Probabilistic Quorum Systems

Dahlia Malkhi • Michael Reiter • Rebecca Wright
Proc. of 16th ACM Symp. on Principles of Distributed Computing (PODC '97), pp. 267-273, 1997.

Presented by Xintong Wang and Ryan Marcotte

Quorum Systems

- Definition: a set of subsets of servers, every pair of which intersects.

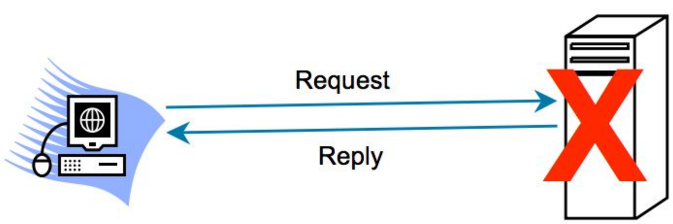
  Given a universe U of servers where $\mathrm{U} = \{u_1, u_2, ..., u_n\}$ and $|\mathrm{U}| = n$, a (strict) quorum system $\mathcal{Q}$ over a universe U is a set system over U such that

  (1) $\mathcal{Q} \subseteq \mathcal{P}(\mathrm{U})$

  (2) $\forall \mathrm{Q}_1, \mathrm{Q}_2 \in \mathcal{Q}, \mathrm{Q}_1 \cap \mathrm{Q}_2 \neq \emptyset$

  Each Q is a quorum and $\mathcal{Q}$ is a (strict) quorum system.

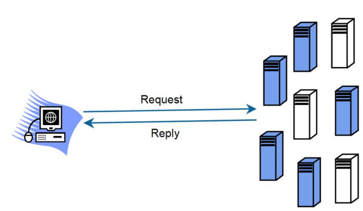
Quorum Systems

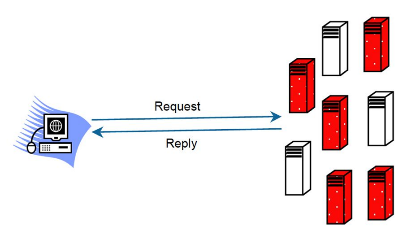


Quorum Systems

- Motivation:

  System-wide consistency can be maintained by allowing any quorum to act on behalf of the entire system.

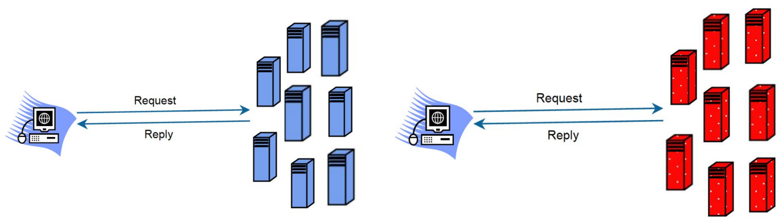




Quorum Systems

- Why not performing every operation at every server?

  Using quorums reduces the load on servers and increases service availability despite server crashes.



Quorum Systems

- Quorum systems have been used to implement a wide variety of distributed objects and services:
  1. Replicated databases
  2. Read/write storage
  3. Group communication

t-dissemination Quorum System [MR97]

- A (strict) quorum system with (2) changed to
  $\forall Q_1, Q_2 \in \mathcal{Q}, |Q_1 \cap Q_2| \geq t+1$
- A collection of subsets of servers, each pair of which intersect in a set containing sufficiently many correct servers to guarantee consistency of the replicated data as seen by clients.

Access Strategy (Client)

- An access strategy w for a set system $\mathcal{Q}$ specifies a probability distribution on the elements of $\mathcal{Q}$, $w : \mathcal{Q} \rightarrow [0,1]$ satisfies $\sum_{Q \in \mathcal{Q}} w(Q) = 1$.
- Example:

$$\mathcal{Q} = \{\{1,4,6\},\{2,4,7\},\{3,5,6,7\},\{1,2,3,5\},\{1,2,3,4\},\{2,3,4,5\},\{3,4,5,6\},\{4,5,6,7\},\{5,6,7,1\},\{6,7,1,2\},\{7,1,2,3\}\}$$

$$w = \{0,0,0,0,\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7}\} \qquad w' = \{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0,0,0,0,0,0,0\}$$

Measurements on Quorum Systems

- Load - the rate at which the busiest server will be accessed by an optimal strategy.
- Fault Tolerance - the number of servers that can fail without disabling the system.
- Failure Probability - the probability that the system is disabled.

