Distributed and Pervasive Systems

Percorso Mobility and Pervasive Computing
Corso di Laurea Magistrale in Informatica

Prof. Claudio Bettini
EveryWare Lab, Dipartimento di Informatica
Università degli Studi di Milano

1

Copyright

2

Introduction

3

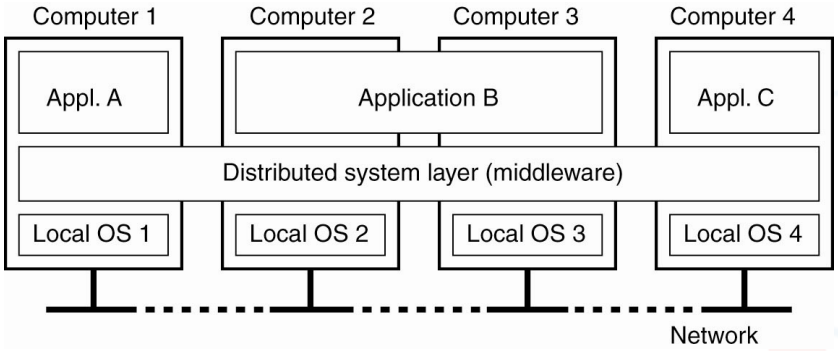
Definition of a Distributed System (1)

- A distributed system is:

  A collection of independent computers that appears to its users as a single coherent system

4

Definition of a Distributed System (2)



A distributed system organized as middleware. The middleware layer extends over multiple machines, and offers each application the same interface.

5

Examples

- A set of networked PC with shared file system, and process execution dynamically assigned to one of them
- Massively multiplayer online games with cooperating components distributed over different devices/computers
- WWW: distributed management of uniquely identified documents

6

Definition of a Distributed System (3)

Definizione ironica di Leslie Lamport:

"*You know you have one when the crash of a computer you've never heard of stops you from getting any work done*"

7

Goals of a distributed system

- Making resources accessible
- Distribution transparency
- Openness
- Scalability

8

Transparency in a Distributed System

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Different forms of transparency in a distributed system

Openness

- **An open distributed system should offer:**
  - Interoperability
  - Portability
  - Extensibility

Openness

- **Openness can be achieved by:**
  - the use of standard protocols
  - the publication of key interfaces
  - testing and verifying the conformance of components to published standards

Scalability Problems

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Examples of scalability limitations.

Scalability Problems

- Characteristics of decentralized algorithms:
  - No machine has complete information about the system state.
  - Machines make decisions based only on local information.
  - Failure of one machine does not ruin the algorithm.
  - There is no implicit assumption that a global clock exists.

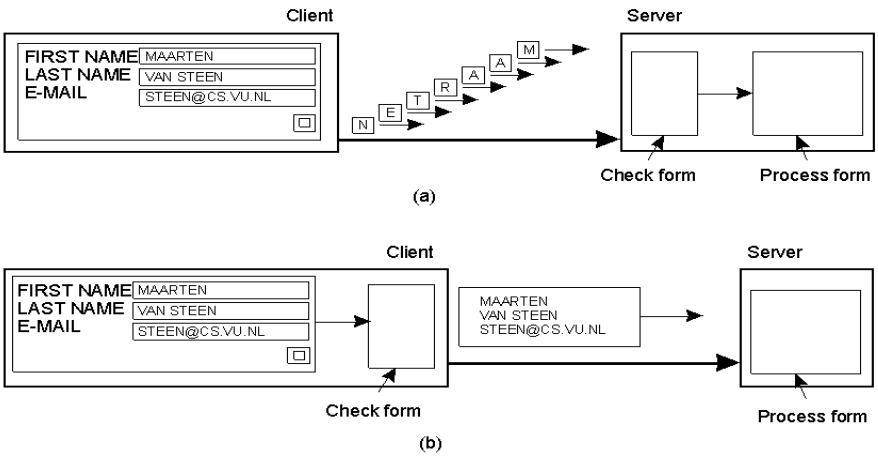
Scaling Techniques (1)

