

Pandit Extended Abstract

Eric Gerard Moynihan

March 2022

1 Introduction

Pandit at its core is a **distributed service level proxy**. This means it is a proxy tasked with interpreting data on a service level and caching it in a distributed manner, allowing it to speed up read and write times of clients querying the services in question.

Pandit is designed to be able to proxy any data source such as REST APIs or SQL databases. It reads in user defined Protocol Buffers [?] that are extended to provide information on how to parse, cache and return data to clients.

It uses these protobufs to provide a gRPC [?] service that will be available to every client of every node.

A single instance of the Pandit daemon will be ran on each host in a cluster of machines. When a client queries a service, the daemon will first check the cache for the data. If it cannot find it, it will delegate the request to an authoritative container in the cluster.

Since the mapping between the data returned by the authoritative container and the data returned by the gRPC service is user-defined, Pandit provides options to annotate specific fields in return types to allow for the efficient caching of common requests, greatly reducing the amount of queries that need to be sent to the authoritative container.

2 Existing Solutions

Service level proxys usually fall into two categories:

References

- [1] Profocol Buffers documentation - [developers.google.com](https://developers.google.com/protocol-buffers/).
- [2] gRPC documentation - grpc.io.