Distributed Computing

MAP-I

October 8, 2007

Overview

Distributed computing refers to algorithms running on a set of machines connected by a network. Its importance has increased as computation migrated from monolithic mainframes to decentralized structures connected by the internet.
Examples of distributed systems appear in many areas such as telecommunication (Skype), web applications (Amazon, Google), distributed data processing and massively multiplayer games.

Pitfalls in Distributed Systems

Concurrency, which occurs naturally in a distributed system, is already a difficult subject. On top of that, the problems that arise in asynchronous distributed systems subject to processor or link failures are difficult to comprehend.

Pitfalls in Distributed Systems

Concurrency, which occurs naturally in a distributed system, is already a difficult subject. On top of that, the problems that arise in asynchronous distributed systems subject to processor or link failures are difficult to comprehend.
This means that people may waste years trying to solve an impossible problem; or they may build a platform that will malfunction, rarely and impredictably, in a non-repeatable and incomprehensible way.

Pitfalls in Distributed Systems

Concurrency, which occurs naturally in a distributed system, is already a difficult subject. On top of that, the problems that arise in asynchronous distributed systems subject to processor or link failures are difficult to comprehend.
This means that people may waste years trying to solve an impossible problem; or they may build a platform that will malfunction, rarely and impredictably, in a non-repeatable and incomprehensible way.
The best approach is to start from a good theoretical background

Approach

This course aims at providing the theoretical foundations of distributed systems. It is targeted to graduate students and researchers wishing to advance the state-of-the-art in distributed systems. The course is technology agnostic and the abstractions presented are independent of any given technology.

Course

The course focuses on formal models (e.g. I/O automata), abstractions (e.g. logical time), problems (e.g. agreement) and algorithms to solve them. It also focuses on impossibility results (e.g. the impossibility of fault-tolerant consensus in asynchronous networks).
Graduate-level courses similar to this one include the following:

- "Distributed Algorithms" at the MIT, by Nancy Lynch.
- "Theory of Distributed Computing" at the EPFL, by Rashid Guerraoui.

Any job perspectives after the joys of research?

February 02, 2005, Job Opening description by Amazon CTO and renown researcher Werner Vogels:

What kind of things am I looking for in you?

- *You know your distributed systems theory: You know about logical time, snapshots, stability, message ordering, but also acid and multi-level transactions. You have heard about the FLP impossibility argument. You know why failure detectors can solve it. You have at least once tried to understand Paxos by reading the original paper.*
- *You have a good sense for distributed systems practice: You can reason about churn and locality in DHTs. You intuitively know when to apply ordered communication and when to use transactions. You can reason about data consistency in a system where hundreds of nodes are geographically distributed. You like the elegance of systems based on epidemic techniques.*
- *. . .*

Learning Outcomes

- model formally a distributed system
- differentiate between synchronous and asynchronous models
- understand the assumptions and limitations underlying models of distributed systems
- reason about distributed algorithms
- design new distributed algorithms
- invoke impossibility results to avoid wasting time trying to solve an unsolvable problem
- prove impossibility results

Teaching Method

The course will be organized mainly around formal lectures. Some lecture time (around 1/4) will be used for recitation, where a given student will have to present and defend a previously assigned research paper, leading to a discussion involving the other students.

Faculty

The teaching team consists of members of the Distributed Systems Group (GSD) of the Informatics Department of Universidade do Minho. The team has considerable experience of teaching and research in distributed systems. Faculty includes professors Francisco Moura, Rui Oliveira, Paulo Almeida, Carlos Baquero and José Orlando Pereira.