The difference between letting:

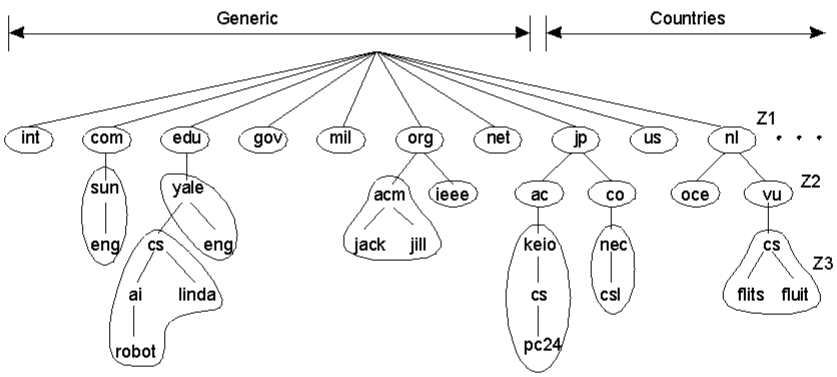
a server (a) or

a client (b)

check forms as they are being filled



Scaling Techniques (2)



An example of dividing the DNS name space into zones.

15

Pitfalls when Developing Distributed Systems

- False assumptions made by first time developer:
  - The network is reliable.
  - The network is secure.
  - The network is homogeneous.
  - The topology does not change.
  - Latency is zero.
  - Bandwidth is infinite.
  - Transport cost is zero.
  - There is one administrator.
  - Debugging distributed applications is analogous to standard applications

16

"*The moment we added our second server, distributed systems became the way of life at Amazon. When I started at Amazon in 1999, we had so few servers that we could give some of them recognizable names like "fishy" or "online-01". However, even in 1999, distributed computing was not easy. Then as now, challenges with distributed systems involved latency, scaling, understanding networking APIs, marshalling and unmarshalling data, and the complexity of algorithms such as Paxos. As the systems quickly grew larger and more distributed, what had been theoretical edge cases turned into regular occurrences.*"

[Jacob Gabrielson, Senior Principal Engineer at Amazon, *Challanges with distributed systems*, 2019]

Types of Distributed Systems

- Distributed Computing Systems
  - Clusters
  - Cloud Computing
  - Edge Computing

Types of Distributed Systems

- Distributed Information Systems
  - Distributed Databases
  - Distributed transactions
- Distributed Pervasive Systems

19

Distributed Computing Systems

20

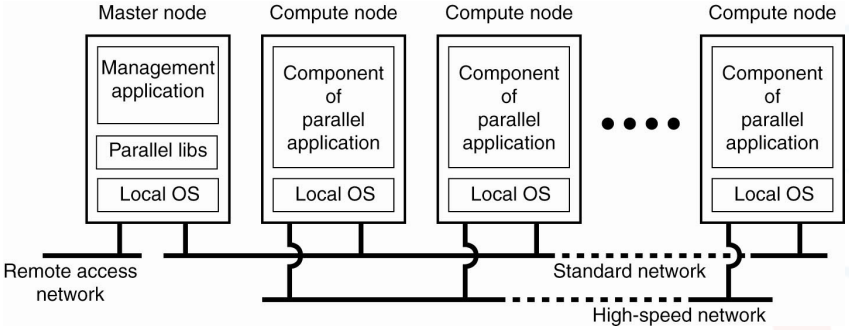
Cluster Computing Systems

- A collection of equal or similar workstations/servers closely connected by high-speed local-area network and usually running the same operating system.
- Goal:
  - High performance computing tasks, or
  - High availability

Symmetric and asymmetric clusters

- Asymmetric approach
  - A master node distributes tasks on a set of computing nodes.
    - Example: Beowulf for Linux clusters, Google Borg (large scale cluster management)
- Symmetric approach
  - There is no master, all nodes have the same software installed.
    - Example: MOSIX, based on transparent process migration among nodes

