Priority-Based Synchronization of Distributed Data

Jin, Jiaxi

http://hdl.handle.net/2144/1432

Boston University
Abstract

We consider the general problem of synchronizing the data on two devices using a minimum amount of communication, a core infrastructural requirement for a large variety of distributed systems. Our approach considers the interactive synchronization of prioritized data, where, for example, certain information is more time-sensitive than other information. We propose and analyze a new scheme for efficient priority-based synchronization, which promises benefits over conventional synchronization.

Basic Algorithm: CPIsync

CPIsync is a Characteristic Polynomial Interpolation-based synchronization algorithm that translates the data synchronization problem into rational function interpolation.

- **Step 1:** Generate characteristic polynomial locally (denoted \( P(z) \))
  
  - \( S_A = \{1,2,4,16,21\} \)
  - \( P_A(z) = (z-1)(z-2)(z-4)(z-16)(z-21) \)
  
  - \( S_B = \{1,2,6,21\} \)
  - \( P_B(z) = (z-1)(z-2)(z-6)(z-21) \)

- **Step 2:** Evaluate polynomials on finite field (4 evaluations on \( F_71 \), at here)
  
  - \( P_A(-1) = 69, P_A(-2) = 12 \)
  - \( P_B(-1) = 1, P_B(-2) = 7 \)
  - \( P_A(-3) = 60, P_A(-4) = 61 \)
  - \( P_B(-3) = 60, P_B(-4) = 45 \)

- **Step 3:** Interpolate \( f(z) = P_A(z)P_B(z) \)
  
  - Busy
  - \( f(-1) = 69, f(-2) = 22 \)
  - \( f(-3) = 1, f(-4) = 55 \)

- **Step 4:** All 3 symmetric differences determined & communicated
  
  - \( S_A - S_B = \{4,16\} \)
  - \( S_A - S_B = \{6\} \)

- **CPIsync needs an upper bound on the differences.**
- **The computational complexity is cubic in the bound.**

Divide-and-conquer: Interactive CPI

- **Take constant upper bound** of CPIsync.
- **Recursively run CPIsync** on smaller subsets.
- **Subsets are produced by recursively partitioning the space.**

Partition tree of Interactive CPI

- Using CPIsync with constant upper bound guess 1
  
  - To synchronize two sets differed by \( m \), the overall number of calls to CPIsync is \( O(m \log m) \) with probability at least \( 1-1/m \).

Conduct Prioritized Synchronization

- **ICPI: Interactive CPI**
  
  - **Original set**: elements of priority A, B, or C
  
  - **Subset A**: elements of priority A
  
  - **Subset B**: elements of priority B
  
  - **Subset C**: elements of priority C
  
  - **Proceding flow of a prioritized synchronization**

Experimental result

- **Synchronizing sets with varying size**
  
  - \( n \) is the proportion of synchronized element.
  - **Performance is effectively independent of set size.**

Estimation Modeling

- **Empirical estimate** (below) on likelihood that synchronization completes within given cut-off times, provided training data (above).

Conclusion

We introduced our algorithm to synchronize sets with prioritized data and corresponding data structure. This new scheme supports some desirable features in busy network and/or harsh environment:

- (i) a memory-less fast restart after interruption
- (ii) high-probability communication and computation complexity related to the number of difference only and effectively independent of the set size.

Reference: Jiaxi Jin, Ari Trachtenberg and David Starobinski, Priority-Oriented Interactive Set Reconciliation, submitted.