Dependable Distributed Systems

Confiabilidade de Sistemas Distribuídos

DI-FCT-UNL, Nuno Preguiça

Course Overview

2018/2019, 2nd SEM

MIEI

Mestrado Integrado em Engenharia Informática

Lectures and Labs

- Nuno Preguiça
  - P3/7 ext ????, asc.di.fct.un.pt/~nmp

Course Ref. (#11555)

CLIP: Course information

- Objectives, Program, Requirements
- Bibliography
- Materials: Slides/Lectures and Suggested Readings
- Documentation
- Evaluation (Methods, Criteria) and Results
- … Events, Messages, Notifications

  ….See also your Email (Reg. CLIP)

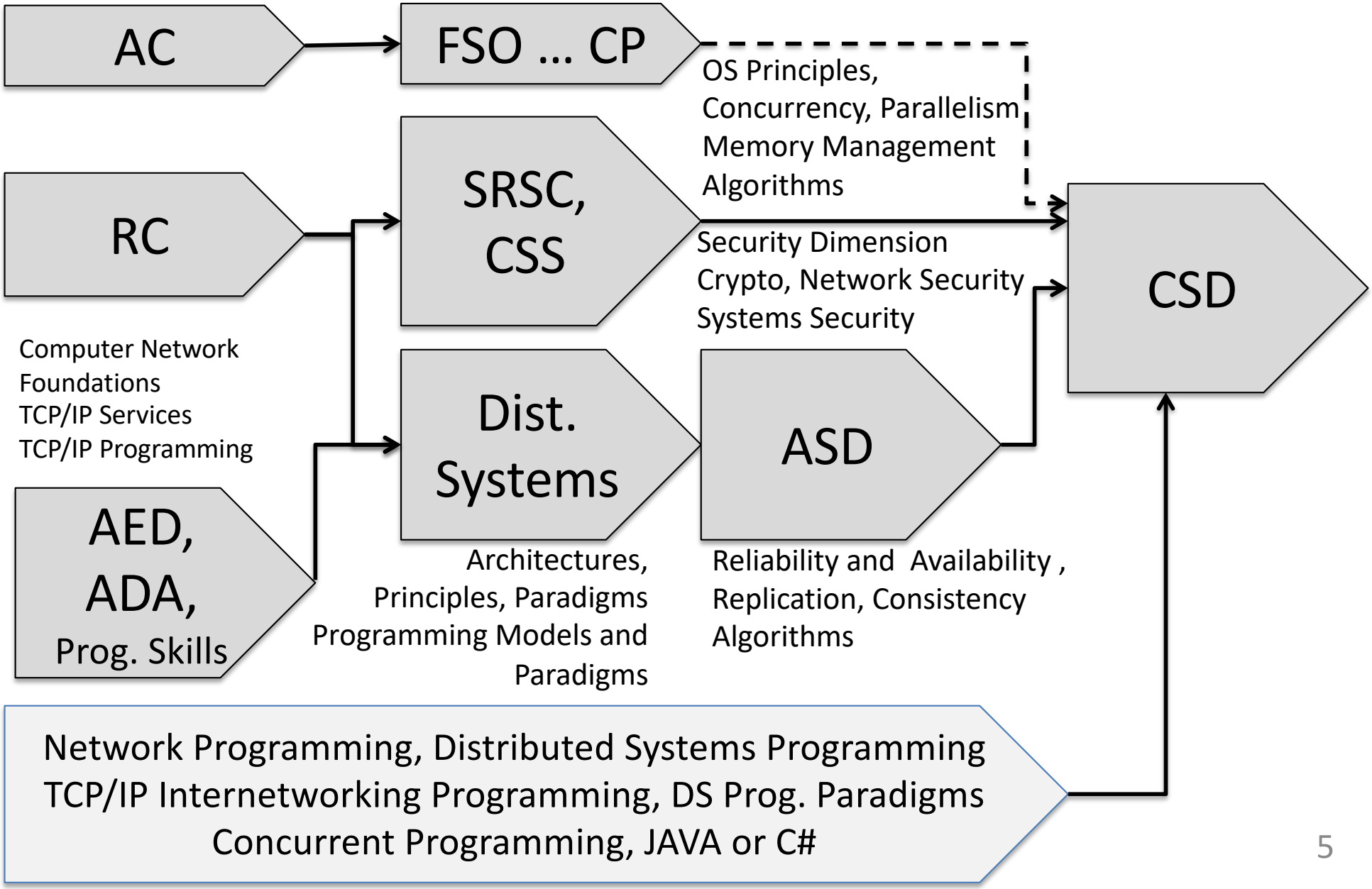
Course Objectives (see CLIP)