- Example:
  $\mathcal{Q} = \{\{1,4,6\},\{2,4,7\},\{3,5,6,7\},\{1,2,3,5\},\{1,2,3,4\},\{2,3,4,5\},$
  $\{3,4,5,6\},\{4,5,6,7\},\{5,6,7,1\},\{6,7,1,2\},\{7,1,2,3\}\}$

  $w = \{0,0,0,0,\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7},\frac{1}{7}\}$ $w' = \{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0,0,0,0,0,0,0\}$

  $L_w(\mathcal{Q}) = \frac{4}{7}$ $L_{w'}(\mathcal{Q}) = \frac{1}{2}$

Load [NW94]

- Consider an access strategy $w \in W$ for a quorum system $\mathcal{Q}$ over a universe U
  The load induced by a strategy w on a server u

  $$l_w(u) = \sum_{u \in \mathrm{Q}_i} w(\mathrm{Q}_i)$$

- The load induced by a strategy w on $\mathcal{Q}$
  $$L_w(\mathcal{Q}) = \max_{u \in U} l_w(u)$$
- The load of $\mathcal{Q}$
  $$L(\mathrm{Q}) = \min_{w \in W} L_w(\mathcal{Q})$$

Interpretation of Load

- Load is a best-case definition (optimal access strategy) of a worst-behavior (busiest server) property.
- Load is a measure of efficiency; all other things equal, systems with lower load can process more requests.
- Load is a property inherent to the combinatorial structure of the quorum system, and not to the protocol using the system.
- When defining load, we are assuming that all the servers in the universe are functioning, so all the quorums of the system are usable.

Fault Tolerance

- Consider a quorum system $\mathcal{Q} = \{\mathrm{Q}_1, ..., \mathrm{Q}_m\}$ and $\mathcal{S} = \{\mathrm{S} \mid \mathrm{S} \cap \mathrm{Q}_i \neq \emptyset, 1 \leq i \leq m\}$
- The fault tolerance of the system $\mathcal{Q}$ is

$$A(\mathcal{Q}) = \min_{S \in \mathcal{S}} |S|$$

- The size of the smallest set of servers that intersects all quorums.

Interpretation of Fault Tolerance

- A quorum system is resilient to the failure of any set of $A(\mathcal{Q}) - 1$ or fewer servers.
- Some particular set of $A(\mathcal{Q})$ failures can disable all quorums in the system.

Failure Probability

- Assume that each server in U fails independently with probability p, the failure probability $F_p(\mathcal{Q})$ of $\mathcal{Q}$ is the probability that every $Q \in \mathcal{Q}$ contains at least one faulty server.
- $$\text{when } p < \frac{1}{2}, \lim_{n \to \infty} F_p(\mathcal{Q}) = 0;$$
  $$\text{when } p = \frac{1}{2}, \exists \mathcal{Q}, \text{s.t. } F_p(\mathcal{Q}) = \frac{1}{2}$$
  $$\text{when } p > \frac{1}{2}, F_p(\mathcal{Q}) \to 1.$$

Load vs. Fault Tolerance Tradeoff

The load of strict quorum system has the following lower bound:

$$L(\mathcal{Q}) \geq \frac{c(\mathcal{Q})}{n}$$

Set of all quorums

Total number of servers

The failure of any full quorum will disable all quorums:

$$A(\mathcal{Q}) \leq c(\mathcal{Q})$$

Size of smallest quorum

$$A(\mathcal{Q}) \leq nL(\mathcal{Q})$$

There is a tradeoff between load and fault tolerance in strict quorum systems

Probabilistic Quorum Systems

Pair of intersecting quorums

$$\sum_{Q,Q':(Q \bigcap Q') \neq \emptyset} w(Q)w(Q') \geq 1 - \epsilon$$

Access probability of each quorum

Small constant in (0,1)