Example of asymmetric cluster



24

The case of Google Borg

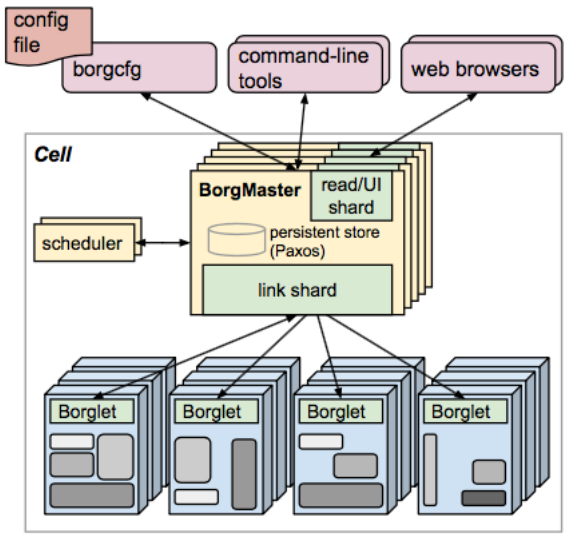
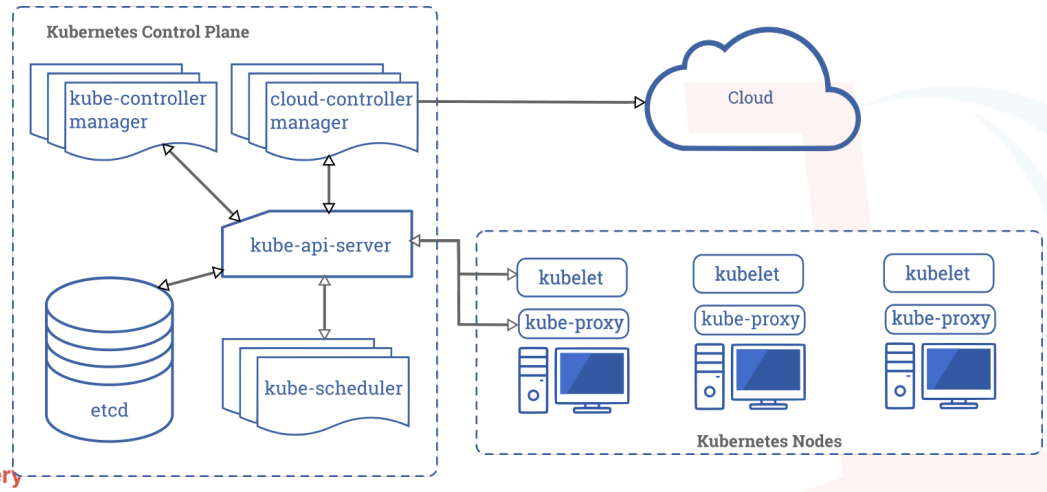


Figure from 'Large-scale cluster management at Google with Borg', Verma et al., Google.

25

Kubernetes



Cloud Computing

- Cloud computing is a model for enabling **ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources** (e.g., networks, servers, storage, applications, and services) **that can be rapidly provisioned and released with minimal management effort or service provider interaction**.

  (NIST, Special Publication 800-145, 2011)

Cloud characteristics

- Nodes are etherogenous in hardware and operating system
- Network connections are etherogenous in their capacities and reliability
- On-demand self-service. A consumer can unilaterally provision computing capabilities
- Broad network access. Capabilities are available over the network and accessed through standard mechanisms

Cloud characteristics

- Resource pooling. The provider's computing resources are pooled to serve multiple consumers
- Rapid elasticity. Capabilities can be elastically provisioned and released
- Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability

Cloud Service Models

- Software as a Service (SaaS).

  Use the provider's applications running on a cloud infrastructure

- Platform as a Service (PaaS).

