Understanding Epidemic Quorum Systems

More details in the technical report:
http://www.gsd.inesc-id.pt/~jpbarreto

João Barreto
joao.barreto@inesc-id.pt

Paulo Ferreira
paulo.ferreira@inesc-id.pt

INESC-ID Lisbon/Technical University of Lisbon
http://www.gsd.inesc-id.pt/
Why Epidemic Quorum Systems?

- **Epidemic Quorum Systems** \([1,2]\) seem advantageous for agreement in weakly-connected environments
  - e.g. mobile and sensor networks.
  - agreement of a value from a set of proposed values is often required in distributed systems (e.g. replication protocols, distributed mutual exclusion, ..)
- **Whereas Classical Quorum Systems** require a *quorum* of processes to be simultaneously connected to agree…
- **Epidemic Quorum Systems** do not require simultaneous connection of quorum
  - Processes vote locally for one of the proposed values
  - Votes propagate epidemically
  - Processes decide one value based on the votes they collect
### Example (from the viewpoint of process $p_1$)

**5 processes with temporary partitions**

<table>
<thead>
<tr>
<th>Time</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Proposes X</td>
<td>Proposes Y</td>
<td>Proposes Z</td>
<td>$p_1$ votes for X</td>
<td>$p_1$ votes for X</td>
</tr>
<tr>
<td></td>
<td>No plurality yet.</td>
<td>No plurality yet.</td>
<td>No plurality yet.</td>
<td>$p_1$ knows that $p_1$ and $p_2$ have voted for X.</td>
<td>$p_1$ knows that $p_1$ and $p_2$ have voted for X.</td>
</tr>
<tr>
<td></td>
<td>$p_1$ knows that $p_1$ and $p_2$ have voted for X.</td>
<td>$p_1$ knows that $p_1$ and $p_2$ have voted for X.</td>
<td>$p_1$ knows that $p_1$, $p_2$, and $p_3$ have voted for X.</td>
<td>X has a plurality of votes. <strong>Hence $p_1$ decides X.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_1$ fails to obtain quorum of 3 accessible processes.</td>
<td>$p_1$ fails to obtain quorum of 3 accessible processes.</td>
<td>$p_1$ fails to obtain quorum of 3 accessible processes.</td>
<td>$p_1$ fails to obtain quorum of 3 accessible processes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_1$ obtains quorum of 3 accessible processes.</td>
<td>$p_1$ obtains quorum of 3 accessible processes.</td>
<td>$p_1$ obtains quorum of 3 accessible processes.</td>
<td>$p_1$ obtains quorum of 3 accessible processes.</td>
<td></td>
</tr>
</tbody>
</table>

#### Epidemic Plurality
Winner must have more votes than the votes that any other value may collect in the worst case; ties are broken in favour of the process with the lowest identifier.

- $p_1$ votes for X. No plurality yet.

#### Epidemic Majority
Winner must have more than half the votes.

- $p_1$ votes for X. No majority yet.

#### Classical Majority
Quorum of more than half the processes must be acquired.

- $p_1$ fails to obtain quorum of 3 accessible processes.
However, Epidemic Quorum Systems are not well studied...

Previous work does not answer the following questions:

2. Under **which conditions** do Epidemic Quorum Systems become **advantageous** over Classic Quorum Systems?

3. What are the **inherent trade-offs** of Epidemic Quorum Systems?

4. How can we precisely **characterize the liveness** of Epidemic Quorum Systems?
   – In particular, how do proposed Epidemic Quorum Systems compare in terms of availability and performance?

5. What is the **best** Epidemic Quorum System?

And known results for Classical Quorum Systems [3] are not valid for Epidemic Quorum Systems.
Defining an Epidemic Coterie (EC)

Intuitively, we define an EC as:

• a collection of <Quorum; Anti-Quorums> pairs
• where:

4. for each pair, the Quorum and every Anti-Quorum are disjoint sets of processes
5. and, given a pair in the EC, we cannot:

1. pick one of its Anti-Quorums and make it the new Quorum (swapping it with the original Quorum),
2. add arbitrary processes (not yet included in the pair) to any Quorum/Anti-Quorum,
3. and obtain a different pair of the EC.

