PAXOS

An asynchronous consensus protocol
Abstract

The Paxos algorithm, when presented in plain English, is very simple.
THREE TYPES OF PROCESSES

Proposers  A proposer is a process that has a value to propose

Acceptors  Acceptors are the processes that ultimately choose which proposed value will be decided

Learners  A learner only cares about learning which value was decided
How the game is played

• **Election:** Proposers first try to get a majority of acceptors to follow them.

• **Legislating:** After acquiring a majority, a proposer can *try* to enforce her value, by getting acceptors to accept it, *but...*

• **Playing nice:** If an elected proposer finds that some previous value has been proposed, she proposes that value instead.

• **Winning the game:** once a majority of acceptors have accepted a value, the value is *chosen/decided*
Greetings, peasants! I am your fearless leader! Grant me your blessing!

My first decree is: The value should be 12

We are with you, oh wise leader!

Sounds good to me!
HOW IT IS SUPPOSED TO WORK

Proposer

Acceptors

IAmLeader YouAreLeader Decree

Learner

Accept
Dealing with multiple proposers

Greetings, peasants! I am your fearless leader! Grant me your blessing!

We are with you, oh mighty leader!
We are with you, oh wise leader #2!

Greetings, peasants! I am your fearless leader #2! Grant me your blessing!

We are with you, oh wise leader!
We are with you, oh wise leader #2!
Dealing with multiple proposers

- I swear I won’t follow an earlier leader!
- And, btw, here is my current accepted value (if any) by leader x.
DEALING WITH MULTIPLE PROPOSERS

Proposer #1

IAmLeader #1  YouAreLeader  Decree

I swear I won’t follow an earlier leader!
And, btw, here is my current accepted value (if any) by leader x.

Proposer #2

IAmLeader #2  YouAreLeader  Decree

Acceptors

Learner

Accept
DEALING WITH MULTIPLE PROPOSERS

- I swear I won’t follow an earlier leader!
- And, btw, here is my current accepted value (if any) by leader x.
**THE CRUCIAL YouAreLeader MESSAGE**

- I swear I won’t follow an earlier leader!
- And, btw, here is my current accepted value (if any) by leader x.

1. Wait for a majority of YouAreLeader messages before proceeding.

2. If none of them contain a previously accepted value, propose your own. Otherwise, propose the value of the most recent leader.
THE CRUCIAL YouAreLeader MESSAGE

1. Wait for a majority of YouAreLeader messages before proceeding.
2. If none of them contain a previously accepted value, propose your own. Otherwise, propose the value of the most recent leader.

Important
By consulting a majority, the new leader makes sure she cannot have missed a chosen value.

(a value must be accepted by a majority to be chosen, and any two majorities overlap!)
ADMINISTRIVIA

Implementation project
• The implementation project is out; due on 10/26
  • Teams of 2

Research project
• Topic declarations due Monday 10/8

Midterm exam
• In class, 3-5pm, Wednesday 10/31
THE RESEARCH PROJECT

Sample topics:

Concrete
- Combining Fast Paxos and Flexible Paxos to reduce latency in a geo-replicated storage system
- Proving the correctness of BitCoin

Motivational
- Why the world needs real-time proofs of distributed systems
- All the things you can do with Flexible Paxos

Survey
- Applying Byzantine Fault Tolerance to blockchains: theory and practice
THE RESEARCH PROJECT

Some of last year’s topics:

• Learning to Perform Batching and Pipelining in Paxos
• Home IoT Hub Replication
• Performance Evaluation of Consensus Algorithms in Blockchain Systems
Preparing for the “research” part of the course

