

Servicios y Aplicaciones Distribuidas

2022/2023

Presentation and Overview

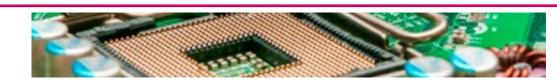




Why this course

- Most current day software
 - Is distributed in nature (example: what we are doing right now!)
 - Many different components
 - Client interacting with user
 - One or several servers communicating with those clients
 - » Often servers also communicate among themselves
 - Do so providing a "service"
 - Users are saved the trouble of managing software
 - Use the Cloud approach for agile deployment
 - Different models for deployment and management
 - Need to also consider other approaches
 - P2P
 - A particularly fruitful application line is Blockchain







Overview: Goals of the course

- Four main goals
 - Discuss the importance of architecture in software
 - Present the concept of Software Service as a fundamental structuring principle
 - Discuss the importance of distributed systems and services in today's software world
 - Present the Principles under which services are provided and architected
 - Present the main problems and technologies available
 - Learn through practice:
 - Give the student the chance to confront the implementation of a service using current-day technologies to overcome the difficulties.







- Main focus areas
 - Fundamental concepts
 - Programming Coordination in Distributed Systems
 - Architectural approaches
 - Client/Server
 - Messaging
 - P2P
 - Microservices
 - Cloud Computing & SaaS
 - Practical Considerations
 - Technologies
- Hands-on project



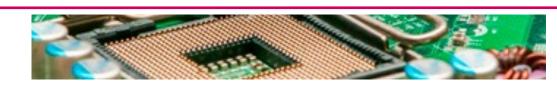




Fundamental Concepts

- Apps vs Services
- Distributed Systems
- State
- Coordination and consistency
- Implementing services (SaaS)
 - Main difficulties
 - PaaS example







Programming coordination

- Shared state concurrency control (refresher)
- CSP and Hoare's coordination model
- Asynchronous programming
 - Promises
 - Streams/Event sourcing
 - Message Queues



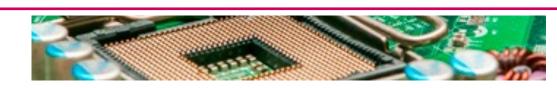




Architectural approaches

- Goals
 - Scalability
 - Fault tolerance
 - Maintainability
- Client/Server and centralized systems
- Messaging and Reactive/Async systems
- P₂P
 - P2P in action: Blockchain



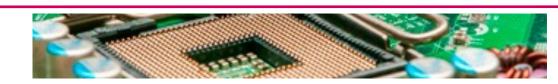


Cloud Computing



- Cloud Computing
 - Enabling technologies
 - Everything virtual
 - Trends in software business models
 - Main models
 - laaS
 - PaaS
 - SaaS
 - Multi-tenancy
 - Deviations



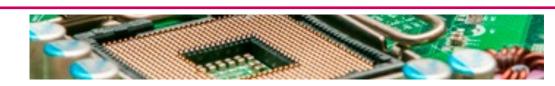


THE POLY

Practical considerations

- Quality of Service
 - SLA and QoS guaranties
 - Reliability and Scalability
- Consistency
 - The CAP theorem
- Security
 - Authentication & Authorization
- Service deployment and management

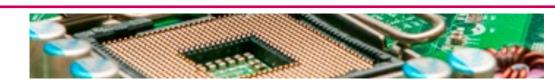






- Event-driven programming
 - Node.js
 - Promises & observables
- ZeroMQ
 - State-less message queues
- Message queues
 - Kafka
- Light virtualization
 - Containers
 - Docker
- PaaS example: KPaaS over Kubernetes





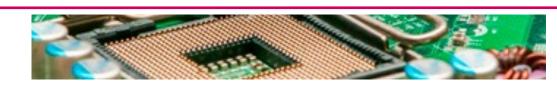


- SaaS
 - Multi-component Service
 - Implemented on top of a PaaS (attempt)
 - Provided to you
 - Using Docker images/containers
 - Will need to create account on docker hub





- Two tests
 - o 50% of the grade (25% Theory, 25% Seminars and Labs)
- Portfolio 25%
 - Work proposed during seminars
- Project 25%







- On-campus
 - Seminars
 - Theory
 - Labs







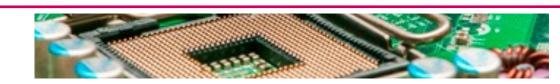
- Linux Advised
- OSX and Windows 10 can work
- Software:
 - Docker
 - Docker desktop is advisable
 - Use specific docker images to develop
- Deployments on specific PaaS made available from Master's cooperating entity
 - In the form of a PaaS
 - Need to build docker images
- Gitlab UPV
 - GIT repos within the SAD/2022-2023 namespace.
 - One per team
 - Image registry







- José Bernabeu
 - o bernabeu@upv.es
 - TA1-MUI21 (Theory)
 - L1-SAD (Labs)
- Juan Salvador Sendra
 - o jsendra@dsic.upv.es
 - TS1-SAD (Seminars)
- Tutorships on Demand (e-mail) and remote





- The NIST Definition of Cloud computing (Peter Mell, Timothy Grance)
- Sistemas distribuidos : principios y paradigmas (Andrew S. Tanenbaum)
- The architecture of open source applications: elegance, evolution, and a few fearless hacks (Amy Brown , Greg Wilson)
- Understanding fault-tolerant distributed systems (Cristian, Flaviu)
- Web services : concepts, architectures and applications (Gustavo Alonso et al)
- RESTful web services (Leonard Richardson, Sam Ruby)
- https://docs.kumori.systems (PaaS documentation)
- <u>https://docker.com</u> (more information about docker)
- Bitcoin and Cryptocurrency Technologies (Arvind Narayanan et al...),
 https://d28rh4a8wqoiu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf?a=1



