New Lower Bounds for Agreement

Dan Alistarh, Seth Gilbert,
Rachid Guerraoui and Corentin Travers

Introduction

- Consensus is an important problem
  - State machine replication
- ***How fast*** can we solve consensus?
- We study a generalization of consensus –

k-set agreement:
- each process proposes a value
- eventually, each process decides on one of the proposed values
- at most *k* values may be decided

Setup

- Eventually Synchronous message passing [DLS]:
  - We split the execution into *rounds*
  - There exists a time *GST* after which every message is delivered in the same round in which it was sent

Setup (2)

- Asynchronous shared memory:
  - Single Writer Multiple Reader registers
  - Wait-free implementation of snapshot
  - Weak agreement objects

Previous work

Synchrony

Synchrony:
[t/k] + 1 rounds

Early-deciding
set agreement:
[f/k] + 2 rounds

Eventual Synchrony

?

Asynchrony

∞

k-set agreement is impossible in an asynchronous system where k processes may fail

Cheat sheet

- *k-Set agreement*: a distributed problem in which processes have to collectively agree on a set of *k* values
- *Eventual Synchrony*: in every execution, there is an unknown time *GST* after which the system becomes synchronous
- *The impossibility result*: *k*-set agreement is impossible in an asynchronous system in which *k* failures can occur [BG, HS, SZ]

Contribution

- A new simulation technique, based on [Gafni98]
- Eventual Synchrony:
  - You have to pay *at least one communication round* for tolerating periods of asynchrony
- Synchrony:
  - Re-derive and extend two previous early deciding lower bounds [GHP], [GGP]

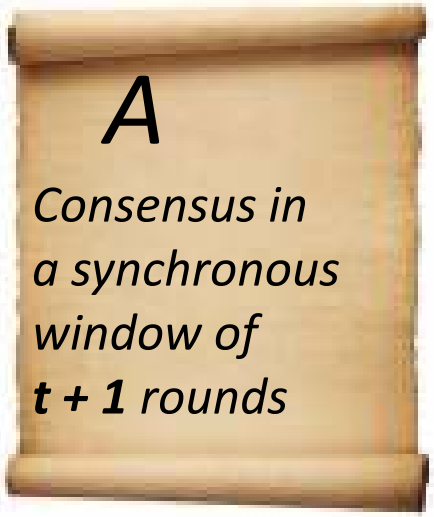
Toy example

Theorem:

No consensus algorithm in an Eventually Synchronous system decides by round *GST + t + 1*, i.e. within a window of synchrony of size *t + 1*.

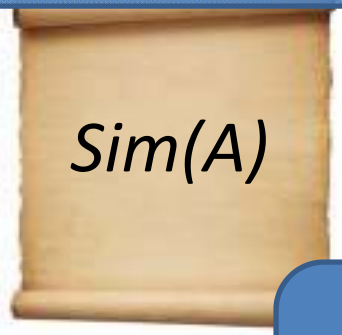
Proof:

Assume



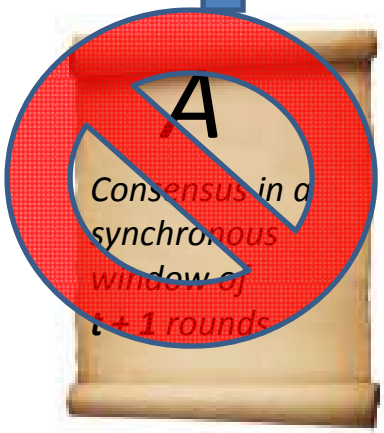
The strategy

If *A* terminates, then *Sim(A)* terminates, solving consensus



N *simulators* in an *asynchronous shared memory* system with at most ONE failure

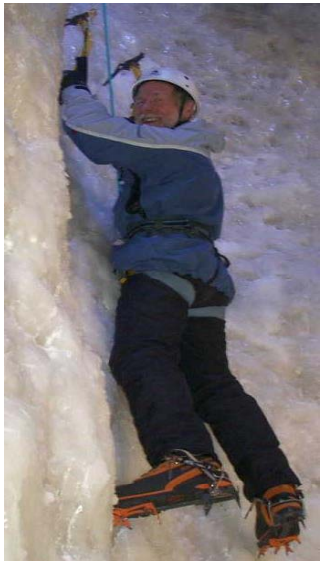
[FLP]: this is *impossible*, CONTRADICTION!



