convergence in Your Enterprise: A *Step-by-Step Guide*,
 - D.S. Linthicum, Addison Wesley, 2010
- Cloud Application Architectures: Building Applications & Infrastructure in the Cloud
 - G. Reese, O'REILLY, 2009
- Cloud Security & Privacy: An Enterprise Perspective on Risks & Compliance,
 - T. Mather, S. Kumaraswamy, S. Latif, O'REILLY, 2009
- Enterprise Cloud Computing: A strategy Guide for Business & Technology Leaders,
 - A. Mulholland, Meghan-Kiffer Press, 2009

Why Cloud Computing?

- **Understand the cloud principles**
 - Very different environment and many very interesting algorithms
 - How does it work? What are its strengths & weaknesses?
- **Learn new technologies**
 - MapReduce, Ajax, NoSQL, DHT, etc.
 - How to build a system such as Facebook?
- **Build complex systems**
 - Need scalable, efficient, robust and secure systems

Why Cloud Computing?

- **Cloud is the technology of the future**
 - Need to understand the current state of the technology
 - Anticipate what's possible in the future
- **Understand the cloud impact on society**
 - Vulnerability, privacy concerns, data survivability, etc.
- **Hands-on on cloud technologies**
 - The best way to understand the technology
 - Gain practical experience and skills in cloud technologies

Need for Cloud Computing

- **The Big Data Problems**

- Web-scale problems
- Very large data centres
- Different models of computing

- **How big the data is?**

- Google processes 20PB of data a day in 2008. More than 1 billion unique users use Google every month (in 2013)
- 5 EB is the number of all words even spoken by human beings
- CERN's LHC generates 15 PB per year.

- **Process & store exabytes of data**

- Banks, financial firms, academia, governments, military, etc.

Big Data Problems

- Web-scale Problems

- Data intensive
- Processing intensive

- Examples

- Crawling, indexing, searching, mining
- Scientific data (astronomy, particle physics, etc.)
- Sensor networks
- Web 2.0 applications.

Big Data Problems

- Large Data centres

- Redundancy
- Efficiency
- Utilisation
- Management
- Centralised computing resources !!



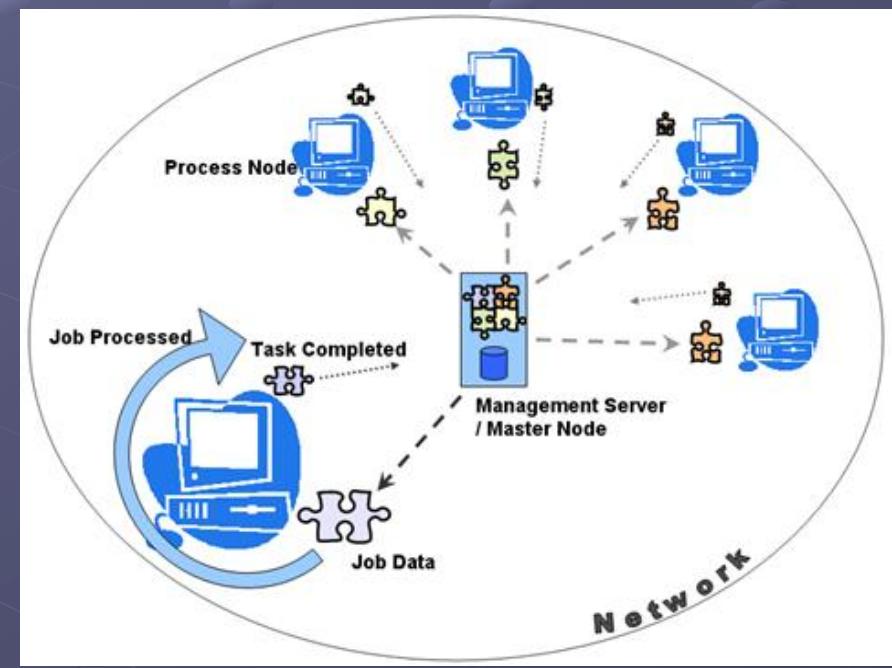
Centralised vs Distributed

- **Centralised system**

- Early computing was performed on a single processor; uni-cpu computing can be called centralised computing
- Have the overall supervision of computations

- **Distributed system**

- Collection of independent computers, interconnected via a network, capable of collaborating on a task.

