ACID database transactions in Crux reduced distributed network, with applications to the Dissent protocol

CPSC 490 Spring 2015 Proposal
Student: Sage Price
Group Members: Christine Hong, Kojiro Murase
Advisor: Bryan Ford

Summary:
Our goal is to prototype, demonstrate, and experimentally evaluate a general scheme, Crux, to take most any distributed algorithm or system of any kind, and systematically transform it into an equivalent distributed system that offers a locality-preserving property.

Detail:
Current distributed algorithms and systems suffer from their distribution in the form of slow processing or high ping for cross-systemic communication. While this is expected for any form of long-distance communication, such delays are also forced upon users who are near to each other in terms of network topology, due to the need to route messages through an entire network.

To illustrate the issue, suppose one internet user is located in New Haven and trying to communicate with another in Hartford through some distributed system (such as an anonymized messaging system, publish/subscribe service, or Twitter). In this situation, their interaction may be routed through servers in Mexico, Brazil, Australia, and Germany. Though the two users can send TCP or UDP packets with low ping, their use of distributed systems is impacted by the system's large network topology.

Such delays in communication are obviously not ideal, but most high-level scaleable distributed protocols fail to provide a solution to this problem. In our project, we will seek to expand on the Crux scheme, a means of systematically transforming a distributed system into one that preserves locality in communication at the cost of increased per-node overhead (as participation requires action duplication). That is, users who are close to one another topologically will be able to interact through a given complex
distributed system without suffering substantially increased latency over direct one-to-one communication.

Our project will expand on prior research on Crux, as detailed in a draft paper by Michael Nowlan, Jose Faleiro, and Bryan Ford\(^1\). In particular, we will want to implement, prove, and improve the existing experiments in the paper. We will also prototype and demonstrate one or more additional example applications requiring stronger consistency guarantees. Specifically, we would seek to test the viability a distributed system that requires ACID-type database transactions.

To do so, we will model a distributed stock-trading system in distributed instances of a NoSQL database. A key point of such a system for our project is the atomicity necessary for trading systems: for any given item being sold, the system should only process purchases and sales for as many items as are available. Atomicity is difficult to preserve in a distributed system, and the layering and additional operation overhead necessary for Crux complicate maintaining such transactions at a global level. This level of security was not demonstrated in prior iterations of Crux testing, and successful implementation would prove viability as well as helping to further generalize a means by which future reductions of distributed systems can be implemented. Our end goal is to be able to automatically generate multiple clusters/instances of the existing system in a manner which makes sense geographically and thereby maintain locality-preserving transactions without making changes to any algorithms underlying the distributed system we are using.

The Crux scheme is one module of the Dissent\(^2\) project, a larger effort to create a large-scale, decentralized practical anonymous group communication system. Dissent is a joint project between Yale and UT Austin seeking to provide strong, provable security guarantees with good efficiency. It seeks to build on older relay-based anonymous communication systems by expanding on DC-net and verifiable shuffle algorithms to resist timing-analysis attacks (a notable vulnerability of Tor) as well as offering accountable anonymity which would prevent malicious users from flooding the network. Naturally, one goal of Dissent is to do this with reasonable efficiency, and Crux can potentially provide reduced local latency in Dissent’s collective signing, timestamping, and consistency processes.

**Process:**

\(^2\) [http://dedis.cs.yale.edu/dissent/](http://dedis.cs.yale.edu/dissent/)
Because the Crux project will be one element of this semester’s Dissent research, which is itself one module of the larger Yale-UT Austin collaboration, there will be multiple layers of accountability in the development process. Our Dissent team (consisting of Bryan Ford, postgraduate researchers, and undergraduates fulfilling their project requirement) holds twice weekly status meetings. Our smaller Crux team also holds twice weekly meetings, including regular meetings with Jose Faleiro.

Within the Dissent group, we will use a Github-based open-source code review and management process for the Go code underlying Dissent. We will use PlanetLab (or Emulab) to run experiments on different modules of Dissent; this will be particularly critical to testing the Crux scheme. More narrowly, within the Crux team, our project members will define our roles and delegate tasks dynamically; the dependency-heavy nature of our experiments will require concurrent and parallel workstreams.

**Goals of project:**

1. Implement the Crux scheme on a scale-out NoSQL database such as Cassandra, MongoDB, or HyperDex. This will require us to extensively familiarize ourselves with one database system.
2. Experimentally test the validity of the scheme in this context to confirm that the underlying distributed systems algorithms are not affected, while providing negligible impact on latency in topologically near interactions.
3. Automate the systematic transformation of the distributed system’s network in question so as to generate a geographically reasonable arrangement for preserving locality.
4. Prototype and demonstrate the Crux scheme on a distributed ‘stock-trading’ system requiring ACID database transactions.
5. Assess benefits of algorithm to weigh implementation in next build of Dissent in order to provide locally sensitive communication in next-generation cryptographically secure network.

**Deliverables:**

2. Simulation results and analysis showing benefits of algorithm for ACID database transactions.
3. Written report on work involved for implementation of Crux scheme.
4. Determine viability of applying Crux algorithm to Dissent communication protocol.