N processes, t failures, Eventually Synchronous *message passing system*

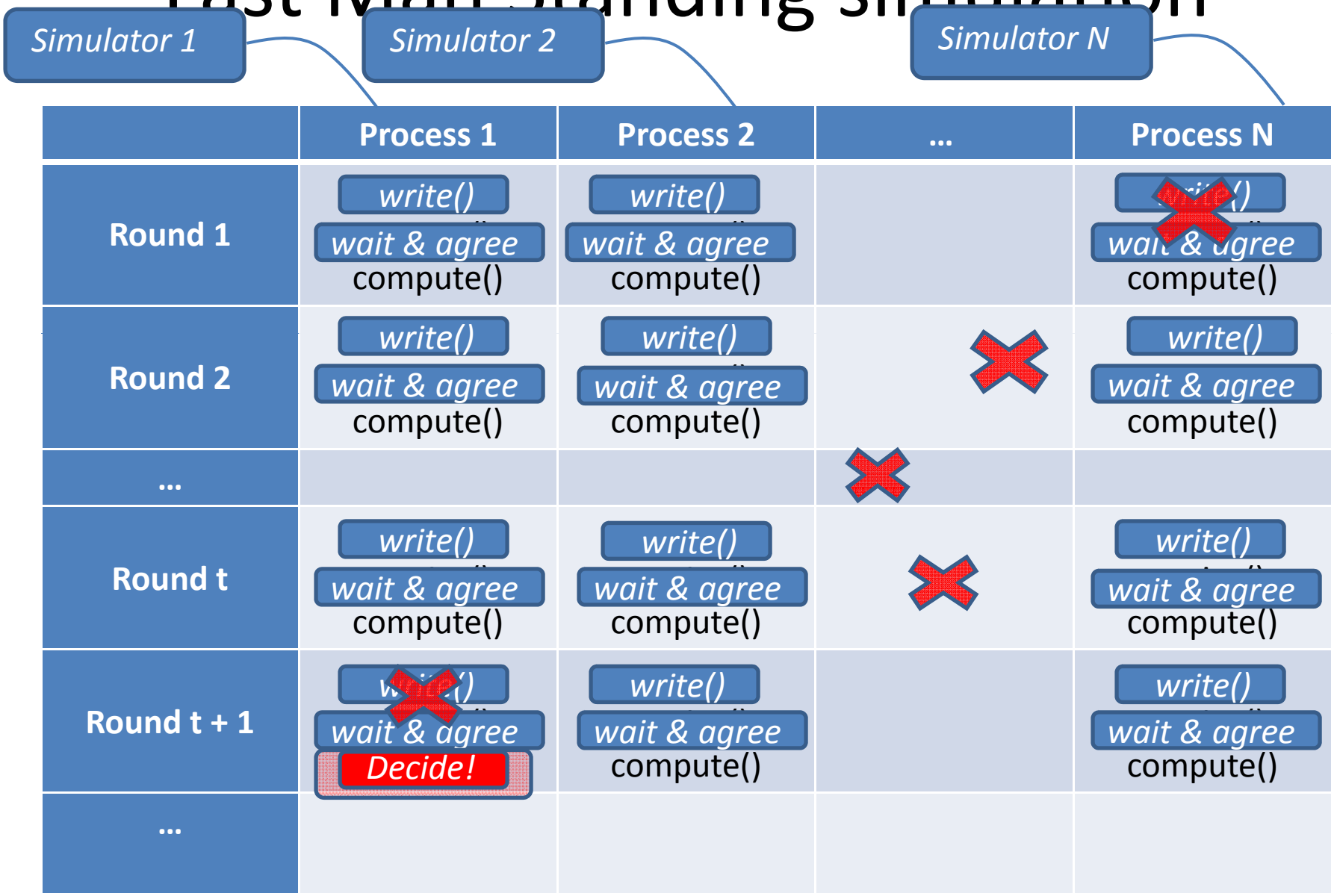
M. Fischer

N. Lynch



M. Paterson

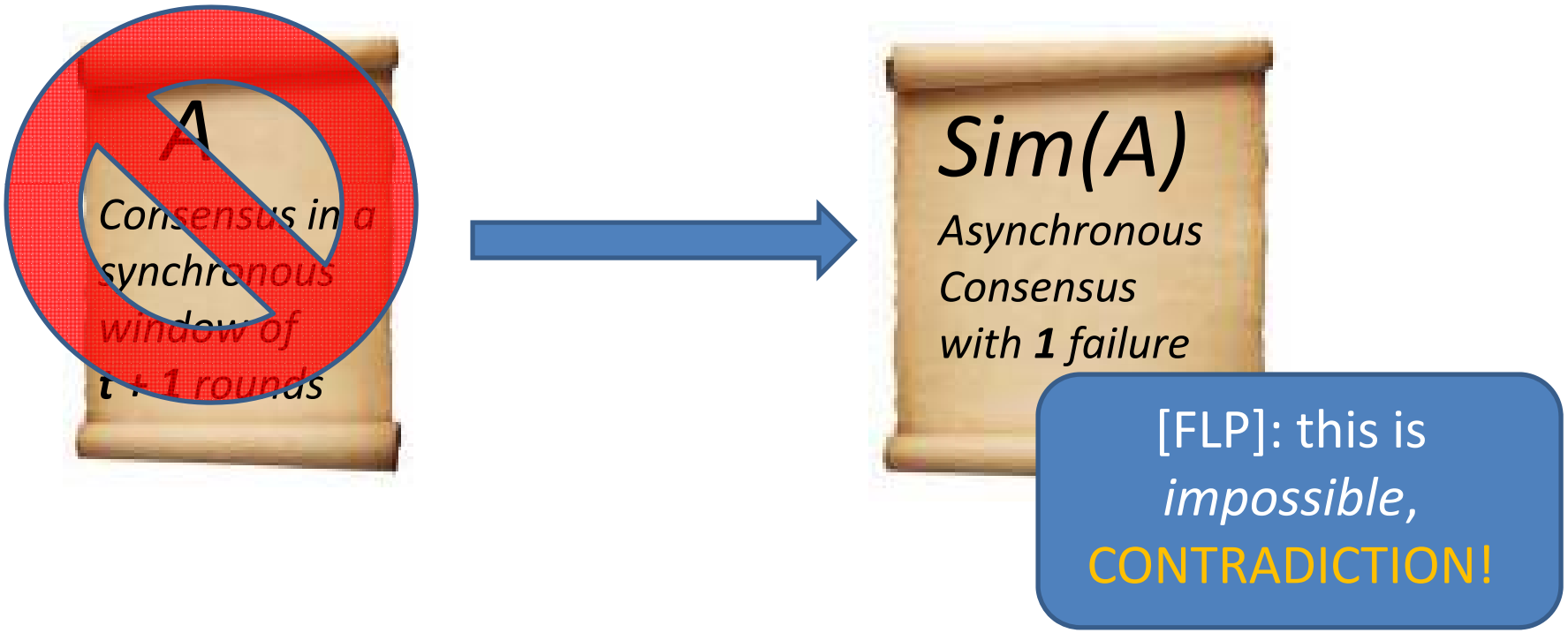
Last Man Standing simulation



What happened?

- We simulated protocol *A* for *t + 1* rounds
- *The Last Man Standing* either *decides or crashes*
- If it decides and writes its decision, *we're done*
- The other simulators **RESET** and **CONTINUE** the simulation
- So, if *LMS* crashes,
  there are *NO simulator failures left*
- …so we can go on simulating -> solve consensus!
- …and *we're done*

High-level overview



Result for Eventual Synchrony

Lower bound:

There is no *k-set* algorithm that decides by round *GST + [t/k] + 1*, i.e. within a window of synchrony of size *[t/k] + 1*.

Upper bound:

We have an algorithm *K3* that decides by round *GST + [t/k] + 3* in every execution, for $t \gg k$.

The slide to remember

You need to pay *at least one communication round* to tolerate periods of asynchrony

Simulations are a *very powerful tool* to obtain lower bounds in distributed computing

The *impossibility of asynchronous k-set agreement* is a central result in distributed computing