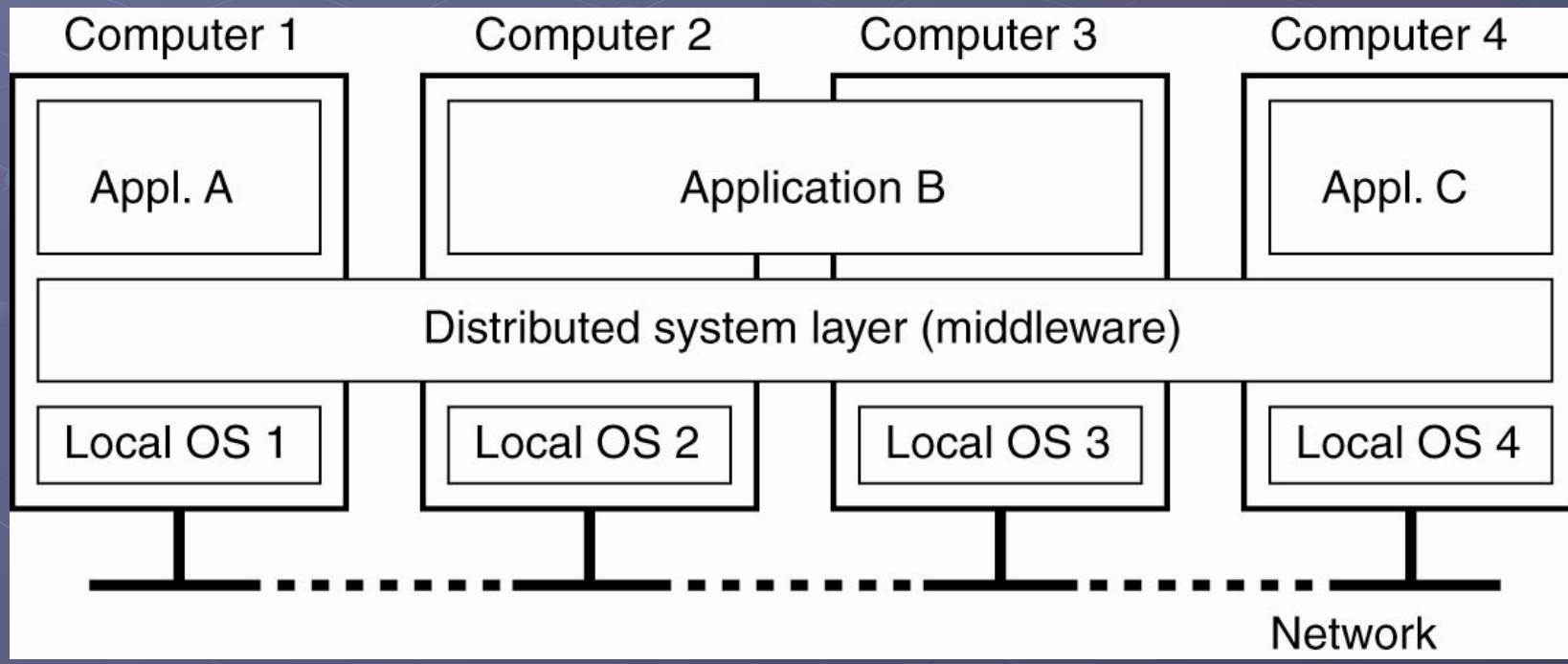


Centralised vs. Distributed

- *Distributed computing* has become increasingly common for several reasons:
 - Bigger challenges in science and industry
 - Performance (Task farming / Parallelisation)
 - Resource sharing
 - Scalability
 - Availability
 - DS are much cheaper than mainframes

Components

- **Middleware ~ Distributed OS**
 - Middleware technologies based on threads / sockets, RPC, Java RMI, SOAP/Web Services, etc.



