

# Implementasi Protokol Komunikasi RPC Chain pada Sistem Kuorum dengan Raft

Ivan Ryansyach<sup>1</sup>, Kemas Rahmat Saleh Wiharja<sup>2</sup>

<sup>1,2</sup>Fakultas Informatika, Universitas Telkom, Bandung

[iryansyach@student.telkomuniversity.ac.id](mailto:iryansyach@student.telkomuniversity.ac.id), [bagindokemas@telkomuniversity.ac.id](mailto:bagindokemas@telkomuniversity.ac.id)

---

## Abstrak

Sebuah sistem replikasi terdistribusi dapat menyediakan kecepatan baca tinggi dengan menggunakan beberapa node yang ada dalam kluster — namun masih terdapat masalah fundamental, kecepatan tulis hanya akan secepat kemampuan satu buah mesin. Salah satu implementasi sistem tersebut adalah Raft, dimana overhead pada leader adalah salah satu faktor penghambat. Seperti yang diketahui, protokol komunikasi broadcast memerlukan resource yang signifikan untuk memproses request dengan volume yang tinggi. Pada studi ini, kami mengusulkan RPC chaining, sebuah metode invokasi request secara terus-menerus tanpa memerlukan keterlibatan client di setiap hop. Kami mengimplementasikan sistem tersebut pada sebuah environment tersimulasi menggunakan sebuah framework bernama DepFast. Yang kemudian kami uji pada konfigurasi deployment yang umum digunakan serta dapat mencapai sekitar 15-30% throughput yang lebih tinggi pada sistem generasi sebelumnya. Walaupun demikian, masih memungkinkan untuk mencapai throughput yang jauh lebih besar dengan mengoptimalkan sistem yang lebih tinggi.

**Kata kunci: Raft, Broadcast, Chain, RPC**

---

## Abstract

A distributed replication system can provide high read throughput through the use of multiple nodes in the cluster — yet a more fundamental problem remains, write throughput is capped at the speed of a single machine. One such system is Raft, where leader overhead is one of the limiting factors. The underlying communication protocol, broadcast, is known to require significant resources to process high volume of requests. In this study we propose RPC chaining, a continually flowing request invocation method without involving the client during each hop. We build our implementation in a simulated environment with a framework called DepFast. Which are then tested on widely used deployment setup and are able to achieve 15-30% higher throughput on commodity hardware. However, it is still possible to achieve even more throughput using higher end systems.

**Keywords: Raft, Broadcast, Chain, RPC**

---

## 1. Pendahuluan

Sistem kuorum adalah sistem replikasi terdistribusi dengan kemampuan melakukan konsensus. Konsensus memastikan konsistensi dapat dicapai setiap node yang ada, dengan hanya melakukan perubahan entri yang disetujui oleh kuorum (mayoritas node). Algoritma konsensus menggunakan sebuah log untuk mengelola riwayat konsensus tersebut dan memastikan entri log direplikasi di setiap node. Salah satu implementasi yang populer adalah Raft, dimana toleransi node gagal hingga sebanyak total minoritas node, serta memiliki performa komparatif dengan Paxos (implementasi algoritma serupa). Dalam sebuah kluster sistem replikasi, biasanya akan terdapat dua jenis node: leader dan follower. Dalam konteks sistem kuorum, hanya leader yang diijinkan melakukan perubahan ke dirinya maupun replikasi ke follower.