Manage and Scale Hundreds of DBs with Open Source Hera Proxy

Kenneth Kang & Varun Sankar • May 2020 NoCOUG

Topics

High Efficiency Reliable Access (HERA)

About Kenneth Kang & Varun Sankar

Kenneth Kang

- Joined Hera Team in 2015
- Go and C++ Systems Architect
- Previously at GlaxoSmithKline

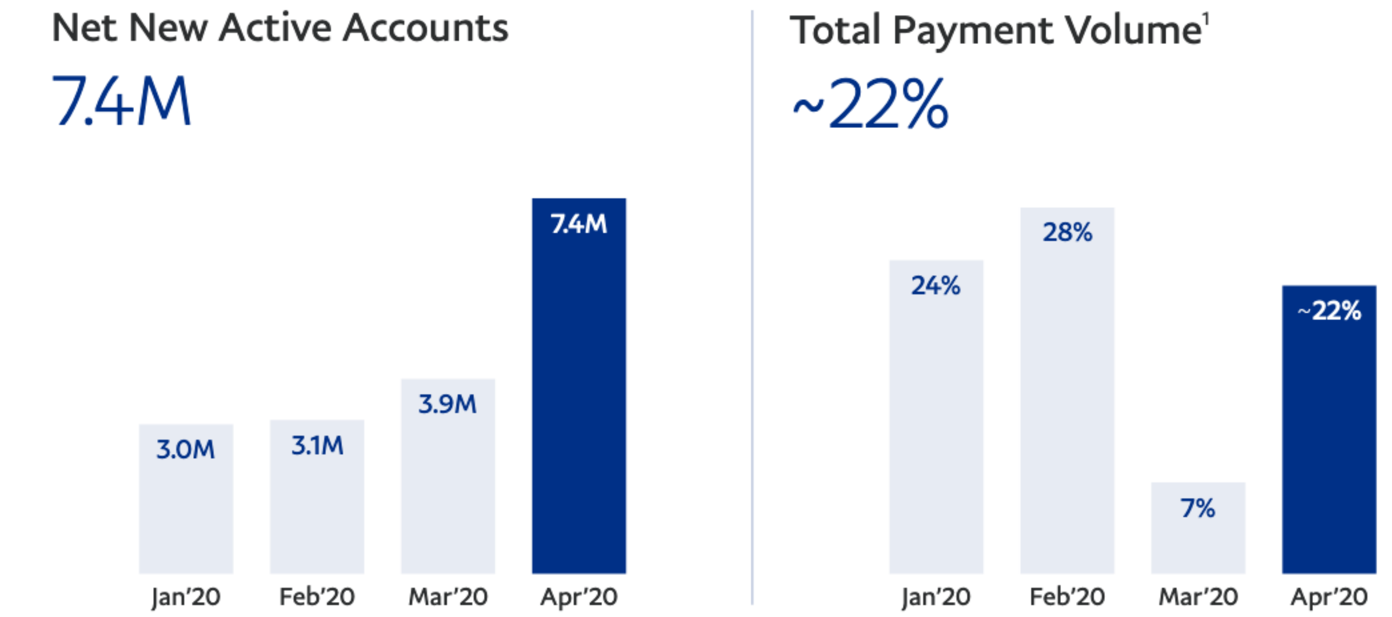
Varun Sankar

- Joined Hera Team in 2018
- Go and C++ Programmer
- Previously at HCL, Cisco, AMD, Aricent

