Distributed Systems

a.y. 2023/2024

Prof. Carlo Ferrari
Dott. Michele Stecca
Introduction to Distributed Systems
...roughly speaking...the “first cloud view”!
Experience and…

- Networked (*embedded*) computers, networked computing

*a common reality*

what are the main features of polymorphic computing infrastructures?
...and expectations.

- New and (potential) future applications
  challenges for actual systems

How do the design process evolve?
How do the new techniques come into use?
How about costs and Quality of Service?
A technical discipline...

- Learn about actual platform and software and their use
- Testing and validation issues
- Deploy and manage either actual systems or their components
...an academic discipline.

- Analyze the fundamental problems and the current methodologies for their solution.
- Set up general model about common and general issues.
- Foresee new research directions and new solutions for open questions.
A (obvious) motivation...

Resource Sharing

...machine cycles, memory, storage, peripherals, programs, data, ...

<table>
<thead>
<tr>
<th>improve</th>
<th>reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>capabilities,</td>
<td>costs</td>
</tr>
<tr>
<td>availability</td>
<td>time</td>
</tr>
<tr>
<td>performance</td>
<td>faults</td>
</tr>
</tbody>
</table>
...in spite of:

- Multi-scale spatial distribution
  - local, regional, geographical (worldwide)
- Non deterministic temporal distribution
  - mobility, energy consumption, lack of connection
- different administrative domains
- access rules, security policies
Technological framework…

- Advanced optical networks
- Wireless connections
- Fast microprocessors
- Parallel architectures
- Communication protocols
- Security mechanisms
Networked computer can be spatially separated by any distance.
...components in a DS:

- Functional components
  Represent system activities

- Interconnection components
  Enable communications
...functional elements:

- System components are the basic objects that interact/cooperate/are securely shared

- Functional elements offer services both inside and outside the system

- From hardware to its software abstraction
interconnection tools:

- Communications are the basic tool for getting interaction/cooperation/secure sharing among system components.

- The communication media (network) enables system functionalities.

- Its performances affects the overall system performances.
... model pitfalls ...

- The network is reliable.
  ... messages always arrive at their destination.

- The network is secure.
  ... messages cannot be corrupted or altered or simply eavesdropped

- The network is homogeneous.
  ... Network elements have the same features
...model pitfalls...

- The topology does not change
  ...the communication layout never changes

- Latency is zero.
  ...no delay in transmission time

- Bandwidth is infinite
  ...no limit in the number of concurrent messages along a single channel
...model pitfalls...

- Transport cost is zero
  ... Information travel at no cost

- There is one administrator.
  ... same rules and policies in the system
...distributed vs centralized...

- Concurrency
  ...at a different level of abstraction

- Lack of a global clock
  ...time is measured locally

- Independent failures
  ...components faults may not affect the system
A Distributed System is a collection of autonomous computers interconnected by a computer network and equipped with distributed system software to form an integrated computing facility.
A Distributed System is a collection of autonomous computers interconnected by a computer network and equipped with distributed system software to form an integrated computing facility.
A Distributed System is a collection of independent computers that appears to its users as a single coherent system.
A **Distributed System** is a collection of independent computers that appears to its users as a **single coherent system**.
A **Distributed System** is a collection of independent computers interconnected via a network, that are capable of collaborating on a task.
A Distributed System is a collection of independent computers interconnected via a network, that are capable of collaborating on a task.
A Distributed System is a collection of individual computing devices that can communicate with each other.
...an infrastructure oriented definition...

A Distributed System is a collection of individual computing devices that can communicate each other
…an endless debate…

- Distributed Computing

- Parallel Computing
…actors…

- End users
- System Administrators
- Application developers

But also:

- Tools developers
…the basic element(s)…

- An autonomous computer (node) with its own architecture, operating system, administrator.

- An interconnection medium with its own protocols, security and privacy mechanisms
...a view from below...

Processes (threads):

- execute concurrently
- interact in order to co-operate to achieve a common goal
- exchange information by means of messages transferred over a communication network
End-user usually do not know any details but those related to the direct interaction with the application...Human-Machine Interface

At the same time they evaluate the overall system performances
Applications are structured by proper assembling of components ... software components!

Architectural issues are managed by developers
...system components...

- component specification
- sintax and semantics
- interface and ...ontology?
...an intranet...
...intranet...

- Made up from LANs linked by backbones
- It enables information flow within organisation (electronic data, documents, ...)
- It provides services like email, file system, print servers...
- It is connected to Internet via router
- in/out communications are protected by firewall
internet...

- It connects intranets (via backbones) ...
- ...and home users (via modems, ISPs)
- It enables email, file transfer, multimedia communications, WWW, peer sharing
- very large and heterogeneous
- open-ended
...the Web...
...the Web...

- It shows a client-server approach
- Resource are denoted in a uniform way
- Publishing and accessing resource and services world-wide across the Internet

- It is an open system (it can be extended, re-implemented, ...)

Portable and handheld devices
Portable and handheld devices...

- Wireless LANs (WLANs) connectivity for portable devices (laptops, PDAs, mobile phones, video/digital cameras, ...)

- Home intranet ... devices are embedded in home appliances (hi-fi, washing machines...)
- ‘remote control’ + communication
...a general organization...