- Have solid understanding of the basic concepts and theory of secure and dependable distributed computing.
- Getting familiar with some basic building blocks (tools and APIs) and techniques, needed to build secure and dependable systems.
- No attempt to be comprehensive:
  - topics covered are inspired by reference authors and what we conduct as research interests in NOVA LINCS
  - … and in “Looking at my crystal ball, what will be important in the near future”

Program Topics

- Introduction – Dependable Systems: Concepts, Properties and Involved Technologies
- Mechanisms for Fault-tolerance
  - State-machine replication, quorum based replication
- Mechanisms for Intrusion Tolerance and Fault-Tolerance
  - BFT
- Decentralized mechanisms for fault-tolerance
  - Blockchains, Smart contracts
- Security in Database Systems and Advanced Cryptographic Methods
  - Erasure-codes, homomorphic Encryption Methods
- Security with trusted computing hardware
  - TPMs, SGX
- Secure networking
- Availability and Operation Continuity: DoS/DDoS Protection

Requirements / Course Plan Sequence



Practical Requirements

- Practice, Autonomy in:
  - Java Programming & debugging, IDEs
    - Internetworking / Dist. Programming with Java: Sockets, REST, SOAP
    - Java Security Prog. (Crypto, JCE, Crypto Primitives)
  - Unix-based development environment (Linux. Mac-OS)
    - Management / Sys Admin
  - Virtualization with Docker (“containerized SW”, TPs)
    - https://www.docker.com

Lectures/Labs

- Lectures
  - 1 Lecture, 2 h / week
- Labs / Work-Assignment / Evaluation Projects
  - 1 Lab Slot, 2 h / week
  - Focus:
    - Introducing the technologies used for developing the projects
    - Requirements and Design Criteria (Discussion)
    - Design and Implementation (Refinement, Discussion)
    - Experimental Evaluation and Demonstrations
      - Presentations, Demos, Discussion

Schedule

2ª F	3ª F	4ª F	5ª F	6ª F
				9h-11h Lect. 112 Ed II
				11h-13h Lab Slot 112-II

Sometimes
we will need to reschedule …

- Backup Slot
- Changing Lect <> Labs ...
- Rescheduling classes in some weeks
- Notification in advance ...

Plan / Topics

	Lectures	**Labs**
08-Mar	Introduction & Paxos	
15-Mar	Quorums, ABD & Randomized algorithms	
22-Mar	Byzantine fault tolerance	
29-Mar	BlockChain and Bitcoin	
05-Apr	Smart contracts	1st proj: 8/4
12-Apr	Database Security	
19-Apr	Easter	
26-Apr	Erasure coding + cloud of clouds	
03-May	Practical Partial Homomorphic Encryption Applicatinos	Test 1 : 2/5
10-May	Trusted Computing with TPMs	
17-May	Trusted Computing Environments (TEEs) and Applications	
24-May	IPSec, Tunneling and Secure VPNs	2nd proj: 24/5
31-May	Secure networking	
07-Jun		Test 2: 6/5

Bibliography / References

- Bibliography (See CLIP)
- Topics vs. Book Chapters and Selected Papers for Readings
  - References in each Lecture

- Slides for the course are based on a previous version produced by Henrique João Domingos.

Assessment (1)

- **2 Midterm Tests: T1 , T2**
  - Covering Theoretical Topics / Lectures
    - Book Chapters and Suggested Readings
  - Typically (ref): 1h30-2h00

- 2 Frequency Elements: Work-Assignments
  - F1, F2

Assessment (2)

- **Frequency Elements TP1, TP2**
  - **65%**: Implementation
  - **20%** Report/Writing (w/ structure based on a given template)
  - **15%** Practical Evaluation
    - Demo or question included in test.