About PayPal

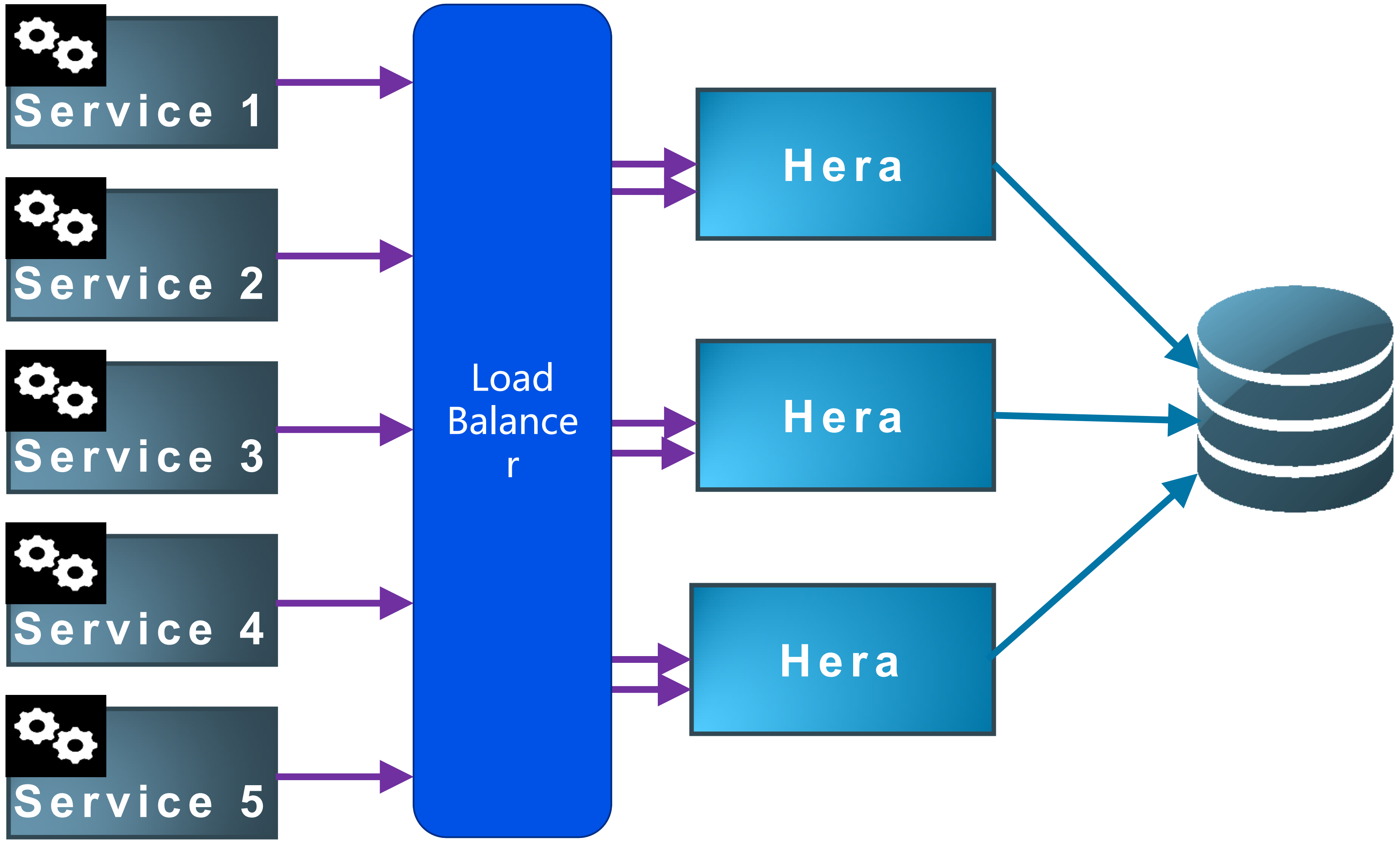
- Q1'20 : 18% TPV growth
- April '20 ~1.2 billion payment transactions

- Hundreds of Live DBs
- Adding more each quarter
- **May 1 '20 was largest transaction day in our history**