  Deploy onto the cloud infrastructure applications created by consumer using tools supported by the provider

Cloud Service Models

- Infrastructure as a Service (IaaS).

  Deploy and run on cloud infrastructure arbitrary software (including operating systems and applications)

Cloud Deployment Models

- **Private cloud**. exclusive use by a single organization comprising multiple consumers
- **Community cloud**. exclusive use by a specific community of consumers from organizations that have shared concerns
- **Public cloud**. open use by the general public.
- **Hybrid cloud**. composition of two or more distinct cloud infrastructures (private, community, or public)

Cloud solutions

- Main (proprietary) public cloud providers
  - Amazon AWS, Google Cloud platform, Microsoft Azure
- Opensource cloud software
  - OpenStack (to create public and private clouds)
- Example of infrastructure as a service
  - Amazon EC2-Elastic Compute Cloud
- Example of platform as a service
  - Google App Engine
- Example of software as a service
  - Google Docs, Microsoft Office365

Cloud computing infrastructure

Example: The Google infrastructure

Picture from
http://media.charleston.net/img/photos/2008/03/06/bizlede_google.jpg

Every Ware Lab

Cloud computing infrastructure

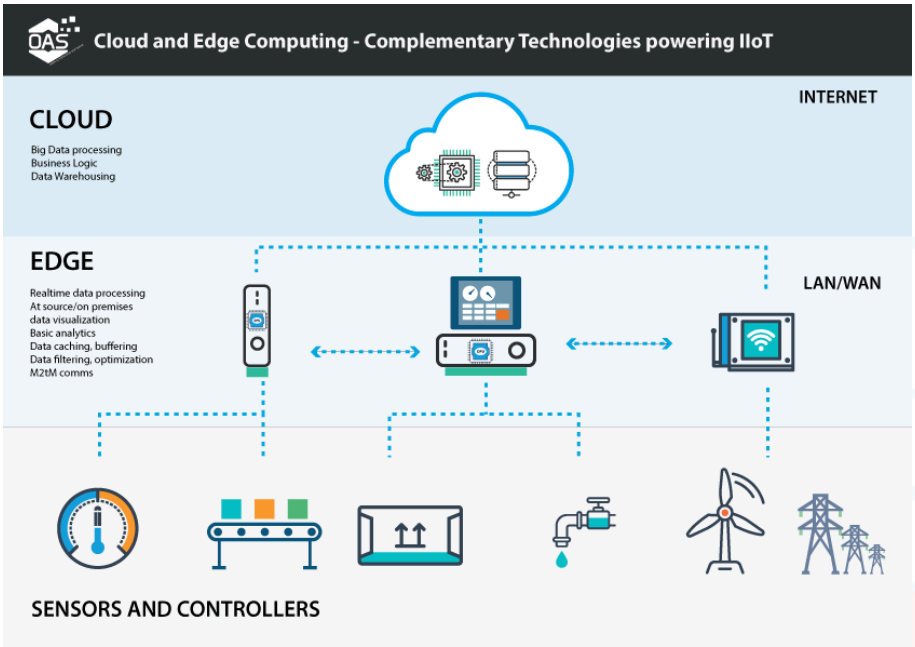
- Special switching and load balancing technology
- Each data center has one or more clusters of thousands of nodes (e.g., PCs running a customized Linux operating system)

Every Ware Lab

Edge (or Fog) Computing

- IOT: sensorized connected devices produce massive amounts of data that must often be processed in real time
- There is a need for reliability and low latency
  - Example: a car needs to obtain data from local environment (e.g., traffic light) in 1 ms and to take quick decisions
- 5G networks provide direct and efficient network connection, but ...
- Processing all data in the cloud does not meet stringent latency requirements. Communication to the cloud also requires energy.

