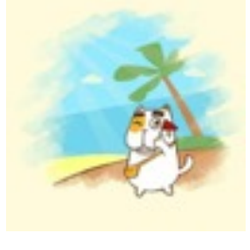
Pard: Parallel database running like a Leopard

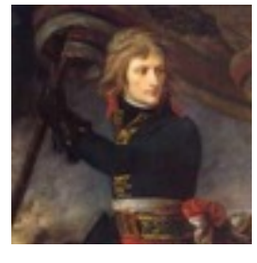
Pard Team

https://github.com/dbiir/pard

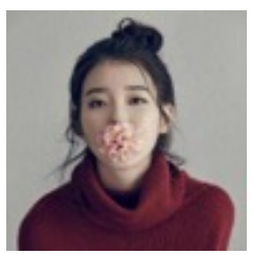
Chen Cheng
@withchencheng



Han Han
@hagen666



Huang Wentao
@huangwentao0831

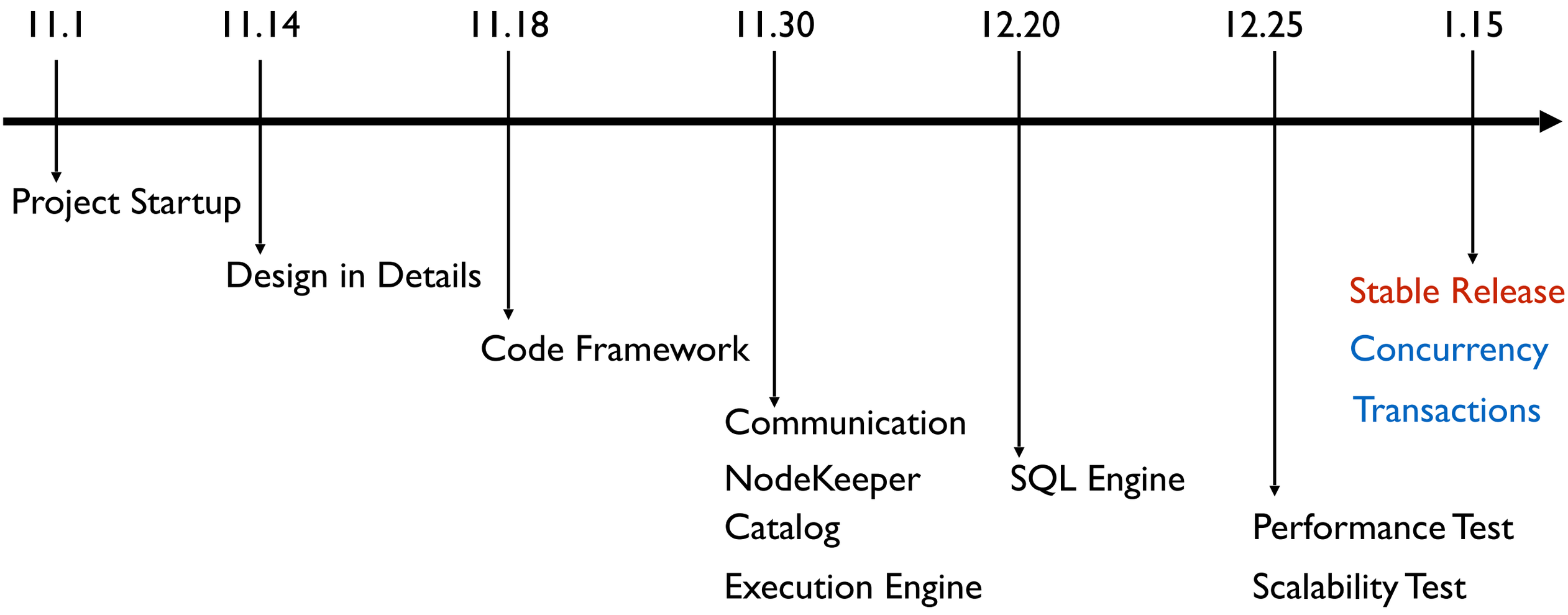


Han Xueran
@lemontreehxr

Jin Guodong
@ray6080

Shao Mingrui
@crazyxuehu

Pard Timeline

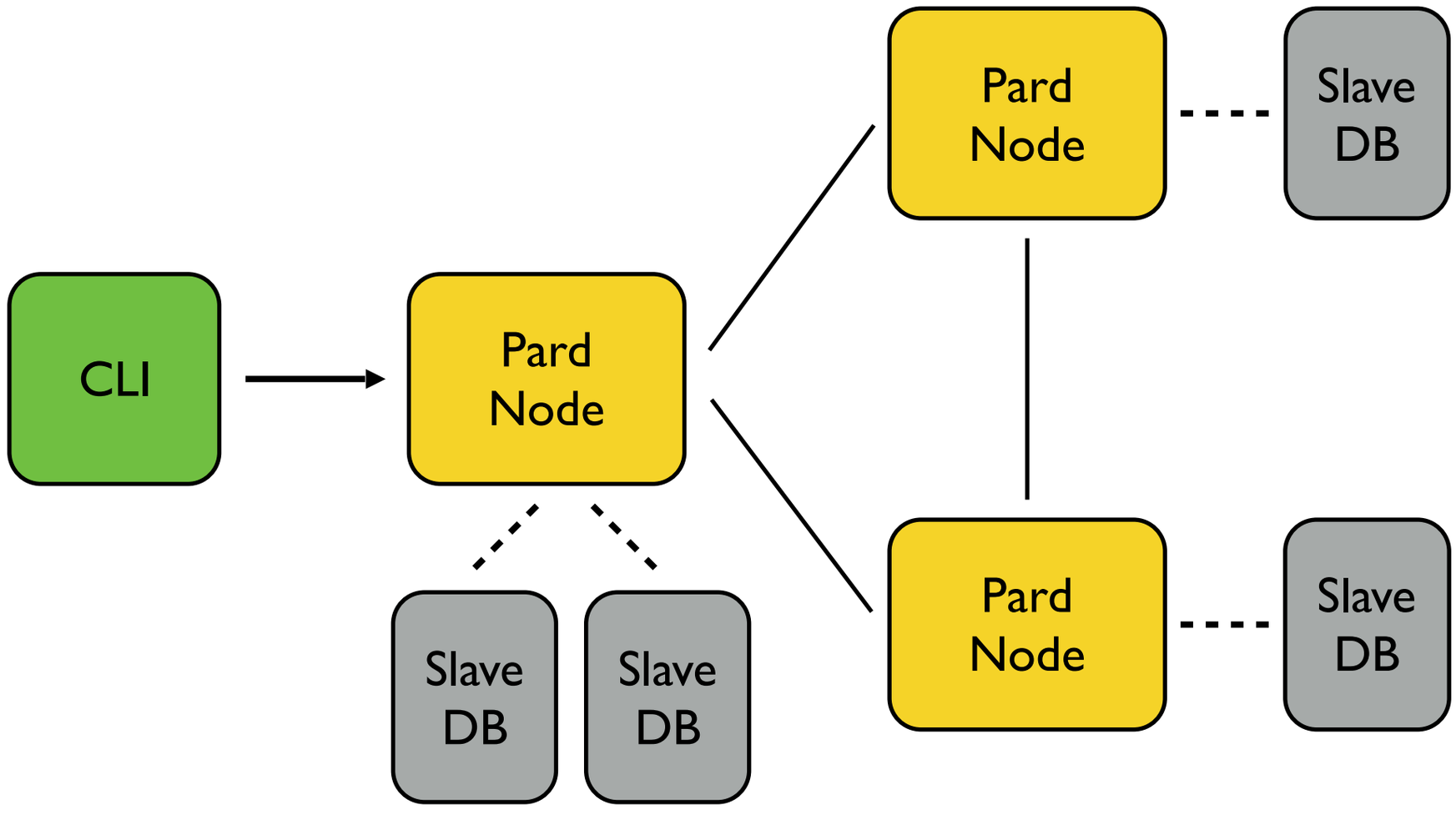


Outline

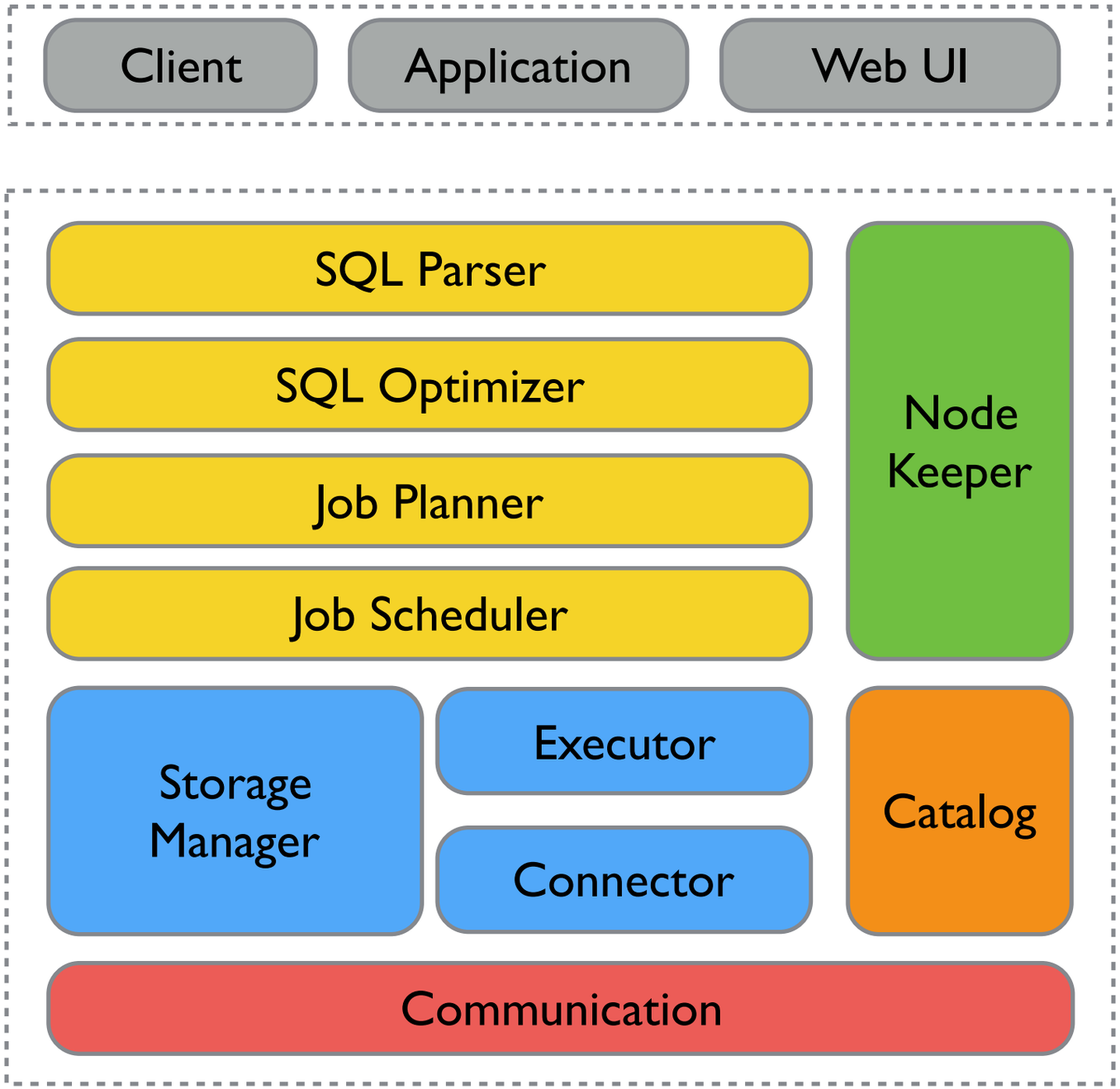
- Architecture
- SQL Engine
- Execution Engine
- Communication
- Catalog
- Project Dependency Management

Architecture

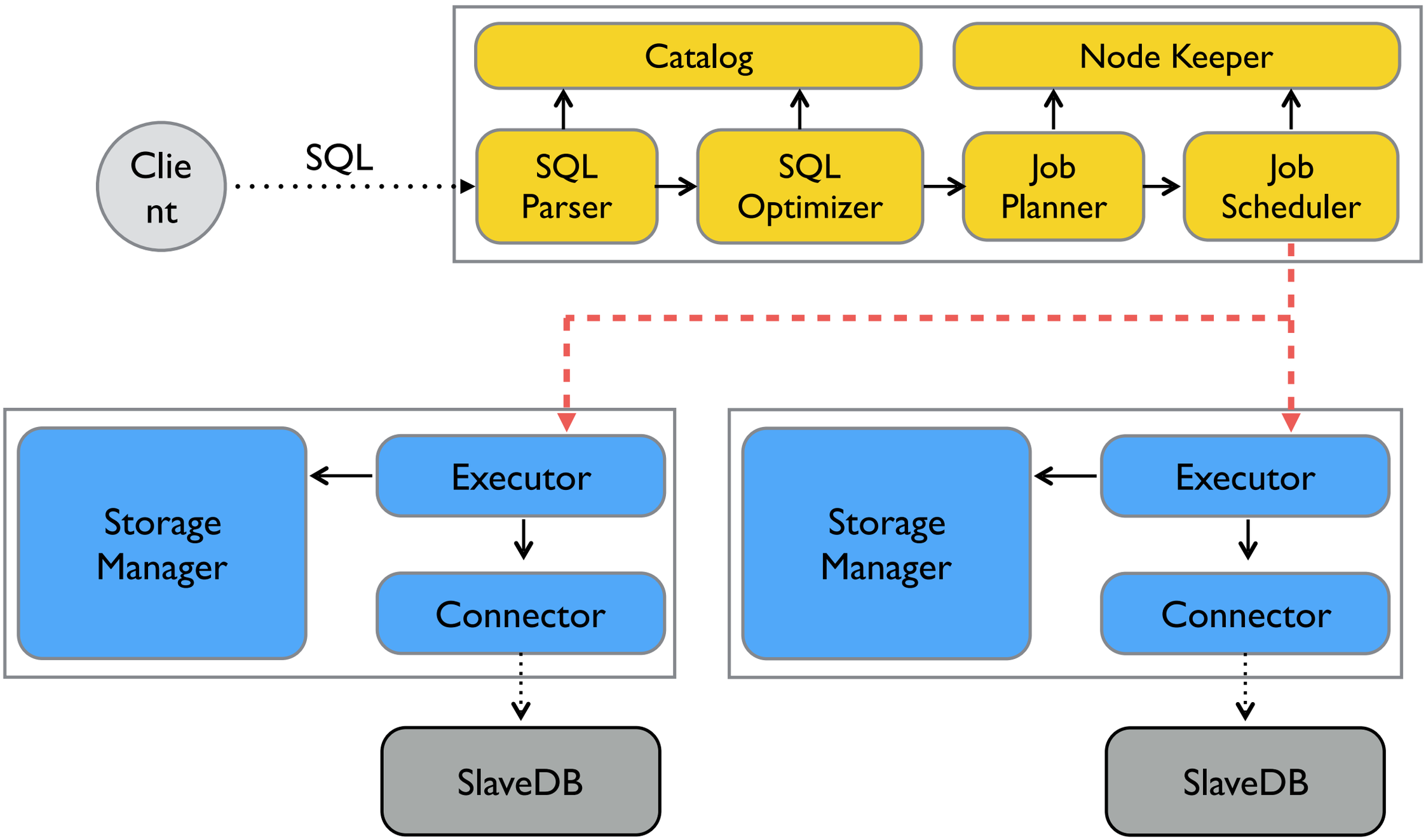
Pard Architecture



Pard Node Internals



Pard Execution Flow



SQL Engine

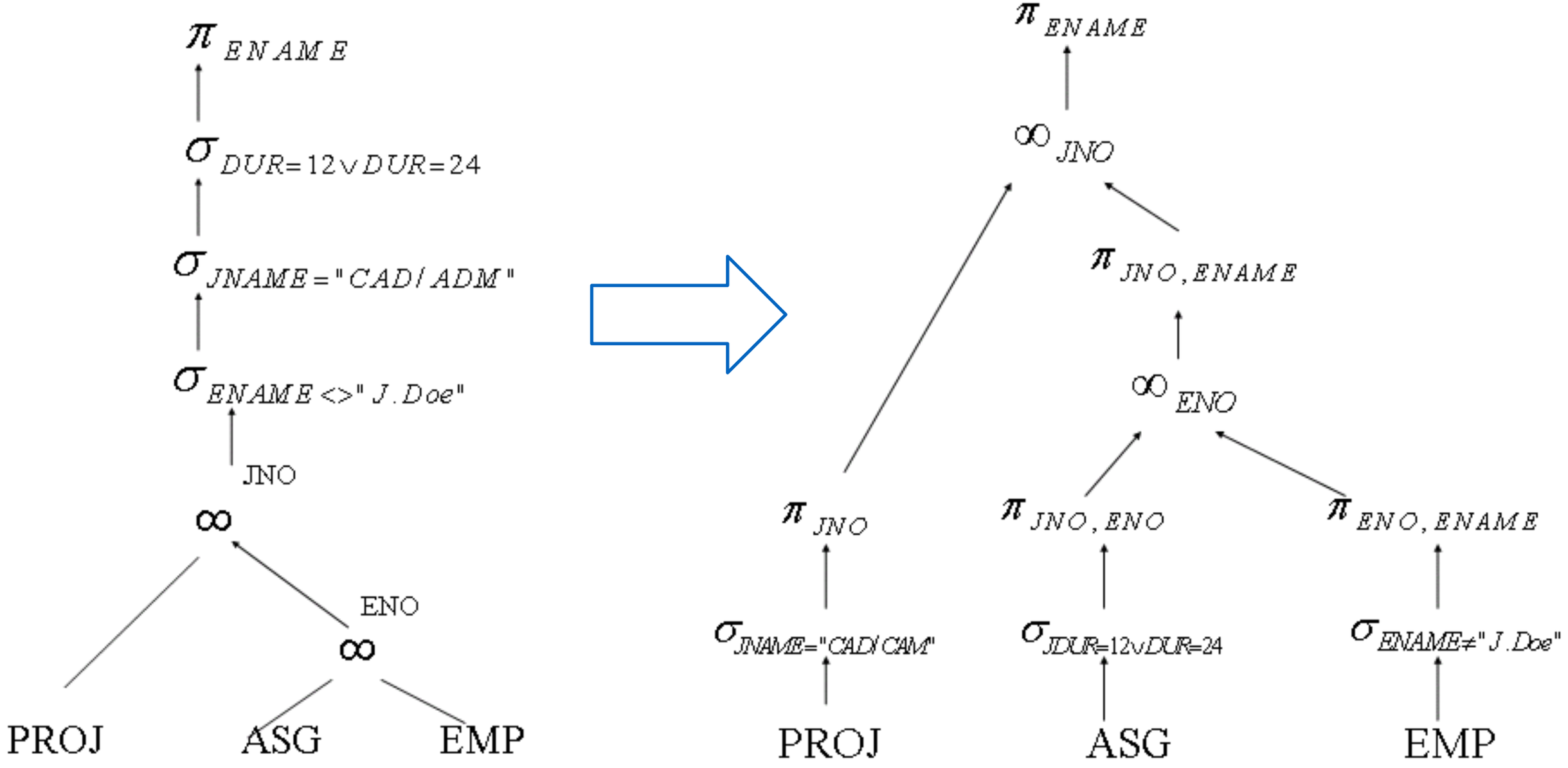
Pard SQL Optimizer

Query Rewriting

Query Localization

Cost Model

Query Rewriting



Query Localisation

Horizontal Rules

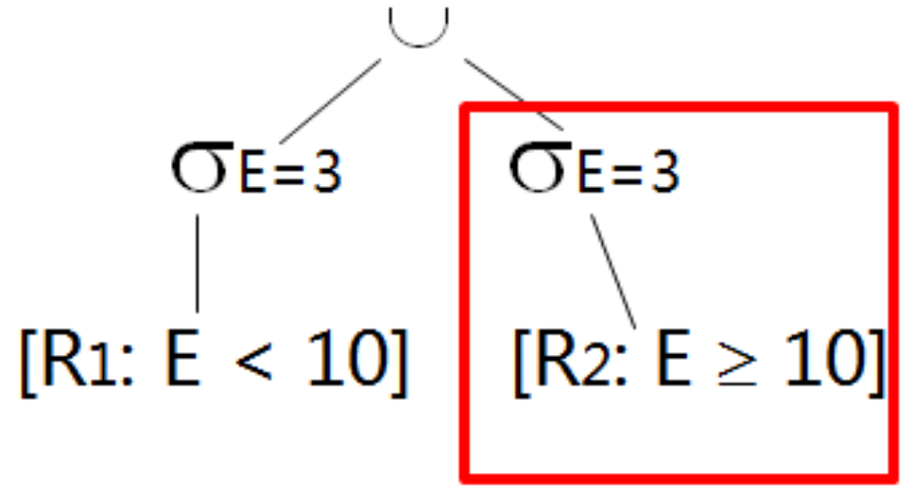
- Eliminate useless selections
- Eliminate useless joins

Vertical Rules

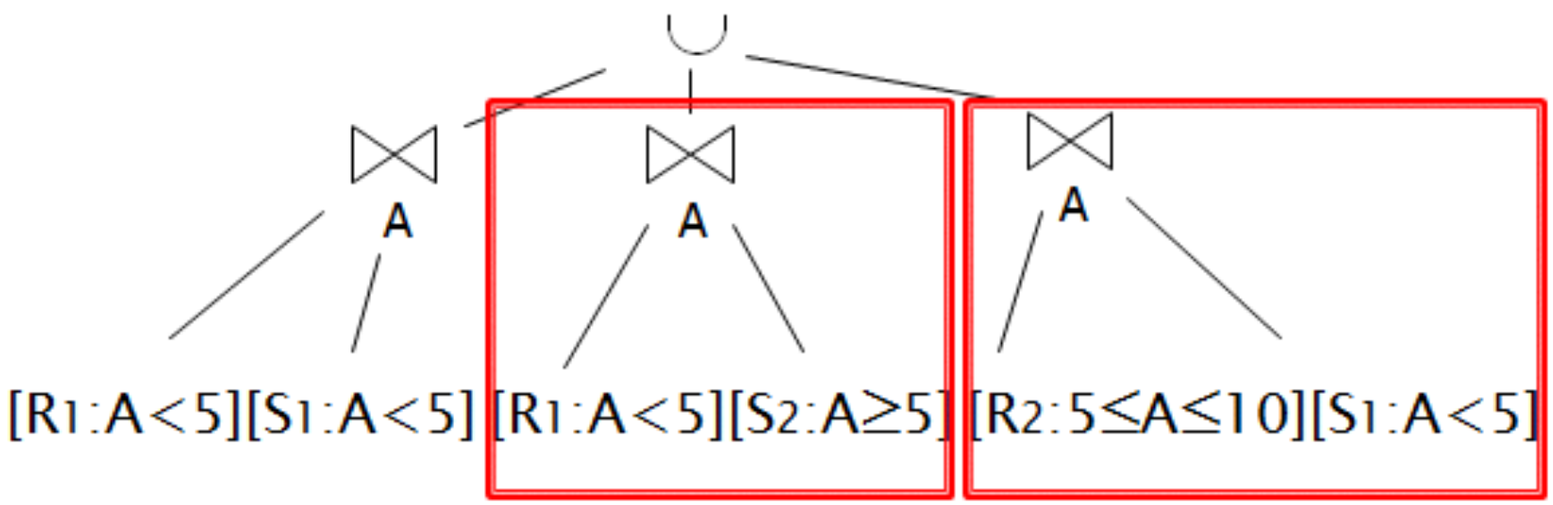
- Eliminate useless joins

Query Localisation Examples

Eliminate useless selections



Eliminate useless selections



Cost Model

Network Transportation Cost

Repartition Cost when join

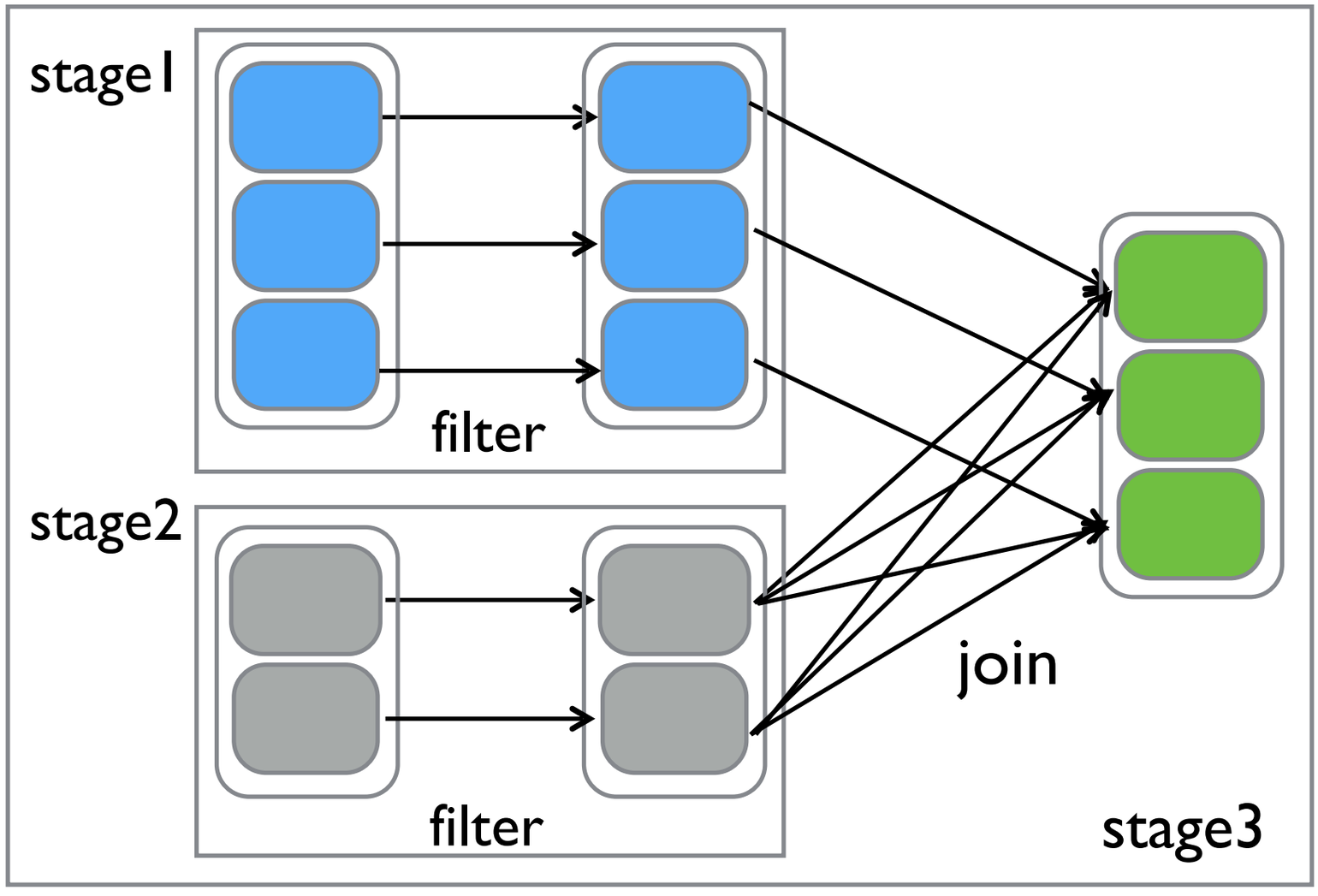
Join Cost (Network & Data Distribution)

- Partition Join
- Semi-Join
- Join Order

Sites Performance

- CPU cores
- Memory Size
- IO

Pard Job Planner



Pard Job Scheduler

Job Management

- Job Scheduler
- Job Status Monitor
- Job Result Collection

Task Management

- Task Distribution
- Task Status Monitor

Execution Engine

Pard Execution Engine

Local Database Connector

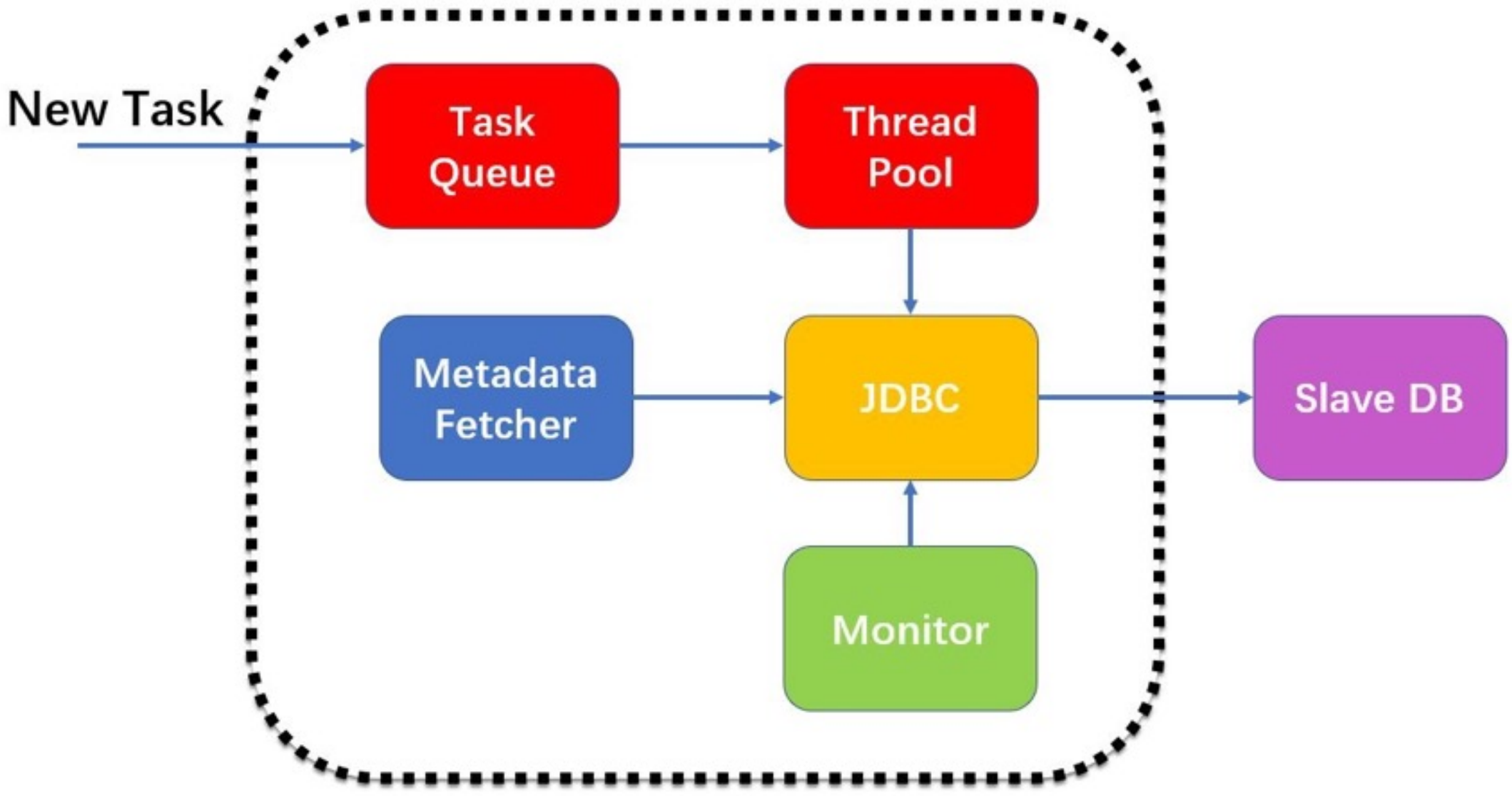
Metadata fetcher

SQL Translator

Task Queue Management

Storage Manager

Execution Engine Design



Local Database Connector

JDBC (PostregSQL, MySQL)

Monitor

Metadata Fetcher

Number of Distinct Values

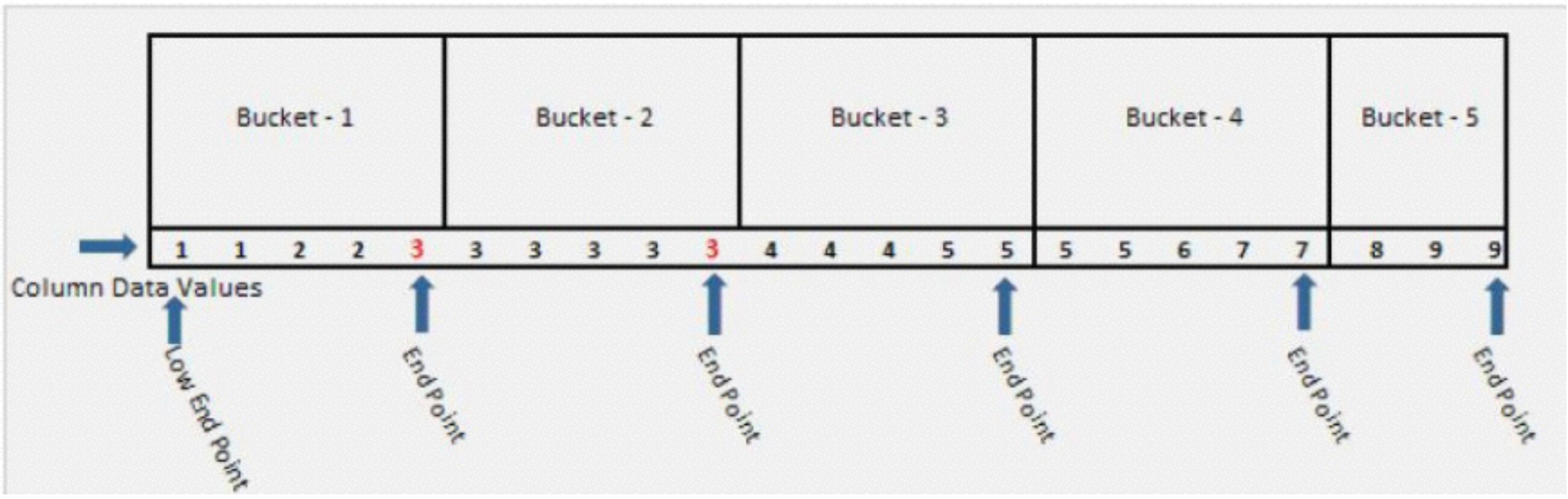
Max Value/Min Value

Number of Null

Average Column Length

Histogram (Frequency Histogram, Height Balanced Histogram)

Height Balanced Histogram



Height Balanced Histogram

Bucket Number	Endpoint
0	1
1	3
2	3
3	5
4	7
5	9

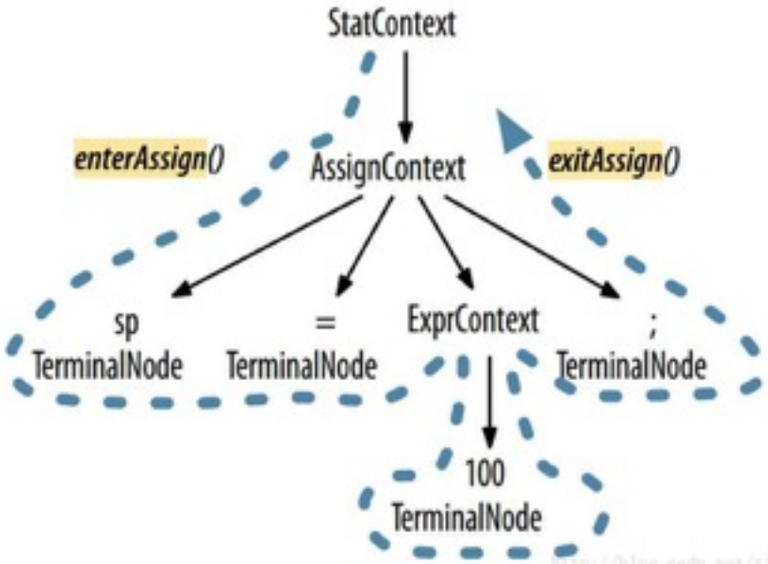
Bucket Number	Endpoint
0	1
2	3
3	5
4	7
5	9

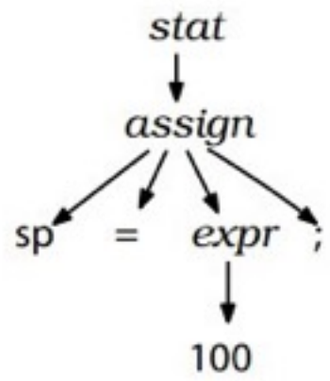
SQL Translator

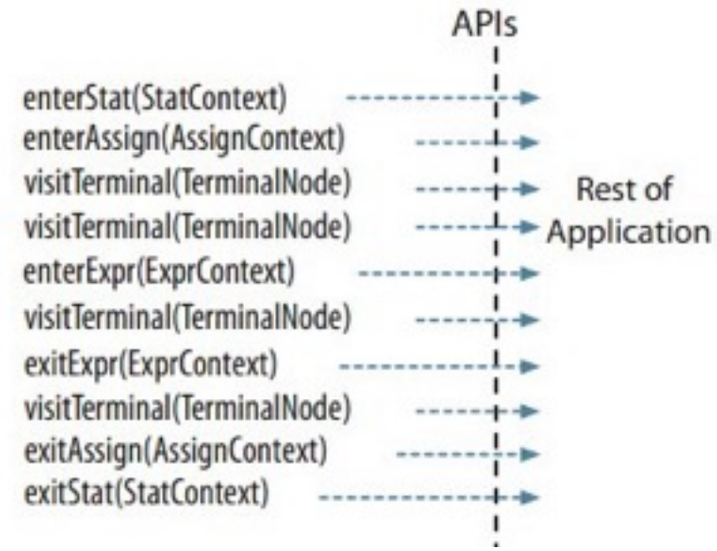
Traversing an Operator Tree to obtain a SQL statement.

- JSqlParser
- Antlr
- Druid

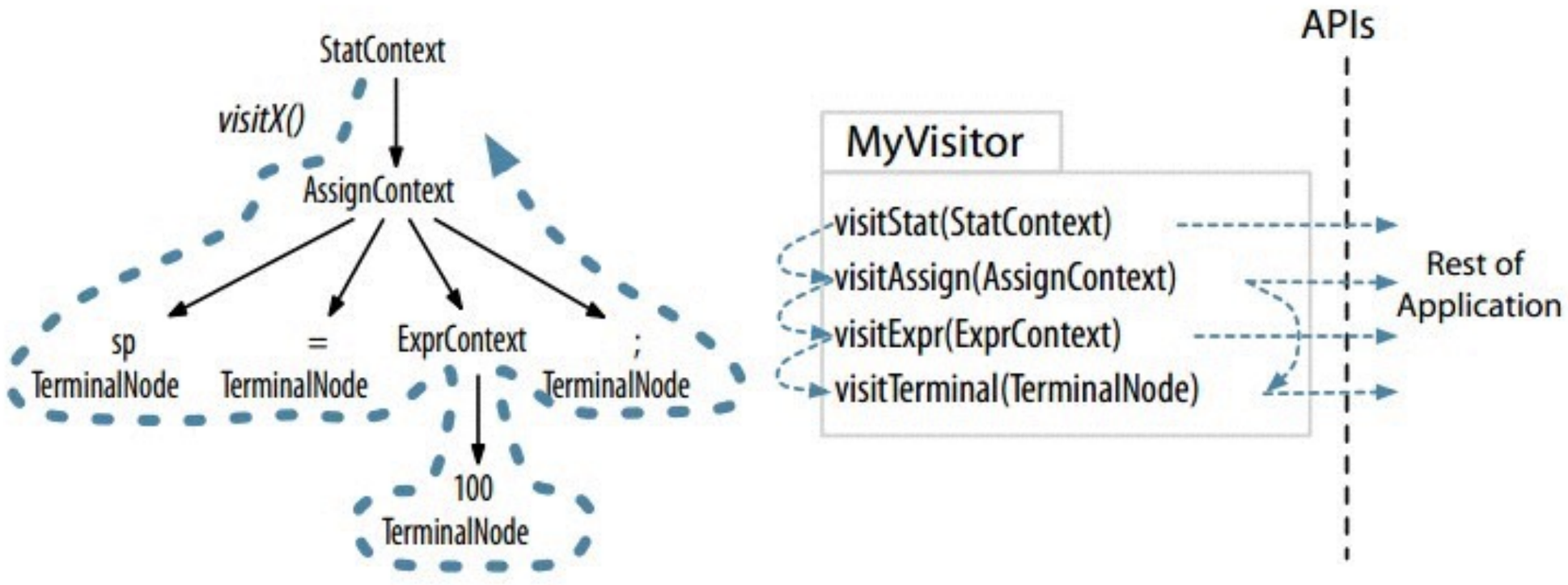
Listener && Parse-Tree-Walker







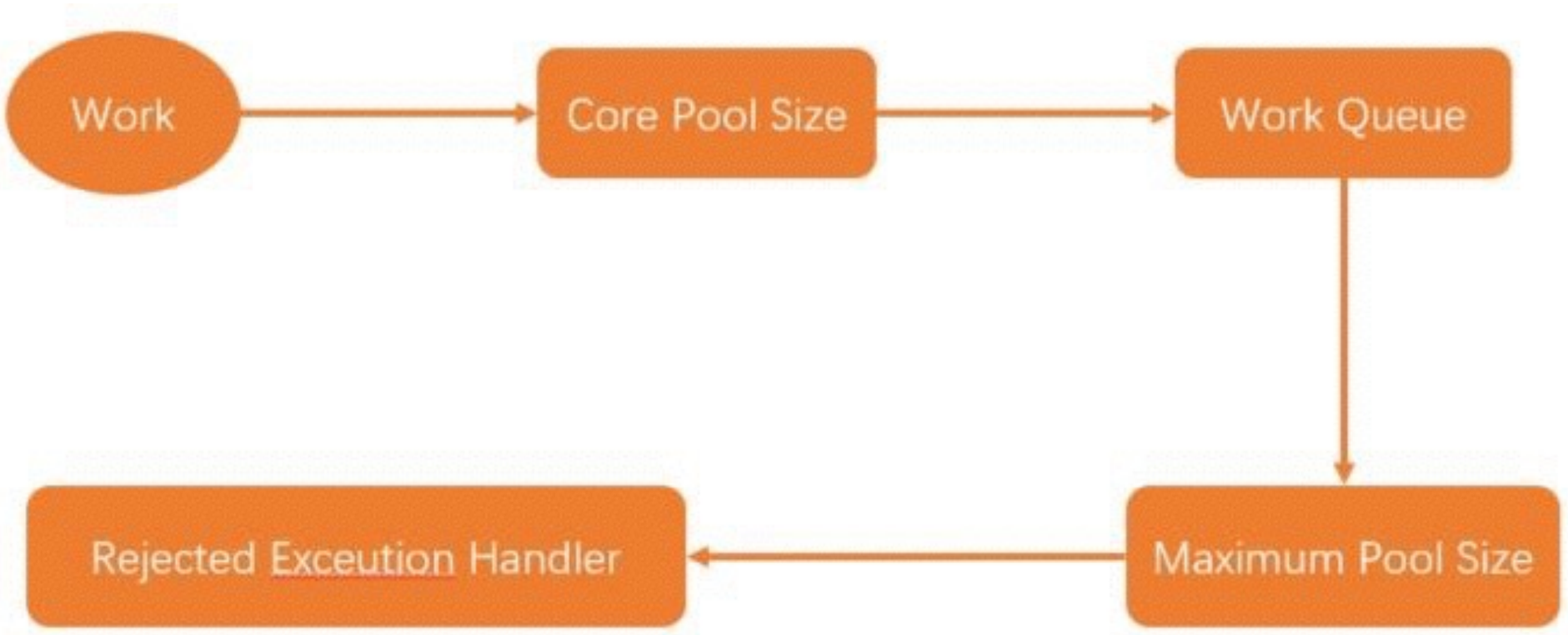
Parse-Tree-Visitor



Task Queue Management

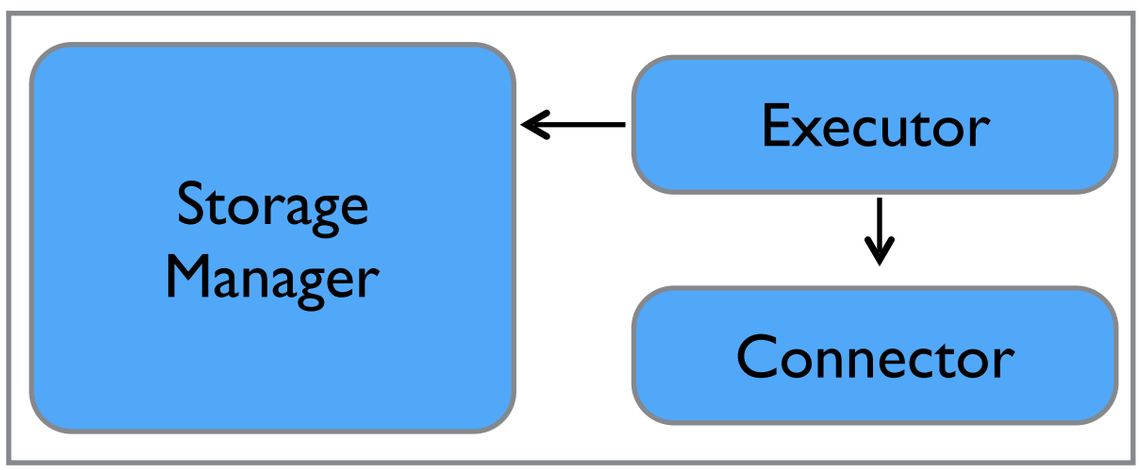
Thread Pool (Executor Service, ThreadPool)

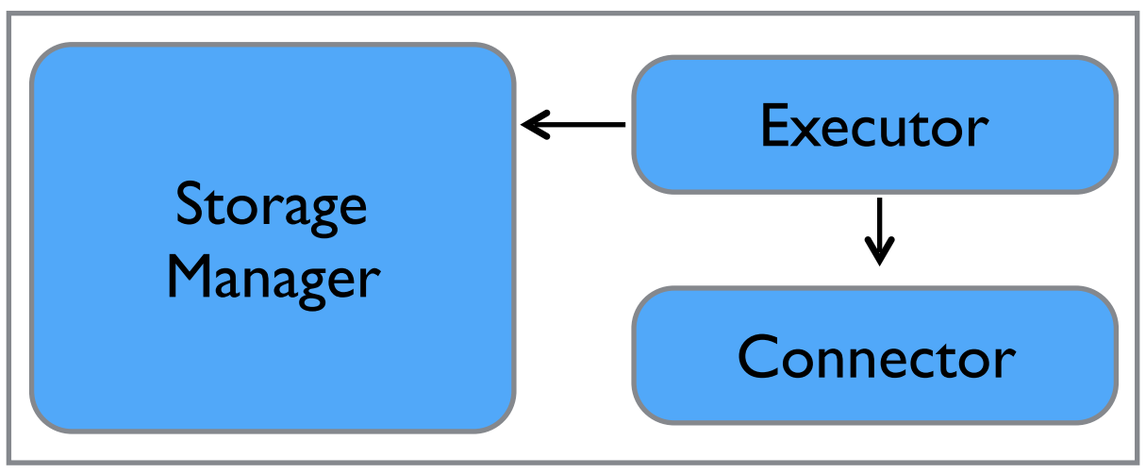
FIFO



Storage Manager

- column store
  - Data compression
  - Projection, Join
- data format
  - byte array
  - schema





Communication

Communication

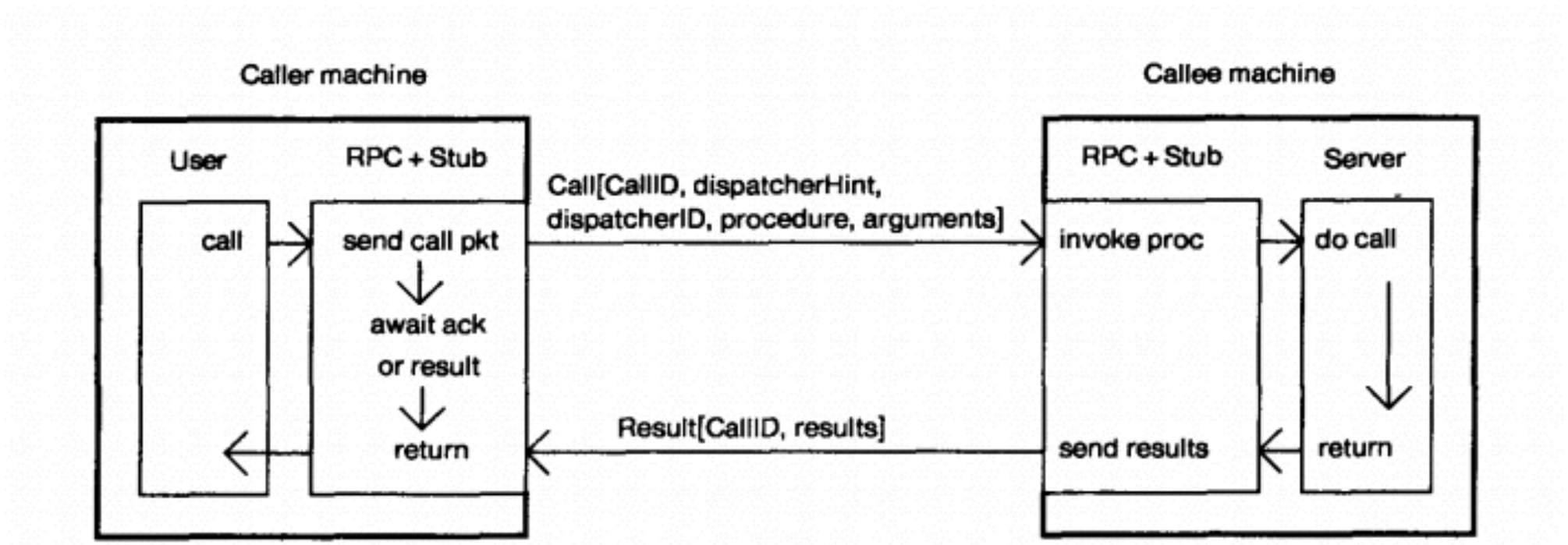
RPC（gRPC）

Data transfer（Netty）

client-server communication（socket）

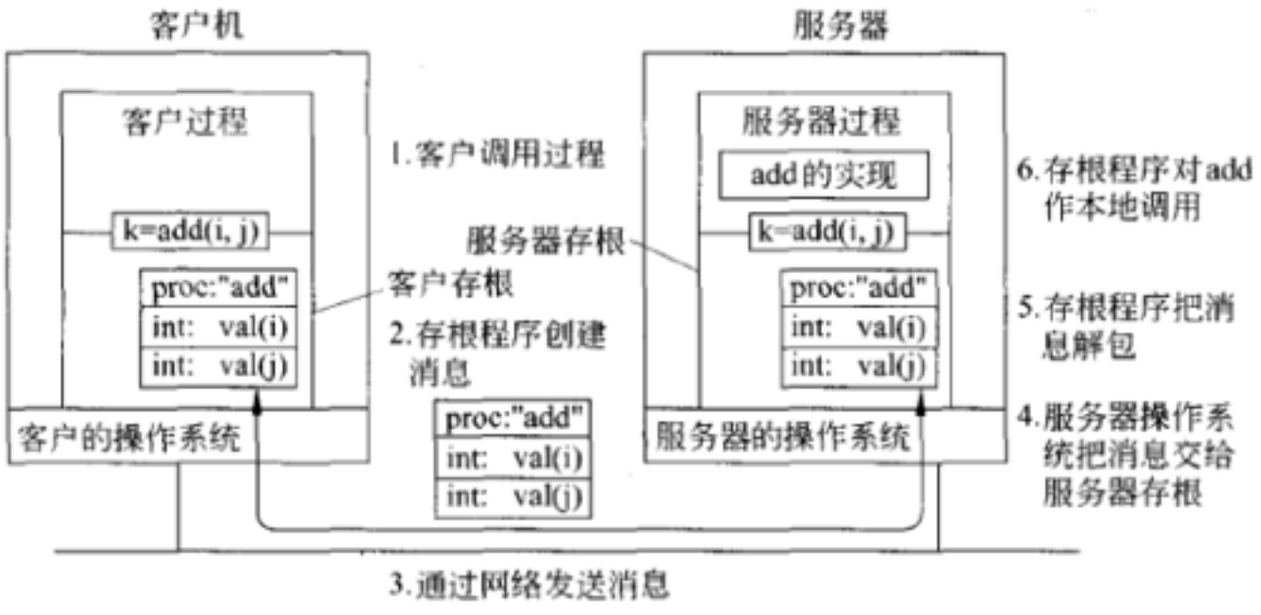
RPC

RPC （Remote Procedure Call）

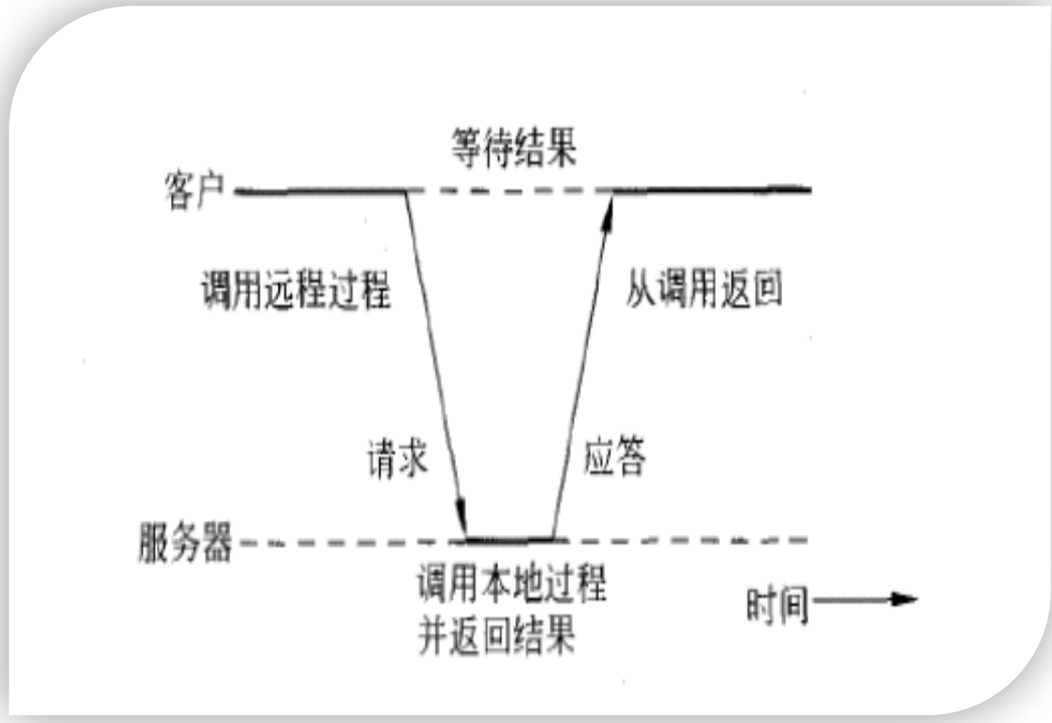


RPC Workflow

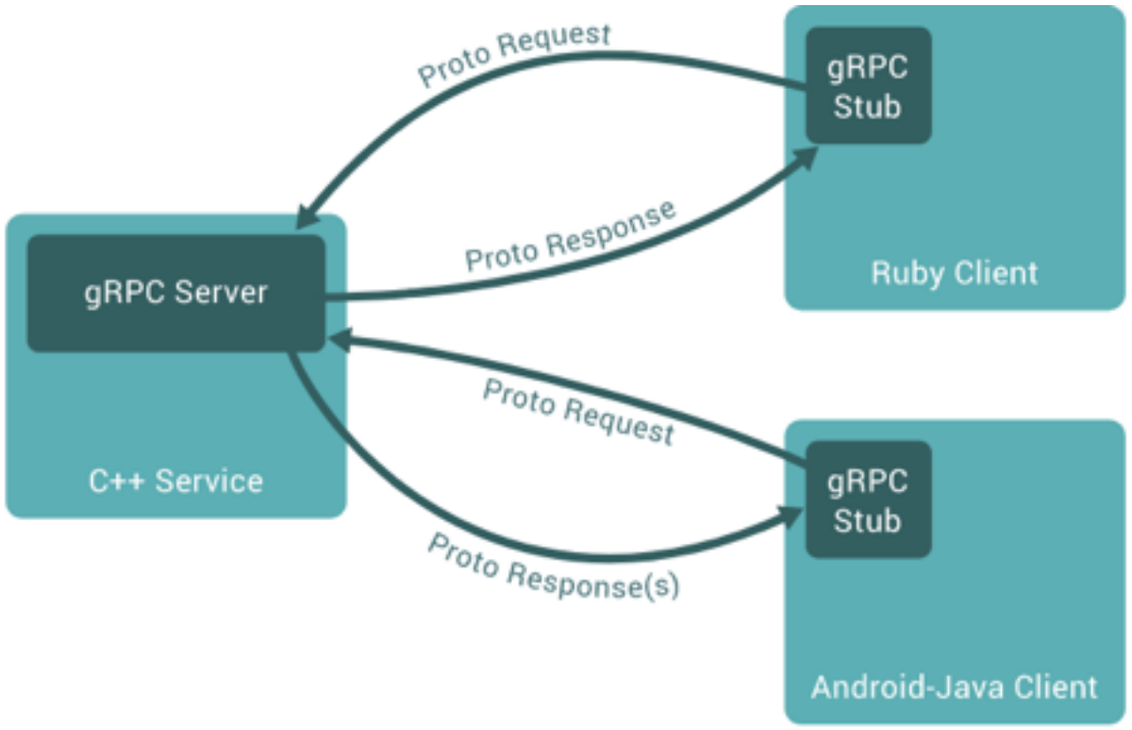
Remote Addition Call :



Overview:



gRPC



- (small quantity)Data Transfer： Google Protocol Buffers
- gRPC lets you define four kinds of service method:
  ①Unary RPC
  ②Server streaming RPCs
  ③Client streaming RPC
  ④Bidirectional streaming RPC

```
rpc LotsOfReplies(HelloRequest) returns (stream HelloR
}
```

- Synchronous vs. asynchronous

grpc / grpc Watch 842 Star 12,179 Fork 2,585

yashykt Merge pull request #13147 from yashykt/testc++ize ... Latest commit d9da738 6 minutes ago

Data Transfer

Task: bulk data transfer
Option: Netty

最近一个圈内朋友通过私信告诉我，通过使用Netty4 + Thrift压缩二进制编解码技术，他们实现了10W TPS（1K的复杂POJO对象）的跨节点远程服务调用。相比于传统基于Java序列化+BIO（同步阻塞IO）的通信框架，性能提升了8倍多。

What is Netty

netty / netty

Watch 1,410 Star 11,690 Fork 5,528

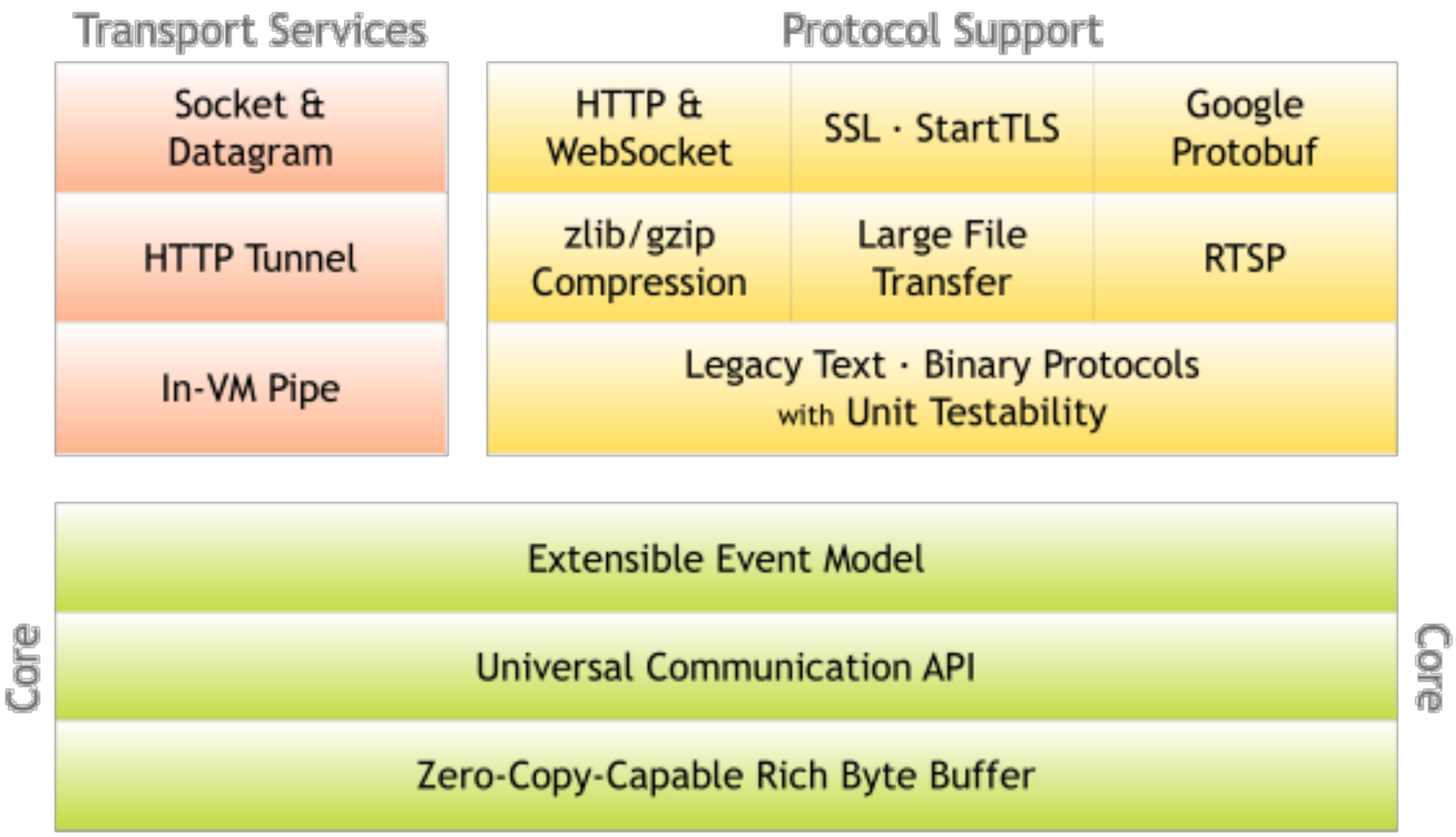
IO

cat Files > Networking

Find file Clone or download

Latest commit 7511c15 7 hours ago

Data Transfer - Using Netty



Catalog

Catalog

What is the catalog?

- the meta data of the distributed database including
- fragmentation
- data distribution
- data replication
- meta data

How to mange the catalog?

- centralized、fully replicated、partitioned(partially replicated)

ETCD

- we use the ETCD to manage the catalog to fully replicated the catalog to every site.

etcd

etcd is a distributed key value store that provides a reliable way to store data across a cluster of machines. It's open-source and available on GitHub. etcd gracefully handles leader elections during network partitions and will tolerate machine failure, including the leader.

Project Dependency Management

Phase I	Phase II	Phase III	Phase IV	Phase V
Interfaces Design Between Modules	Auto Package & Deployment	Test Environment Setup	Module Unit Tests	
Connector Design NodeKeeper	Connector Implementation		NodeKeeper Implementation	
SQL Parser AST & Operators	Cost Model & Planner Design	Optimizer & Planner Implementation	Optimizer & Scheduler Implementation	V 1.0 ALPHA TEST
Catalog Design		Catalog Implementation		
RPC Messages List		Communication Implementation		
Memory Block Design	Storage Manager Implementation	Executor Implementation	Executor Implementation	
Due: 11.17	Due: 11.24	Due: 12.08	Due: 12.15	Due: 12.22

References

- PPT P11 https://github.com/fanju1984/ddb/blob/master/Fall%202017/Slides/05.%20Query%20Optimization-2.pptx 28
- PPT P13 https://github.com/fanju1984/ddb/blob/master/Fall%202017/Slides/05.%20Query%20Optimization-2.pptx 34
- PPT P13 https://github.com/fanju1984/ddb/blob/master/Fall%202017/Slides/05.%20Query%20Optimization-2.pptx 44

Thanks