1. FX-neutral growth rates

About Hera



At its core:
Connection Multiplexer

Open Sourced Mid-2019:
github.com/paypal/hera

Hera History

1998 Open Source as OCC

2006 Unsuccessful Client Side Sharding

2012 Multiplexing

2016 Sharding

2018 Switch to Go Programming Language

2019 Open Source as Hera

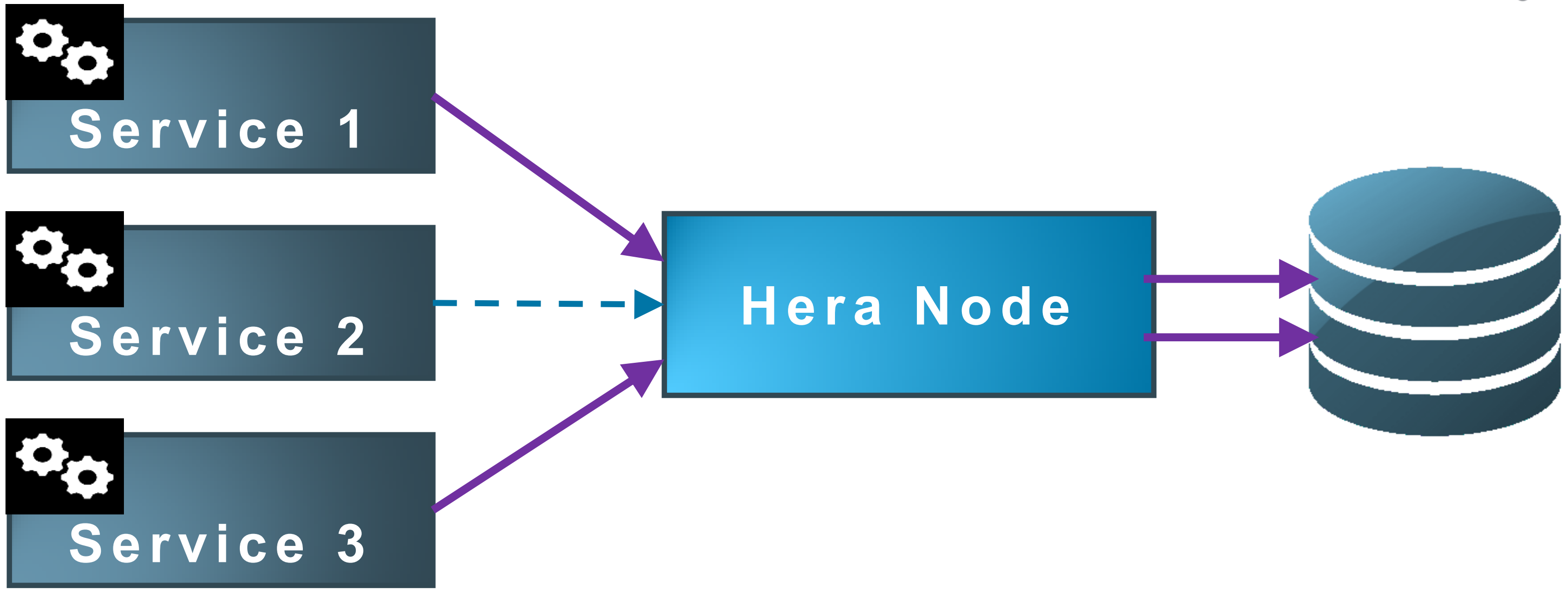
Features

Multiplexing
SQL & Bind Eviction
Sharding

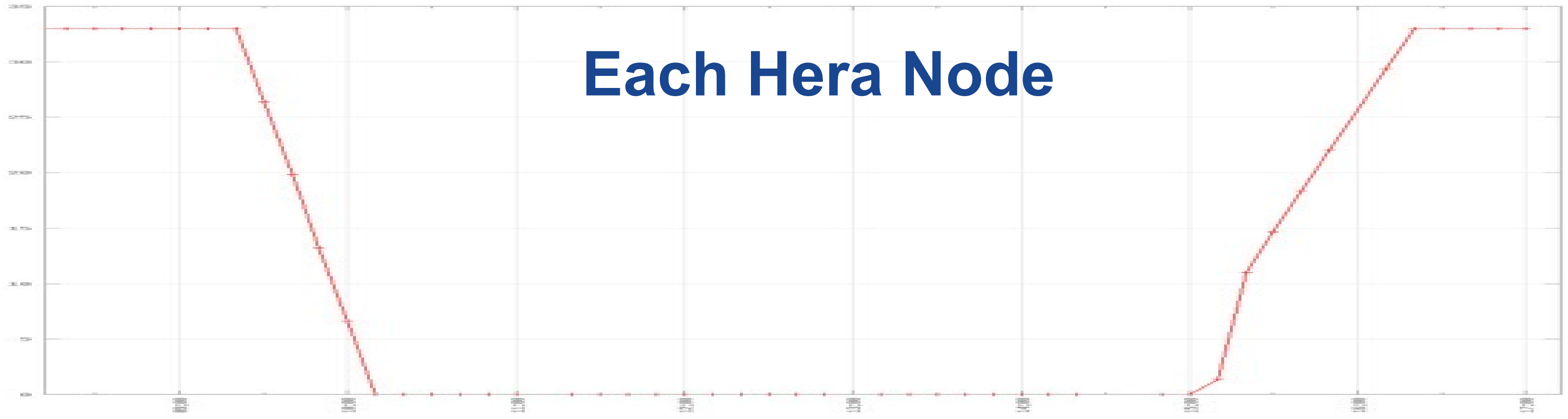
Read Replica Query Failover – Transparently uses two read replicas for higher availability
Read-Write Split – Reduces interconnect traffic in Oracle RAC
MySQL