Assessment and Grade (Summary)

- **Frequency**
  - TP1 min >= 8/20
  - TP2 min >= 8/20
  - Average TP1,TP2 >=8/20
- **FS - Final Score:**
  - T1 (20%)
  - T2 (30%)
  - TP1 (17,5%)
  - TP2 (32,5 %)
- **Appeal Exam:** 50%
  - (equiv. to T1, T2)

- **Grade, if**
  - FS >=9,5/20
  - Average T1,T2 >=8,5/20
  - Average TP1,TP2 >=8/20
- **Appeal: if**
  - FS < 9,5/20
  - TP1 min >= 8/20 (2/5)
  - TP2 min >= 8/20 (2/5)

Dates

- T1: 2/5
- TP1: Deliv. until 8/Apr (*)
- T2: 6/6
- TP2: Deliv. until 24/May (*)

---

*) Deliverables with instructions for delivering defined in the Project Requirements.

Course Motivation

Relevance of Secure and Dependable Computing

- Why Secure and Dependable Computing is important?*
  - Critical Systems
  - Increased reliance on software to optimize everything from business processes to engine fuel economy
  - Relentlessly growing scale and complexity of systems and systems-of-systems
  - Near-universal reliance on a commodity technology base that is not specifically designed for dependability
  - Growing stress on legacy architectures (both hardware and software) due to ever-increasing performance demands
  - Worldwide interconnectivity of systems and the integration of current relevant technology: Clouds, Clouds-of-Clouds, Mobility, IoT
  - Continual threats of malicious attacks on critical systems

(*) Taken from a typical research project: "A high dependability computing consortium", James H. Morris, CMU, http://www.cs.cmu.edu/%7Ejhm/hdcc.htm