- Look at the papers on the website
  - You will each present one
  - Most will present solo, a few in teams of two
- Send me four preferences by Friday
- ~30-35 minutes presentation
- Send me the slides three days before your presentation
First, you should always make a script for your presentation, before you start making slides. This helps you organize your thoughts and present them clearly to your audience. The script should be at the high level, a kind of summary of the presentation with about one or two sentences per slide. Also, you should avoid having lots of text on one slide, as this is guaranteed to put your audience to a deep, dreamless slumber. Where most presentations fail is that their authors, convinced they are producing some kind of stand-alone document, put everything they want to say onto their slides, in great big chunky blocks of text. While speaking, your voice should not be a flat monotonic drone, but you should try to change inflection often, so as to avoid putting your audience to sleep. And, of course, you should never try to read aloud the text written in your slides. If you find yourself doing that during your practice talks, it means there’s something wrong with the presentation. Unless of course you are trying to make a point, as I am doing right now :)}
PRESENTATIONS (FOR REAL THIS TIME)

- Motivation, motivation, motivation!
- Keep it simple
  - Give the high-level intuition
- Avoid the “wall of text”
- Speak normally, with changes to your inflection
- Practice, practice, practice!
EXAMPLES OF ACCEPTOR STATES
(as leader #50 comes to power)

<table>
<thead>
<tr>
<th>Acceptors</th>
<th>Value</th>
<th>By leader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\times$</td>
<td>37</td>
</tr>
</tbody>
</table>
**Examples of acceptor states**  
(as leader #50 comes to power)

<table>
<thead>
<tr>
<th>Acceptors</th>
<th>Value</th>
<th>By leader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\times$</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>$y$</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td></td>
<td>$\times$</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>$y$</td>
<td>41</td>
</tr>
</tbody>
</table>
# Examples of Acceptor States

(as leader #50 comes to power)

<table>
<thead>
<tr>
<th>Acceptors</th>
<th>Value</th>
<th>By leader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>y</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>41</td>
</tr>
</tbody>
</table>
Overview of Paxos

Proposer

Send \textit{IAmLeader}(n) to all
Wait for a majority of responses

Acceptor

If \( n \) is the highest leader \# I have seen: respond with \textit{YouAreLeader}(Value, LeaderWhoProposedValue)

Once majority is received, send \textit{Propose}(n, V) where V is the highest-leader proposal among the responses (or my own value, if none of the responses had a value)

If \( n \) is the highest leader \# I have seen, send \textit{Accept}(n, V) to the learner
Tolerating $f$ failures

Safety

• There are $2f + 1$ acceptors

• A value is only chosen if accepted by a majority $(f + 1)$

• So, even if $f$ of those acceptors fail, one will remain and will be part of any future majority

Liveness

• The leader always waits for $f + 1$ responses. So, even if $f$ replicas fail, it will not block
Greetings, peasants! I am your fearless leader #1! Grant me your blessing!

Greetings, peasants! I am your fearless leader #2! Grant me your blessing!

Greetings, peasants! I am your fearless leader #3! Grant me your blessing!

Greetings, peasants! I am your fearless leader #4! Grant me your blessing!

Greetings, peasants! I am your fearless leader #5! Grant me your blessing!

Greetings, peasants! I am your fearless leader #6! Grant me your blessing!

Greetings, peasants! I am your fearless leader #7! Grant me your blessing!

Greetings, peasants! I am your fearless leader #8! Grant me your blessing!

...
The threat to liveness:
Dueling proposers

This problem can be avoided during synchrony
(proposer faults can be detected accurately using timeouts)

It’s impossible to avoid during asynchrony!

Well, we kind of knew that already…
Paxos cannot be both safe and live during asynchrony!

(that would violate FLP)

So it’s doing the next best thing:

staying **safe all the time** and achieving liveness when the system starts behaving synchronously
Using (multi)Paxos to Implement State Machine Replication

The original Paxos algorithm achieves agreement on one value.

SMR required replicas to agree on the sequence of commands that will be executed.

3. Ensure that all replicas go through the same sequence of state transitions.

MultiPaxos: Run an instance of Paxos for each slot in the sequence.

Important: we don’t need to run phase 1 (election) every time!
Paxos/SMR in real life

Proposers, acceptors and learners are all collocated on $2f + 1$ replicas
**Paxos/SMR in real life**

Proposers, acceptors and learners are all collocated on $2f + 1$ replicas.