Managing and Scaling OLTP

Monitoring Usage

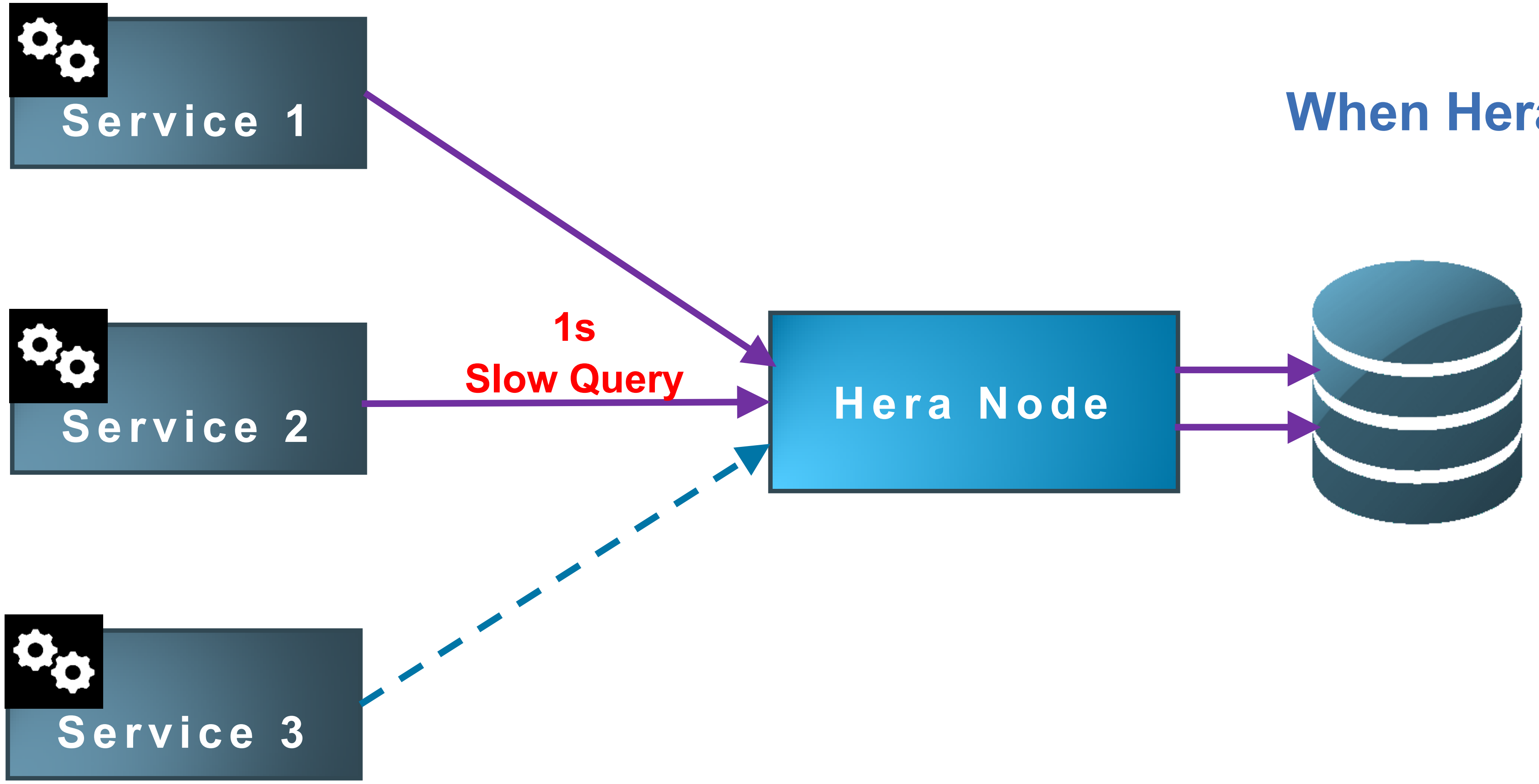


Free DB Connections (acpt)