Every Ware Lab



Every Ware Lab

Fog vs Edge Computing

- "fog" is often used as a synonym of "edge"
- "fog" is sometimes used differently, to refer to an intermediate level of decentralization between cloud and edge using more powerful nodes

Distributed Information Systems

- Distributed Databases
  - Transparent distribution of data and query answering
  - Examples: NoSQL and newSQL DBMS, Blockchain
- Transaction Processing Systems
  - Requests directed to multiple servers gathered in a single transaction (ensuring ACID properties in a distributed environment)
  - Transactions can contain RPC

Transaction Processing Systems (1)

- Example primitives for transactions

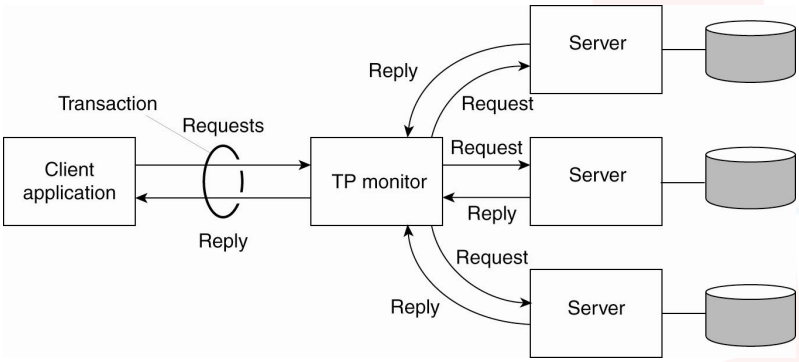
Primitive	Description
BEGIN_TRANSACTION	Mark the start of a transaction
END_TRANSACTION	Terminate the transaction and try to commit
ABORT_TRANSACTION	Kill the transaction and restore the old values
READ	Read data from a file, a table, or otherwise
WRITE	Write data to a file, a table, or otherwise

Transaction Processing Systems (2)

- Characteristic properties of transactions:
  - Atomic: To the outside world, the transaction happens indivisibly.
  - Consistent: The transaction does not violate system invariants.
  - Isolated: Concurrent transactions do not interfere with each other.
  - Durable: Once a transaction commits, the changes are permanent.

Transaction Processing Systems (3)

- The role of a TP monitor in distributed systems.



51

Pervasive Computing

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it

Mark Weiser
The Computer for the 21st Century

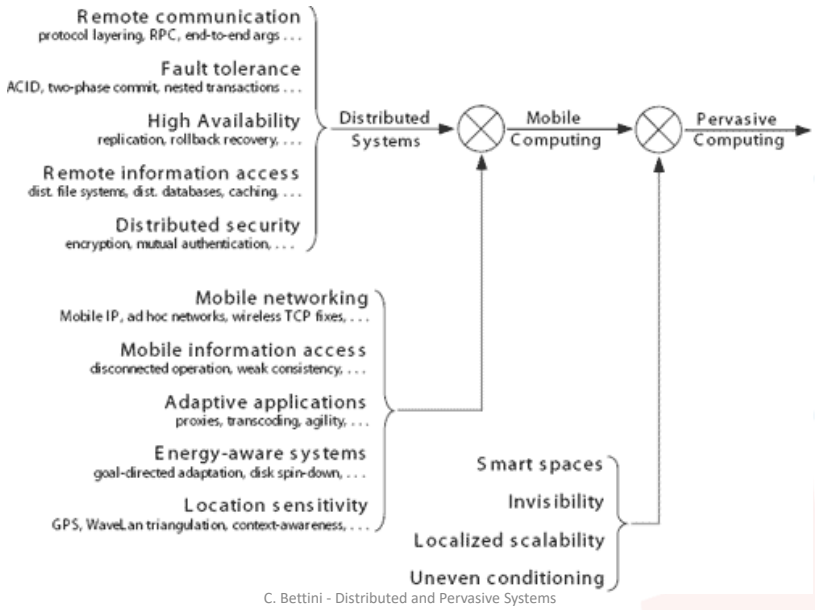
53

Distributed Pervasive Systems

- A distributed system with the following main features
  - it includes unconventional nodes
    - (possibly mobile) objects with computing and communication capabilities (smartphones, smart appliances, smart meters, sensor networks, …)
  - adaptivity
    - the system logic considers the current context and adapts the system behavior for optimizing the system goal