Definition 1. Vote set, vote configuration and q-vote configuration

A vote set $s$ is a set of processes, $s \subseteq U$.

A vote configuration $c$ is a set of non-empty vote sets, $c = \{s_1, s_2, \ldots, s_n\}$, such that $\forall s_1, s_2 \in c, s_1 \cap s_2 = \emptyset$.

A q-vote configuration $qc$ is a pair, $qc = (Q_{qc}, \{A_{qc}^1, A_{qc}^2, \ldots, A_{qc}^{n_{qc}}\})$, where $\{Q_{qc}, A_{qc}^1, \ldots, A_{qc}^{n_{qc}}\}$ is a vote configuration.

Definition 2. Potential vote set

Let $s$ be a vote set. The potential vote set of $s$ in a vote configuration $c$, denoted $[s]_c$, is defined as $[s]_c = s \cup (U \setminus \bigcup \{s : s \in c\})$. In particular, $[s]_c = (U \setminus \bigcup \{s : s \in c\})$.

Further, the potential vote set of $s$ in a q-vote configuration $qc$, denoted $[s]_{qc}$, is defined as $[s]_{qc} = s \cup (U \setminus (Q_{qc} \cup A_{qc}^1 \cup \ldots \cup A_{qc}^{n_{qc}}))$.

Definition 3. Coverage and potential coverage between q-vote configurations

Let $qc$ and $qd$ be q-vote configurations.

We say $qc$ covers $qd$, or $qc > qd$, if and only if:

1. $Q_{qc} \supseteq Q_{qd}$, and
2. There exists an injective function $f : \{1, \ldots, n_{qc}\} \rightarrow \{1, \ldots, n_{qc}\}$ such that
   $\forall 1 \leq k \leq n_{qc} : A_{qc}^{f(k)} \supseteq A_{qd}^k$.

We say $qc$ may cover $qd$, or $qc \triangleright qd$, if and only if:

1. $[Q_{qc}]_c \supseteq Q_{qd}$, and
2. There exists an injective function $f : \{1, \ldots, n_{qc}\} \rightarrow \{1, \ldots, n_{qc}\}$ such that
   $\forall 1 \leq k \leq n_{qc} : [A_{qc}^{f(k)}]_c \supseteq A_{qd}^k$ or $[s]_{qc} \supseteq A_{qd}^k$.

Definition 4. Epidemic Coterie

Let $\mathcal{E}$ be a non-empty set of q-vote configurations. $\mathcal{E}$ is an Epidemic Coterie (EC) if, $\forall c, d \in \mathcal{E} : c \neq d$:

1. $\forall j : (A_{c}^j \cup \{A_{c}^i : i \neq j\}) \nsubseteq d$, and
2. $([s]_c \cup \{A_{c}^i : i \neq j\}) \nsubseteq d$, and
3. $c \nsubseteq d$. 
Epidemic Quorum Algorithm

- Processes propose and vote for a value by adding their vote to the local vote set of that value
  - A process may only vote once and may not retract its vote in each election
- Vote sets propagate epidemically
- Once a process is able to compose a
  - \(<\text{Quorum}; \text{Anti-Quorums}>\) pair from the EC using its local vote sets, it decides the value corresponding to the vote set that is the Quorum
- Once a process determines that no pair in the EC may ever be composed (with the still unknown votes), it starts a new election (with empty vote sets)
Characterizing Availability and Performance

• We assume:
  – An asynchronous system
  – Probability of non-Byzantine permanent failure is uniform and constant
  – Probability of two processes being inaccessible due to transient partitioning is uniform and constant
  – Z values proposed, with uniform probability of being voted for

• Given an EC, determine the probability distributions:
  – Probability that a process, having collected N votes, decides some value
  – Probability that a process, having collected N votes, starts a new election

• Having the probability distributions we have obtained analytical expressions for:
  – Probability that a process eventually decides some value (availability)
  – Probability that a process decides some value within r communication rounds (performance)
Future Work

• Use the formalism to devise novel ECs and study their availability and performance
• Compare Epidemic and Classical Quorum Systems
• What EC offers optimal availability and/or performance?

References