- Meaning of ε
  - Probability of accessing non-intersecting quorums
  - Represents desired level of consistency
  - Different values lead to different quorum systems
- Access strategy *w*
  - Selected to achieve highest level of performance
  - Other access strategies may lead to system failure
  - Change to definition of load

Lower Bound on Load

Strict Quorum Systems:

Size of smallest quorum

$$L(\mathcal{Q}) \geq \max\left\{\frac{1}{c(\mathcal{Q})}, \frac{c(\mathcal{Q})}{n}\right\}$$

Set of all quorums

Total number of servers

Note the similarities!

Improvement over strict quorum systems does have limits

Probabilistic Quorum Systems:

$$L_w(\mathcal{Q}) \geq (1 - \sqrt{\epsilon}) \max\left\{\frac{1}{c(\mathcal{P})}, \frac{c(\mathcal{P})}{n}\right\}$$

Probability of accessing non-intersecting quorums

Set of quorums with high (i.e. lower-bounded) likelihood of accessing an intersecting quorum

Probabilistic Quorum Construction

The quorums are all possible sets of the specified size

$$\mathcal{Q} = \left\{Q \subset U : |Q| = l\sqrt{n}\right\}$$

Total number of servers

They have uniform access probabilities

Access probability of each quorum

$$w(Q) = \frac{1}{|\mathcal{Q}|}, \forall Q \in \mathcal{Q}$$

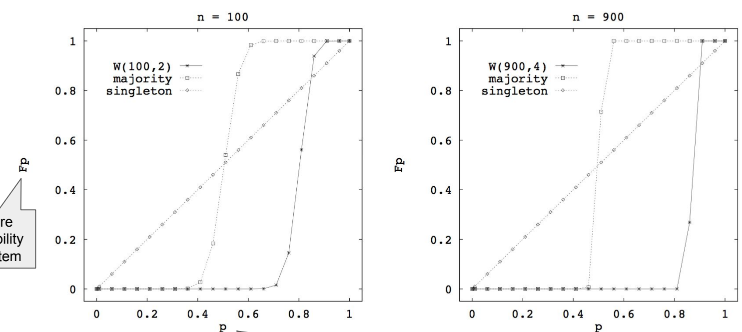
With ε define as

Probability of accessing non-intersecting quorums

$$\epsilon = \exp(-l^2)$$

Performance vs. Majority/Singleton

Failure probability of system



Independent failure probability of each server

Byzantine Fault Tolerance

- Fail-stop failure model
  - Only node failures are node crashes
  - Detectable by other nodes
- Byzantine failure model
  - Most general and difficult failure mode
  - No restrictions on types of failures
  - Failed nodes may generate arbitrary data or pretend to be operational

Probabilistic dissemination quorum systems

Probability of accessing quorums without sufficient intersection

Pair of quorums with sufficiently large intersection

$$\sum_{Q,Q' : Q \bigcap Q' \nsubseteq B} w(Q)w(Q') \geq 1 - \epsilon$$

Access probability of each quorum

$$\forall B \subseteq U \text{ s.t. } |B| = t$$

Number of Byzantine errors that can be tolerated

The universe of all servers

Can be used to overcome any fraction of the total number of servers experiencing Byzantine failure

Improvements and Extensions

- Practical implementation of the system
  - Designing reliable distributed systems
  - Providing reliable storage in mobile ad hoc networks

    Luo, Jun, Jean-Pierre Hubaux, and Patrick Th Eugster. "PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems." *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003.
  - Key predistribution scheme for wireless sensor networks

    Du, Wenliang, et al. "A pairwise key predistribution scheme for wireless sensor networks." *ACM Transactions on Information and System Security (TISSEC)* 8.2 (2005): 228-258.
- Elegant mathematics, but can all claims be achieved in real world?
  - In particular, overcoming constant fraction of Byzantine failures seems prohibitively expensive.