Parallelism, Flynn's taxonomy

Parallel is distributed

...

Distributed,
not
necessarily
parallel (in
terms of
concurrency)

Data

Multiple (MD) Single (SD)

Instructions

Single (SI)

Multiple (MI)

SISD

Single-threaded
process

MISD

Pipeline architecture

SIMD

Vector Processing

MIMD

Multi-threaded
Programming

Challenges

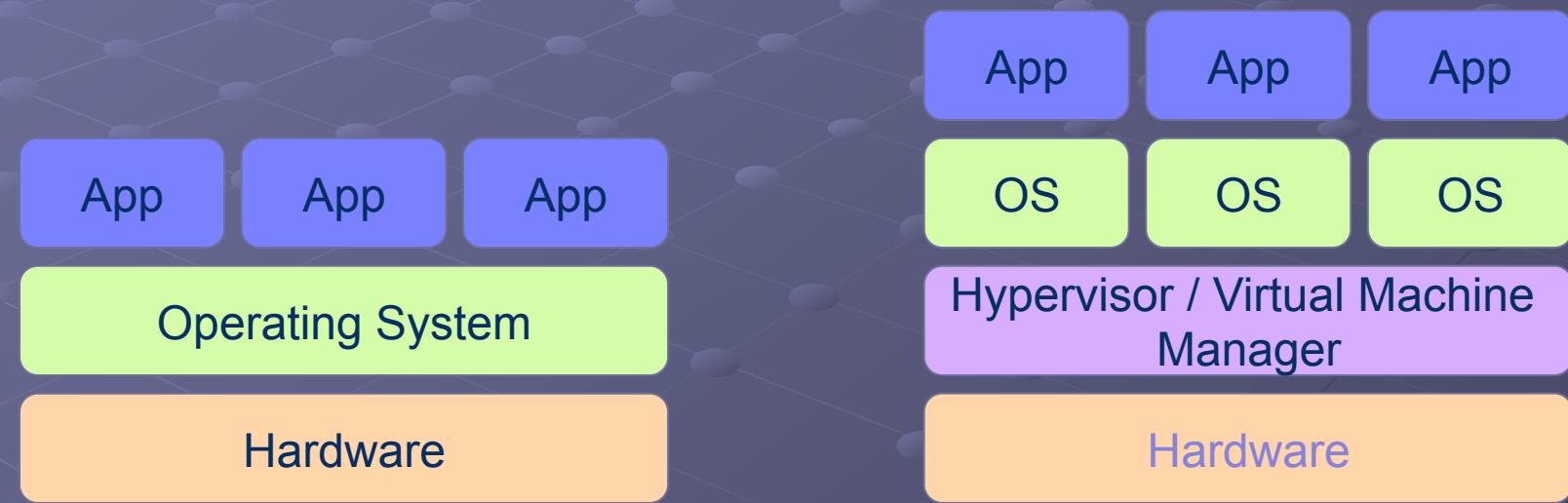
There are still many challenges in distributed / parallel computing:

- Heterogeneity
- Latency
- Remote vs. local memory
- Synchronisation & communications in computing
- Failures
- Security, ...

The Cloud?