A system organization through middleware.
Cluster Computing Systems

Figure 1 - An example of a cluster computing system.
Grid Computing Systems

- Applications
- Collective layer
- Connectivity layer
- Resource layer
- Fabric layer
--- Transaction Processing Systems ---

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN_TRANSACTION</td>
<td>Mark the start of a transaction</td>
</tr>
<tr>
<td>END_TRANSACTION</td>
<td>Terminate the transaction and try to commit</td>
</tr>
<tr>
<td>ABORT_TRANSACTION</td>
<td>Kill the transaction and restore the old values</td>
</tr>
<tr>
<td>READ</td>
<td>Read data from a file, a table, or otherwise</td>
</tr>
<tr>
<td>WRITE</td>
<td>Write data to a file, a table, or otherwise</td>
</tr>
</tbody>
</table>
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...

- Nested transaction
  - Subtransaction
    - Airline database
  - Subtransaction
    - Hotel database

Two different (independent) databases
Transaction Processing Systems...
Enterprise Application Integration

Middleware as a communication facilitator in enterprise application integration.

- Middleware as a communication facilitator in enterprise application integration.
…requirements for pervasive systems…

- Embrace contextual changes.
- Encourage ad hoc composition.
- Recognize sharing as the default.
Where and how should monitored data be stored?

How can we prevent loss of crucial data?

What infrastructure is needed to generate and propagate alerts?

How can physicians provide online feedback?

How can extreme robustness of the monitoring system be realized?

What are the security issues and how can the proper policies be enforced?
Monitoring using a local hub (a) or a continuous wireless connection (b)
…Sensor Networks…

- How do we (dynamically) set up an efficient tree in a sensor network?
- How does aggregation of results take place? Can it be controlled?
- What happens when network links fail?
storing and processing data at the operator’s site
…Sensor Networks…

- storing and processing data at the sensors.
Heterogeneity

That is, **variety** and **difference**

- networks,
- hardware,
- operating systems,
- programming languages,
- implementation
managing heterogeneity...

- standards (protocols, middleware)
  - Middleware: a programming abstraction that masks the heterogeneity (CORBA, Java RMI)

- mobile code support
  - virtual machine (JVM)
…Openness…

- How the system can be extended/re-implemented?

- Services comply with public standard rules

- How new resource-sharing services can be added and made available?
...Openness...

- The challenge: systems consisting of many components engineered by different people
- Interoperability
- Portability
…Openness…

- Key software interface must be available by software developers…IDL (Interface Definition Language)

- Standards are “slow moving”
Open Distributed Systems…

- independent from vendors
- publishable key interfaces
- publishable communication mechanisms
…Openness…

- Separation between mechanisms and policies
Confidentiality: protect against disclosure to unauthorized individuals (medical records)

Integrity: protect against alteration and corruption (financial data)

Availability: protect against interference
…Security…

- Eavesdropping
- Phishing
- Denial of Service
- Security of mobile code
- …and more …
…security…

✓ encryption

✓ knowledge of identity

✓ Is it enough?
…Scalability…

- will a DS remain effective with growth?
- need to control cost of resources, performance loss
- Scaling with respect to size
- Scaling with respect to location
- Scaling with respect to administrative domains
...scalability limitations...

<table>
<thead>
<tr>
<th>Concept</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized services</td>
<td>A single server for all users</td>
</tr>
<tr>
<td>Centralized data</td>
<td>A single on-line telephone book</td>
</tr>
<tr>
<td>Centralized algorithms</td>
<td>Doing routing based on complete information</td>
</tr>
</tbody>
</table>
…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 with respect to size
- Scaling with respect to location
- Scaling with respect to administrative domains
Scalability

• control the cost of physical resources

\[ O(n) \]

• control the overall performance

\[ O(\log n) \]
• preventing the “running out” IP addresses
• avoiding bottlenecks DNS
...Scaling Techniques...

(a) Client: First Name: MAARTEN, Last Name: VAN STEEN, E-mail: STEEN@CS.VU.NL

(b) Client: First Name: MAARTEN, Last Name: VAN STEEN, E-mail: STEEN@CS.VU.NL

Check form -> Process form

Server:
dividing the DNS name space into zones.
...managing scalability...

- Replication

- Caching

...but consistency problems could arise!
- Synchronous communications
- Asynchronous communications
Ability to continue computation in the presence of failures.

- detect failures
- mask failures
- repair failures
- tolerate failures

Failure is a kind of “standard situation”
…Failure handling…

- recovery from failures
- redundancy

Measure of availability i.e. QoS
Processes execute simultaneously and share resources.

- process/object synchronisation
- inter-process communication
Transparency

Concealment from the user and the application programmer of the separation of components in a Distributed Systems, so that the system is perceived as a whole rather than as a collection of independent component
…Transparency…

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

Location transparency

Concurrent transparency
- Replication transparency
- Failure transparency
- Mobility transparency
- Performance transparency
- Scaling transparency
Challenges

- Heterogeneity ... Middleware
- Openness ... Key software interfaces
- Security ... encryption and knowledge of identity
- Scalability
• Failure handling ... Recovery

• Concurrency ... synchronisation and communication

• Transparency ... A single system...?
end of lectures

References: