

RoadMap

Here gather the long-term plans of Pegasus.

Objectives

Shrink the size of Pegasus kernel

Pegasus targets to be a simple, robust key-value database with stable&predictable latency. To achieve the performance requirement, we use C++ for the "kernel", i.e MetaServer and ReplicaServer.

However, C++ is notorious for its inefficiency (in development) and high learning curve, which eventually leads to a high contribution barrier.

We expect the C++ codebase, including the functionalities of Pegasus kernel, can be simpler, smaller, and more stable.

Build an ecosystem around Pegasus

As Pegasus grows, we have learned that it's impossible to maintain a monolithic system bundled with sophisticated features. So we need an ecosystem with a variety of useful tools, controllers, surrounding the kernel and altogether manages the clusters more intelligently.

Improve the stability of Pegasus

Performance is usually the main reason people choose Pegasus,

Key Results

- ☐ admin-cli: use golang to build a more user-friendly administration command-line tool.
- ☐ meta-proxy: a stateless proxy in front of MetaServer that lets the users unaware of the specific address of MetaServer.
- ☐ pegasus-cpp-client: maintain the client as a separate project.
- ☐ backup-manager: a service that manages backup externally.
- ☐ bulkload-manager: a service that manages bulk-load externally.
- ☐ pegic: a command-line tool that reads pegasus data in a user-friendly way.