- An example of a distributed system
- Outsourcing of hardware and software
- Large on-demand computing resources and data centres
- Different models of delivery and computing
- Highly interactive web applications and services

Key Technology: Virtualisation



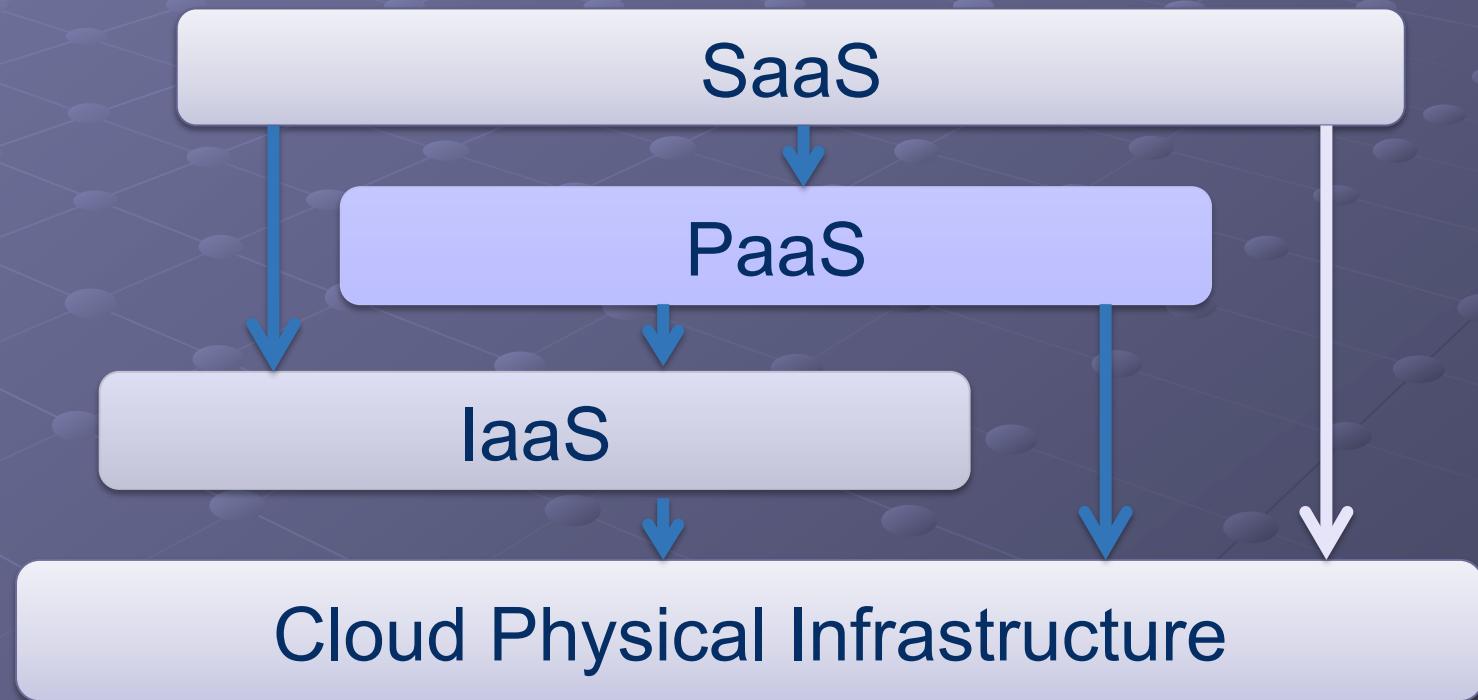
Traditional Stack

Virtualised stack

Different delivery Models

- Infrastructure as a Service (IaaS) / Utility computing
 - Why buy machines when you can rent cycles?
 - Examples: Amazon's EC2, GoGrid, Rackspace, etc.
- Platform as a Service (PaaS)
 - Give me nice API and development infrastructure
 - Example: Google App Engine, Cloud Foundry
- Software as a Service (SaaS)
 - Just run it for me!
 - Example: Gmail, Salesforce

Different delivery Models: Hierarchy



Different types as well

- **Public cloud**

- Web API allowing any user to launch virtual machines / services remotely on the cloud

- **Private cloud**

- Only users from your facility can use the cloud

- **Community cloud**

- Only users from a community can use the cloud (Magellan)

- **Hybrid cloud**

- Infrastructure built from mix of public, community and private clouds

Remember, resources are always provisioned “on-demand”, and “shut down” when no longer required