# Distributed Systems

## Perfil de Sistemas Distribuídos

Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

MIEI 2018/2019

Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

## Everything is distributed

*"Distributed systems once were the territory of computer science Ph.D.s and software architects tucked off in a corner somewhere. That's no longer the case."*
*(2014 http://radar.oreilly.com/2014/05/everything-is-distributed)*

## Everything is distributed

*"Distributed systems once were the territory of computer science Ph.D.s and software architects tucked off in a corner somewhere. That's no longer the case."*
*(2014 http://radar.oreilly.com/2014/05/everything-is-distributed)*

This talk shows:

- Distributed Systems teaser problems
- Unit outline

# Two (gangster) Generals Paradox



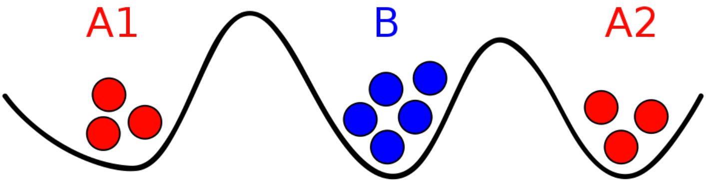
> A group of gangsters are about to pull off a big job. The plan of action is prepared down to the last detail: Some of the men are holed up in a warehouse across town, awaiting precise instructions. It is absolutely essential that the two groups act with complete reliance on each other in executing the plan.

in "Some Constraints and Trade-offs in the Design of Network Communications". Akkoyunlu, Ekanadham and Huber. 1975.

# Two (gangster) Generals Paradox

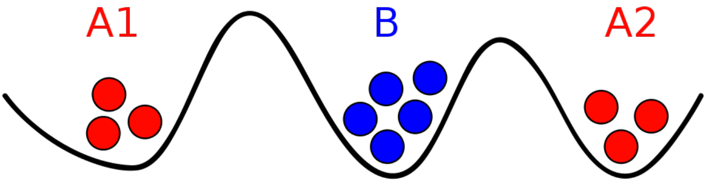


Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero




- Red gangsters have more mobsters if together, but need to attack at the same time
- Messengers are unreliable
- How to coordinate an attack?

# Two (gangster) Generals Paradox



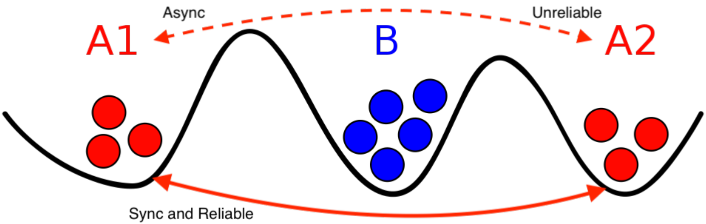

- Red gangsters have more mobsters if together, but need to attack at the same time
- Messengers are unreliable
- How to coordinate an attack?

http://en.wikipedia.org/wiki/Two_Generals'_Problem

# Two (gangster) Generals Paradox



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero




- The battleground is shared unreliable medium
- Build a private reliable tunnel
- Problem (mostly) solved

# Synchronous model



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

Expected in small scale and cost dominated. Ex: Cars subsystems

- Processing delays have a known bound
- Message delivery delays have a known bound
- Rate of drift of local clocks has a known bound
- Difference between local clocks has a known bound

# Asynchronous model



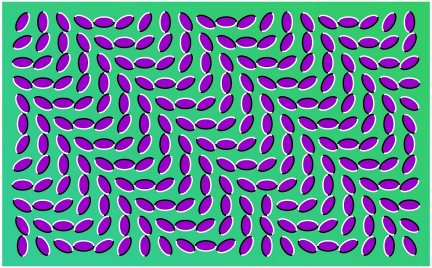
Adapted to large scale and can spread costs. Ex: Internet

- Processing delays are unbounded or unknown
- Message delivery delays are unbounded or unknown
- Rate of drift of local clocks is unbounded or unknown
- Difference between local clocks is unbounded or unknown

# Counting nodes

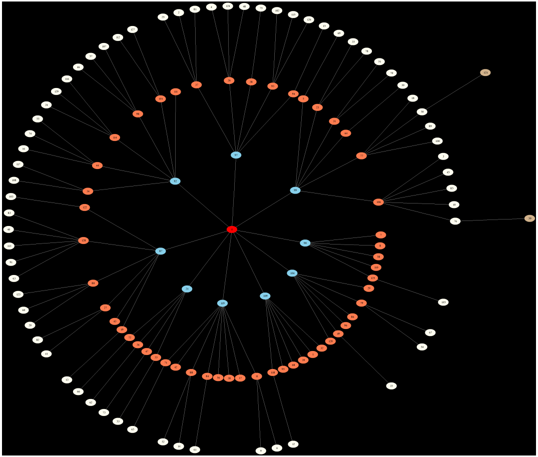


- In concurrent programming counting motivated synchronization
- How to count nodes in a large asynchronous distributed system?



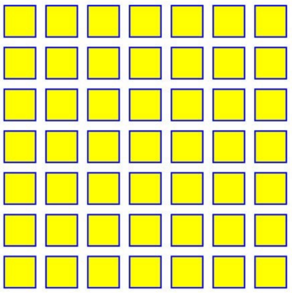
# Counting nodes
## Solution 1 - Build a Tree

Build a distributed Breadth First Search tree from a root node



# Counting nodes
## Solution 2 - Distributed Averaging

Average out 1 (litre of water) from a special node



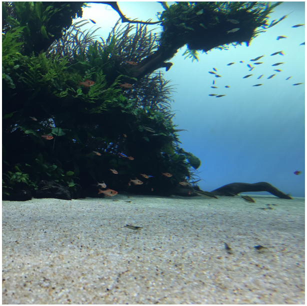
On $7 * 7$ nodes each will converge to $v_i = 0.020408163265306\ldots$

Local estimate is $1/v_i \approx 49$

Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

Randomly collect $k$ fishes, tag and release them, wait, recapture

Smaller populations lead to more tagged recaptures



Distributed random sample is key on some new blockchain protocols

# Consistency Models



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

### Eventually Consistent. CACM 2009, Werner Vogels

- In an ideal world there would be only one consistency model: when an update is made all observers would see that update.
- Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.

### CAP theorem. PODC 2000, Eric Brewer

Of three properties of shared-data systems – data consistency, system availability, and tolerance to network partition – only two can be achieved at any given time.

Both strong and eventual consistemcy are covered in the course

Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero


- Paradigma cliente-servidor
- Serialização de dados
- Programação baseada em atores
- Programação orientada às mensagens
- Programação orientada aos recursos

# Unidades Curriculares - SD
## Fundamentos de Sistemas Distribuídos



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

- Programação baseada em eventos
- Modelação e correção de sistemas distribuídos
- Tempo lógico e observação global
- Transações distribuídas

# Unidades Curriculares - SD

## Tolerância a Faltas

- Fundamentos de tolerância a faltas
- Sistemas distribuídos tolerantes a faltas
- Acordo em SD, replicação com coerência forte
- Replicação de sistemas de bases de dados

# Unidades Curriculares - SD
## Sistemas Distribuídos em Larga Escala



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

- Modelos e limitações da computação distribuída
- Topologias distribuídas e escaláveis
- Desenho de sistemas para larga escala
- Escalabilidade de tempo lógico
- Sincronização de dados e coerência progressiva

# Perfil de Sistemas Distribuídos



Paulo S. Almeida, José O. Pereira, Rui Oliveira, Carlos Baquero

Obrigado e boas escolhas