CSE 60641: Operating Systems

- **Implementing Fault-Tolerant Services Using the State Machine Approach: a tutorial Fred B. Schneider, ACM Computing Surveys 22(4) :299-319, December 1990**
  - **Distributed systems paper, SOSP Hall of Fame winner**
  - **Citation: The paper that explained how we should think about replication ... a model that turns out to underlie Paxos, Virtual Synchrony, Byzantine replication, and even Transactional 1-Copy Serializability**
  - **Theoretical foundations, we will read one or maybe two more papers like this one. These papers should be a paper in your bag-of-tricks**

Paxos

- "A fault-tolerant file system called Echo was built at SRC in the late 80s. The builders claimed that it would maintain consistency despite any number of non-Byzantine faults, and would make progress if any majority of the processors were working. As with most such systems, it was quite simple when nothing went wrong, but had a complicated algorithm for handling failures based on taking care of all the cases that the implementers could think of. I decided that what they were trying to do was impossible, and set out to prove it. Instead, I discovered the Paxos algorithm. Paxos contains the first three-phase commit algorithm that is a real algorithm, with a clearly stated correctness condition and a proof of correctness. "

  – Leslie Lamport

Paxos algorithm

- Assume a collection of processes that can propose values. A consensus algorithm ensures that a single one among the proposed values is chosen. If no value is proposed, then no value should be chosen. If a value has been chosen, then processes should be able to learn the chosen value. The safety requirements for consensus are:
  - Only a value that has been proposed may be chosen,
  - Only a single value is chosen, and
  - A process never learns that a value has been chosen unless it actually has been
  - Leslie Lamport

Virtual Synchrony

Virtual synchrony is an interprocess messaging passing (sometimes called event queue management) technology. Virtual synchrony systems allow programs running in a network to organize themselves into process groups, and to send messages to groups (as opposed to sending them to specific processes). Each message is delivered to all the group members, in the identical order, and this is true even when two messages are transmitted simultaneously by different senders. Application design and implementation is greatly simplified by this property: every group member sees the same events (group membership changes, and incoming messages) and in the same order. – Wikipedia

Exploiting virtual synchrony in distributed systems". K.P. Birman and T. Joseph. SOSP ‘87

Byzantine Generals problem

- We imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement. The generals must have an algorithm to guarantee that
  - All loyal generals decide upon the same plan of action
  - A small number of traitors cannot cause the loyal generals to adopt a bad plan
- The Byzantine Generals Problem, Leslie Lamport, Marshall Pease and Robert Shostak, ACM Transactions on Programming Languages and Systems 4, 3, July 1982.

Distributed Systems

- Clients and Services
  - The goal is to provide fault tolerant services
- Failure models:
  - Fail stop – t component failure requires t+1 copies
  - Byzantine – t component failure requires 2t+1 copies
  - More precise than statistical measures such as MTBF
- Replica coordination: all replicas receive and process the same sequence of requests
  - Agreement: Every non faulty state machine replica receives every request
  - Order: Every non faulty state machine replica processes the requests it receives in the same relative order

- Agreement
  - All non faulty processors agree on the same value
  - If the transmitter is non faulty, then all non faulty processors use its value as the one on which they agree
- Order
  - A replica processes the next stable request with the smallest unique identifier
    - Use logical clocks (we will read this paper later)
    - FIFO channels
    - Synchronized real time clocks
- Stability: request is stable if a larger unique identifier has been received from every client

- Replica generated identifiers
- Client generated identifiers

- Tolerating Faulty Output devices
  - Outputs used outside the system
  - Outputs used inside the system

- Tolerating faulty clients
  - Replicating clients
- Reconfiguration