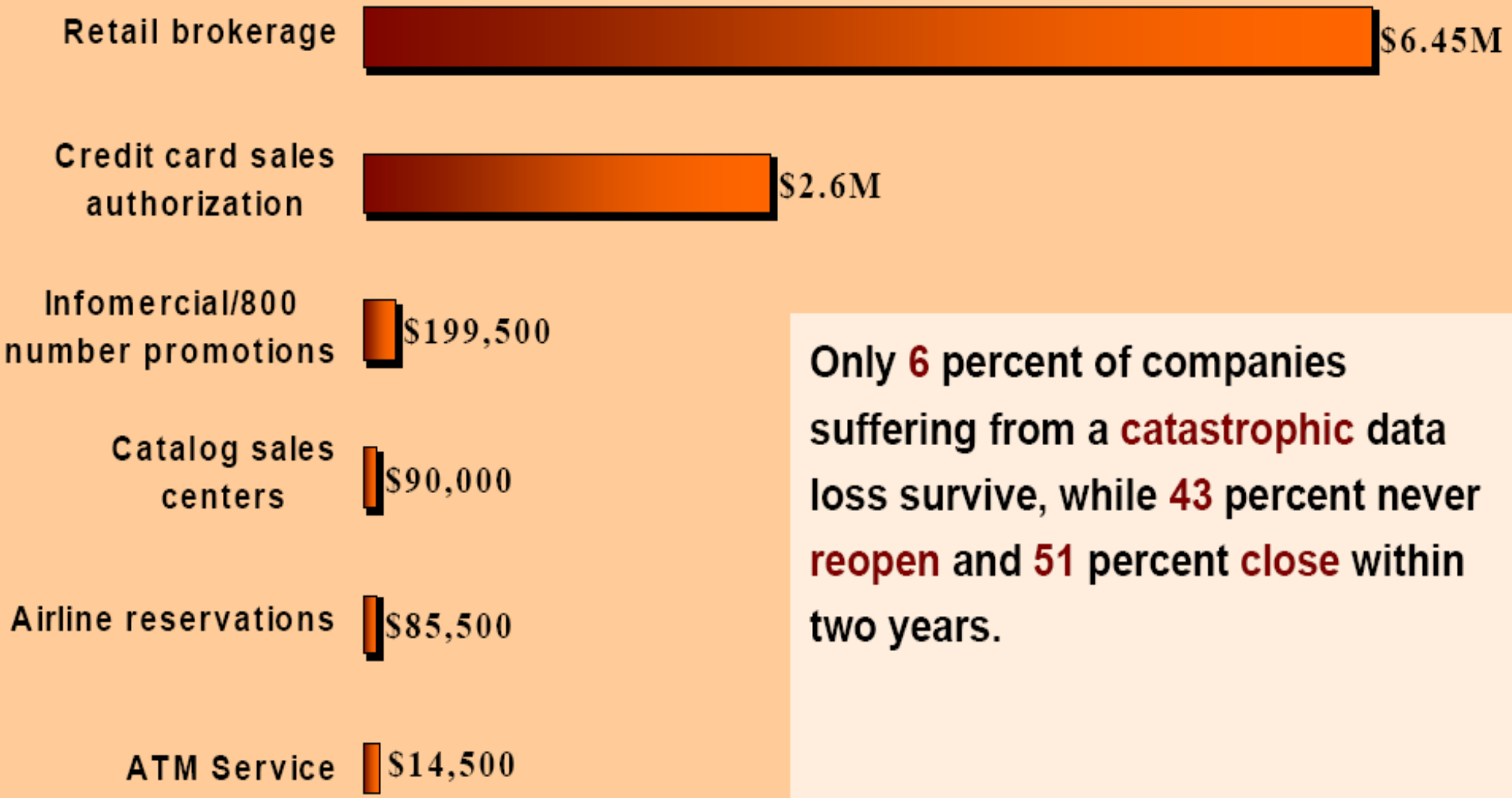
More motivation …

- The cost of poor software is very high
  - Annual cost to US economy of poor quality software: $60B in 2002
  - *source: US NIST Report 7007.011, May 2002.*
- Industry (and new critical systems and applications) needs greater dependability and security
  - Improved quality of products
  - Improved quality of development processes
  - Better system and network security, to avoid:
    - viruses, trojans, denial of service, ...
    - network penetration, loss of privacy/confidential data, ...
  - Improved customer satisfaction
  - Regulation and compliance issues in more exigent sectors
  - The high cost of downtime

The Cost of Downtime

The average financial impact per hour of interrupted computer operations (by industry)

(1996 Cost of Downtime Study – by Contingency Planning Research)



Only 6 percent of companies suffering from a catastrophic data loss survive, while 43 percent never reopen and 51 percent close within two years.

SOURCE: UNIVERSITY OF TEXAS

Cost of Downtime …

- From "Assessing the Financial Impact of Downtime" by Vision Solutions, Inc. 2008 (http://www.strategiccompanies.com/pdfs/Assessing%20the%20Financial%20Impact%20of%20Downtime.pdf)
- Typical hourly cost of downtime by Industry
  - Brokerage Service: $6,480,000
  - Energy: $2,800,000
  - Telecom: $2,000,000
  - Manufacturing: $1,600,000
  - Retail: $1,100,000
  - Healthcare: $636,000
  - Media: $90,000

The enormous cost of system failures …
(very underestimated today ?)

- Average costs for downtime in data-centers *[1,3]:
  - 42.000 to 300.000 USD $ per hour
  - Wasted expenses + Loss of Revenues
  - Damages in reputation and loyalty of potential customers

(*) Examples:

[1] A. Arnold, Assessing the finantial impact of downtime, Apr 2010, www.businesscomputingworld.co.uk/assessing-the-financial-impactof-downtime/

[3] ChannelInsider, Unplanned it outages cost more tha $5000 per minute, Technical Report, www.channelinsider.com/c/a/Spotlight/Unplanned-ITOutages-Cost-More-than-5000-per-Minute-Report-105393/, May 2011

Industry is more and more embracing Secure and Dependable Computing

- New HW and SW Platforms: Trusted hardware, Smartcards, Pervasive Computing and autonomic-computing
- The case for Healthcare Management Systems, HMRs; Finance: Fraud, AML; Citizenship Systems (Identity Thefts)…
- New Apps (Mobility and IoT): how to avoid an Internet of unsecure or unreliable things ? (ex., IMDs, Privacy-Preservation...)
- Cloud-Based App and Services w/ Dependability Requirements

Industry is more and more embracing Secure and Dependable Computing

- Major Personal Computing dependability and security initiatives (regulatons, standards) under way:
  - Trusted Computing Group
    - Ex., : Intel, HP, IBM, Microsoft
    - Intel SGX, Arm TrustZone initiatives (among other efforts)

The best dependability solutions: optimization (best balance) of tradeoffs

Reliability vs. Availability vs. Security and Privacy

… AND…

- Performance
- Openness
- Scalability
- Transparency criteria (Distributed Systems)
  - Access
  - Location
  - Migration vs. Relocation
  - Replication
  - Concurrency (sharing)
  - Failures

Can we have designed solutions addressing properly (and balancing) these different criteria together ?