Distributed system design, 100K ft level

April 6, 2016
Distributed system design

• What do you need to think about when designing a distributed system?
Distributed system design

• What do you need to think about when designing a distributed system?
  • System API
  • Node roles
  • Network
  • System state
  • Failures
System API

• Who are the clients of the system?

• What do they assume about the system?

• How do they contact the system?

• Concurrent clients?

• Do clients know about one another?

• How can clients interfere with one another?

• Do we trust the clients? How much and with what?
System API: Assignment 4

• Who are the clients of the system?
• What do they assume about the system?
• How do they contact the system?
• Concurrent clients?
• Do clients know about one another?
• How can clients interfere with one another?
• Do we trust the clients? How much and with what?
• Who are the clients of the system?
  • Other hosts, no particular designation/filtering or assumptions about who they might be

• What do they assume about the system?
  • A4: front-end always available, serializable consistency semantics,…
  • A4 bonus: front-end may fail (client has to retry)

• How do they contact the system?
  • Front-end node services all client requests. put(), get(), testset() API calls

• Concurrent clients?
  • Yes.

• Do clients know about one another?
  • Maybe. Indirectly know about each other through the system.

• How can clients interfere with one another?
  • Yes, they have to coordinate on their own.

• Do we trust the clients? How much and with what?
  • Yes, all clients trusted to follow API.
Node roles

• What are the different roles that nodes play in the system?
• What makes each role distinct and necessary?
• Which roles need to interact?
• What do different node roles assume about one another?
  • What is the API between node roles? (cross-cutting)
  • All API questions apply: e.g., what is the trust between roles?
Node roles: A4

• What are the different roles that nodes play in the system?
• What makes each role distinct and necessary?
• Which roles need to interact?
• What do different node roles assume about one another?
  • What is the API between node roles? (cross-cutting)
  • All API questions apply: e.g., what is the trust between roles?
Node roles: BitTorrent

- What are the different roles that nodes play in the system?
- What makes each role distinct and necessary?
- Which roles need to interact?
- What do different node roles assume about one another?
- What is the API between node roles? (cross-cutting)
- All API questions apply: e.g., what is the trust between roles?
Roles in BT

- What are the different roles that nodes play in the system?
- What makes each role distinct and necessary?
- Which roles need to interact?
- What do different node roles assume about one another?
  - What is the API between node roles? (cross-cutting)
  - All API questions apply: e.g., what is the trust between roles?
Network

- What is the network model; what does the network provide?
- What is the network API? And, what are its semantics?
- How do we name entities in the network and how do we find/look them up?
- What is the network topology?
- Do we trust the network? With what?
Network: CODA

- What is the network model; what does the network provide?
- What is the network API? And, what are its semantics?
- How do we name entities in the network and how do we find/look them up?
- What is the network topology?
- Do we trust the network? With what?
Network: CODA

- What is the network model; what does the network provide?
- What is the network API? And, what are its semantics?
- How do we name entities in the network and how do we find/look them up?
- What is the network topology?
- Do we trust the network? With what?
System state

• What is the distributed system state?
• What is not distributed system state?
• What nodes have what state in the system?
• What distributed state can clients observe?
• What are the semantics of distributed state? Is this a function of node type, location, or other features of the system?
System state: BT

- What is the distributed system state?
- What is not distributed system state?
- What nodes have what state in the system?
- What distributed state can clients observe?
- What are the semantics of distributed state? Is this a function of node type, location, or other features of the system?
System state: A4

- What is the distributed system state?
- What is not distributed system state?
- What nodes have what state in the system?
- What distributed state can clients observe?
- What are the semantics of distributed state? Is this a function of node type, location, or other features of the system?
Failure (cross-cutting)

- What failures are outside the scope of what the system can deal with?
- Can the network fail, how? How does the system respond?
- Can nodes in the system fail, how? How does the system respond?
- Can clients fail? How does that impact the system?
- Can the system provide graceful degradation?
- Is there fate sharing in the system? Between what roles?
Failure: A4

- What failures are outside the scope of what the system can deal with?
- Can the network fail, how? How does the system respond?
- Can nodes in the system fail, how? How does the system respond?
- Can clients fail? How does that impact the system?
- Can the system provide graceful degradation?
- Is there fate sharing in the system? Between what roles?
Recap

• What do you need to think about when designing a distributed system?
  • System API
  • Node roles
  • Network
  • System state
  • Failures