54

Distributed Pervasive Systems



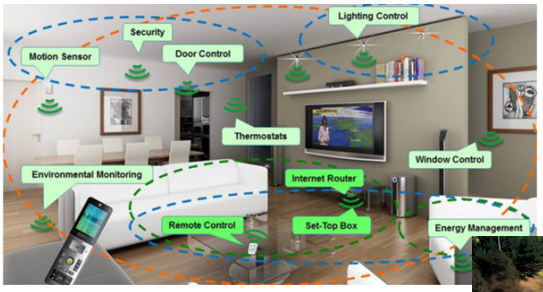
55

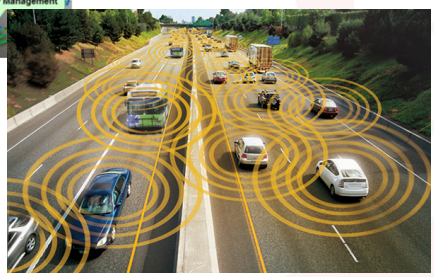
Distributed Pervasive Systems

- Examples:
  - Smart Environment (Home, Building, City) Systems
    - Smart Home services
    - Smart energy management
    - Smart transportation
      - e.g., using *crowdsensing* through smartphones and sensors
  - e-Health systems for
    - Tele-healthcare
    - independent living and ageing well
    - accessibility

Every Ware Lab

Example: Smart Spaces





Every Ware Lab

E-Health Care Systems

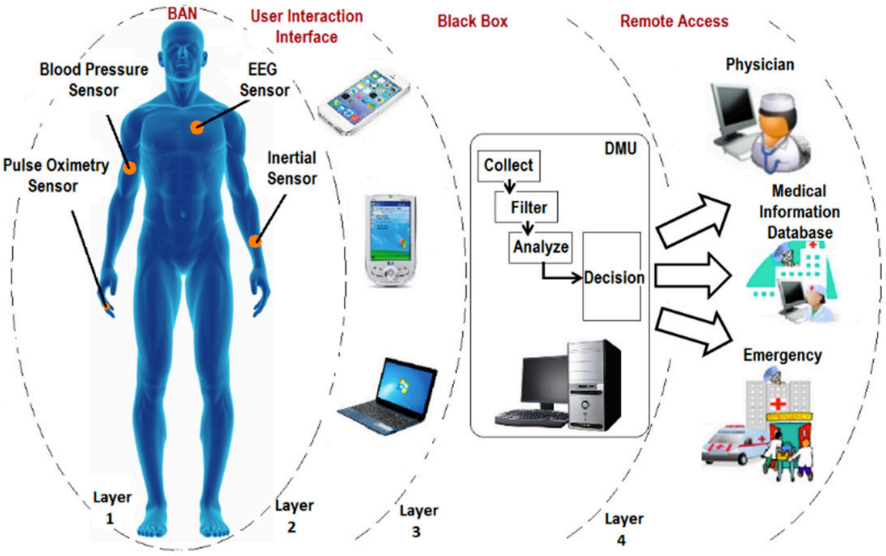


Figure from: Ghamari, M. et al.A Survey on Wireless Body Area Networks for eHealthcare Systems in Residential Environments. *Sensors* **2016**, *16*, 831.

58

References

- Google Borg cluster management system (2016 video presentation https://www.youtube.com/watch?v=0W49z8hVn0k)
- Kubernetes.io
- NIST definition of Cloud (http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf)
- Amazon EC2 (http://aws.amazon.com/ec2/)
- Openstack (https://www.openstack.org/)
- Mark Weiser paper on Pervasive Computing (https://www.ics.uci.edu/~corps/phaseii/Weiser-Computer21stCentury-SciAm.pdf)

65