Replication Models: Summary

Ken Birman

Cornell University. CS5410 Fall 2008.

Snapshot: Replication "models"

- By now we're starting to see that "replication" comes in many flavors
  - No model: UDP multicast (IPMC), Scalable Reliable Multicast, TCP. Often called "best effort" but not always clear what this really means. In practice, loss occurs on sockets, not network. SRM uses timesouts, NAKs, retransmission to recover from loss, but with timeout at the core, model is like TCP –weak semantics,
  - State machine model (GMS views, Paxos). Needs strong determinism. No partitioning (split brain). Group membership confers strong semantics. Can't guarantee termination (FLP)
  - Even stronger: Byzantine (State Machines + malicious nodes), Transactional (for databases with ACID properties)
  - Probabilistic: Ricochet, Gossip: Converge towards guarantees

Replication protocols

Type	Capsule Summary	Pros	Cons
UDP multicast	Fast, pretty reliable unless overloaded. But not always supported ("fear of multicast", WAN issues)	Raw speed: send 1, get n-1 deliveries for free	Router load, "n:1" effect (instability), no flow control
SRM (Scalable Reliable Multicast)	A reliable protocol that runs over UDP multicast, well known and fairly popular. eBay uses it internally.	Uses UDP multicast for NAK, retransmissions	Great when all goes well, but prone to sudden destabilization
GMS view updt	Usually 2-phase, hence "pretty fast". Can't partition (no split brain)	State machine model applies	Slower than UDP multicast, scales poorly
Vsync	Hosted within GMS, like a reliable UDP multicast + view synchrony	Like state machine but more flexibility	User needs to take cs5410 first! And can it scale?
Paxos	Like GMS view update, several versions. One has a very elegant proof of safety	State machine model	Slower than UDP multicast, scales poorly
Byzantine	These assume that at most *t* of *N* members of the service are malicious. Trusts clients.	State machine model	Hardens service but not its clients
Ricochet	Seeks rapid, probabilistically reliable delivery	Very stable, scalable	Not as strong as vsync or state machine model
Transactions	ACID database guarantees (1-copy serializability)	Famous model	Very poor scalability
Gossip	Convergent probabilistic guarantees, constant overhead costs	Very robust at constant (low) cost, scales well	Too slow for some uses

Today: Ricochet

- Remainder of today's lecture will look at Ricochet
  - Time-critical multicast protocol
  - May become a standard in Red Hat Linux and other data center / enterprise settings
  - Great stability and scalability, quasi-realtime guarantees
- Paper in NSDI 2007 has details