A Distributed Systems Reading List

Introduction

I often argue that the toughest thing about distributed systems is changing the way you think. The below is a collection of material I've found useful for motivating these changes.

Thought Provokers

Ramblings that make you think about the way you design. Not everything can be solved with big servers, databases and transactions.

- Harvest, Yield and Scalable Tolerant Systems - Real world applications of CAP from Brewer et al
- On Designing and Deploying Internet Scale Services - James Hamilton
- The Perils of Good Abstractions - Building the perfect API/interface is difficult
- Chaotic Perspectives - Large scale systems are everything developers dislike - unpredictable, unordered and parallel
- Data on the Outside versus Data on the Inside - Pat Helland
- Memories, Guesses and Apologies - Pat Helland
- SOA and Newton's Universe - Pat Helland
- Building on Quicksand - Pat Helland
- Why Distributed Computing? - Jim Waldo
- A Note on Distributed Computing - Waldo, Wollrath et al
- Stevey's Google Platforms Rant - Yegge's SOA platform experience

Latency

- Latency Exists, Cope! - Commentary on coping with latency and it's architectural impacts
- Latency - the new web performance bottleneck - not at all new (see Patterson), but noteworthy
- The Tail At Scale - the latencychallenges inherent of dealing with latency in large scale systems

Amazon

Somewhat about the technology but more interesting is the culture and organization they've created to work with it.

- A Conversation with Werner Vogels - Coverage of Amazon's transition to a service-based architecture
- Discipline and Focus - Additional coverage of Amazon's transition to a service-based architecture
- Vogels on Scalability

- SOA creates order out of chaos @ Amazon

Google

Current "rocket science" in distributed systems.

- MapReduce
- Chubby Lock Manager
- Google File System
- BigTable
- Data Management for Internet-Scale Single-Sign-On
- Dremel: Interactive Analysis of Web-Scale Datasets
- Large-scale Incremental Processing Using Distributed Transactions and Notifications
- Megastore: Providing Scalable, Highly Available Storage for Interactive Services - Smart design for low latency Paxos implementation across datacentres.
- Spanner - Google's scalable, multi-version, globally-distributed, and synchronously-replicated database.
- Photon - Fault-tolerant and Scalable Joining of Continuous Data Streams. Joins are tough especially with time-skew, high availability and distribution.
- Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing - Data warehousing system that stores critical measurement data related to Google's Internet advertising business.

Consistency Models

Key to building systems that suit their environments is finding the right tradeoff between consistency and availability.

- CAP Conjecture - Consistency, Availability, Parition Tolerance cannot all be satisfied at once
- Consistency, Availability, and Convergence - Proves the upper bound for consistency possible in a typical system
- CAP Twelve Years Later: How the "Rules" Have Changed - Eric Brewer expands on the original tradeoff description
- Consistency and Availability - Vogels
- Eventual Consistency - Vogels
- Avoiding Two-Phase Commit - Two phase commit avoidance approaches
- 2PC or not 2PC, Wherefore Art Thou XA? - Two phase commit isn't a silver bullet
- Life Beyond Distributed Transactions - Helland
- If you have too much data, then 'good enough' is good enough - NoSQL, Future of data theory - Pat Helland
- Starbucks doesn't do two phase commit - Asynchronous mechanisms at work
- You Can't Sacrifice Partition Tolerance - Additional CAP commentary
- Optimistic Replication - Relaxed consistency approaches for data replication

Theory

Papers that describe various important elements of distributed systems design.

- Distributed Computing Economics - Jim Gray
- Rules of Thumb in Data Engineering - Jim Gray and Prashant Shenoy
- Fallacies of Distributed Computing - Peter Deutsch
- Impossibility of distributed consensus with one faulty process - also known as FLP [access requires account and/or payment, a free version can be found here]
- Unreliable Failure Detectors for Reliable Distributed Systems. A method for handling the challenges of FLP
- Lamport Clocks - How do you establish a global view of time when each computer's clock is independent
- The Byzantine Generals Problem
- Lazy Replication: Exploiting the Semantics of Distributed Services
- Scalable Agreement - Towards Ordering as a Service
- Scalable Eventually Consistent Counters over Unreliable Networks - Scalable counting is tough in an unreliable world

Languages and Tools

Issues of distributed systems construction with specific technologies.

- Programming Distributed Erlang Applications: Pitfalls and Recipes - Building reliable distributed applications isn't as simple as merely choosing Erlang and OTP.

Infrastructure

- Principles of Robust Timing over the Internet - Managing clocks is essential for even basics such as debugging

Storage

- Consistent Hashing and Random Trees
- Amazon's Dynamo Storage Service

Paxos Consensus

Understanding this algorithm is the challenge. I would suggest reading "Paxos Made Simple" before the other papers and again afterward.

- The Part-Time Parliament - Leslie Lamport
- Paxos Made Simple - Leslie Lamport
- Paxos Made Live - An Engineering Perspective - Chandra et al
- Revisiting the Paxos Algorithm - Lynch et al

- How to build a highly available system with consensus - Butler Lampson
- Reconfiguring a State Machine - Lamport et al - changing cluster membership
- Implementing Fault-Tolerant Services Using the State Machine Approach: a Tutorial - Fred Schneider

Other Consensus Papers

- Mencius: Building Efficient Replicated State Machines for WANs - consensus algorithm for wide-area network

Gossip Protocols (Epidemic Behaviours)

- How robust are gossip-based communication protocols?
- Astrolabe: A Robust and Scalable Technology For Distributed Systems Monitoring, Management, and Data Mining
- Epidemic Computing at Cornell
- Fighting Fire With Fire: Using Randomized Gossip To Combat Stochastic Scalability Limits
- Bi-Modal Multicast
- ACM SIGOPS Operating Systems Review - Gossip-based computer networking
- SWIM: Scalable Weakly-consistent Infection-style Process Group Membership Protocol

P2P

- Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications
- Kademlia: A Peer-to-peer Information System Based on the XOR Metric
- Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems
- PAST: A large-scale, persistent peer-to-peer storage utility - storage system atop Pastry
- SCRIBE: A large-scale and decentralised application-level multicast infrastructure - wide area messaging atop Pastry