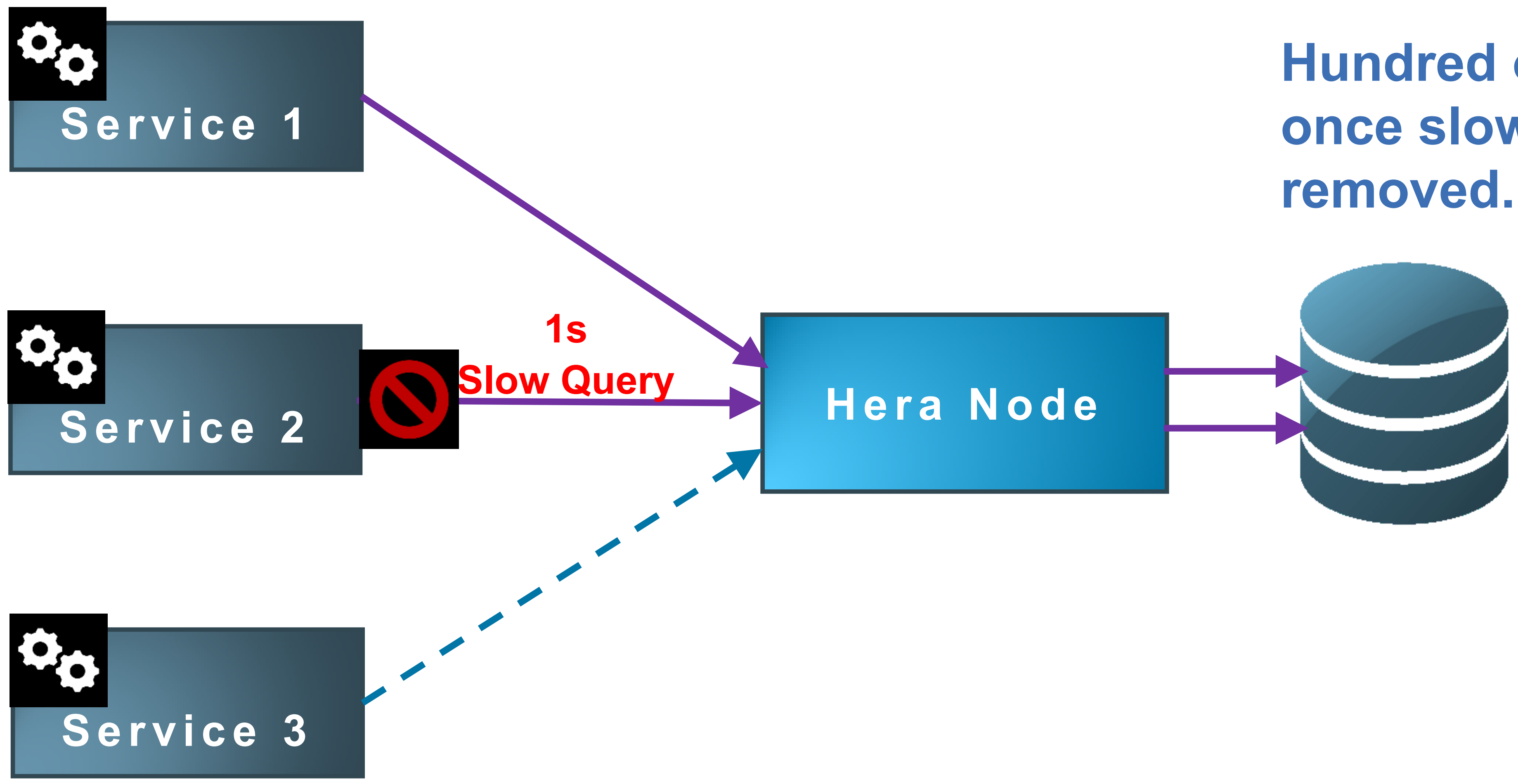
Time

Managing Overload:Slow Query Eviction



When Hera is overloaded

Resiliency: Slow Query Eviction



Hundred of queries can run once slow query is removed.

Bind Eviction

Previously

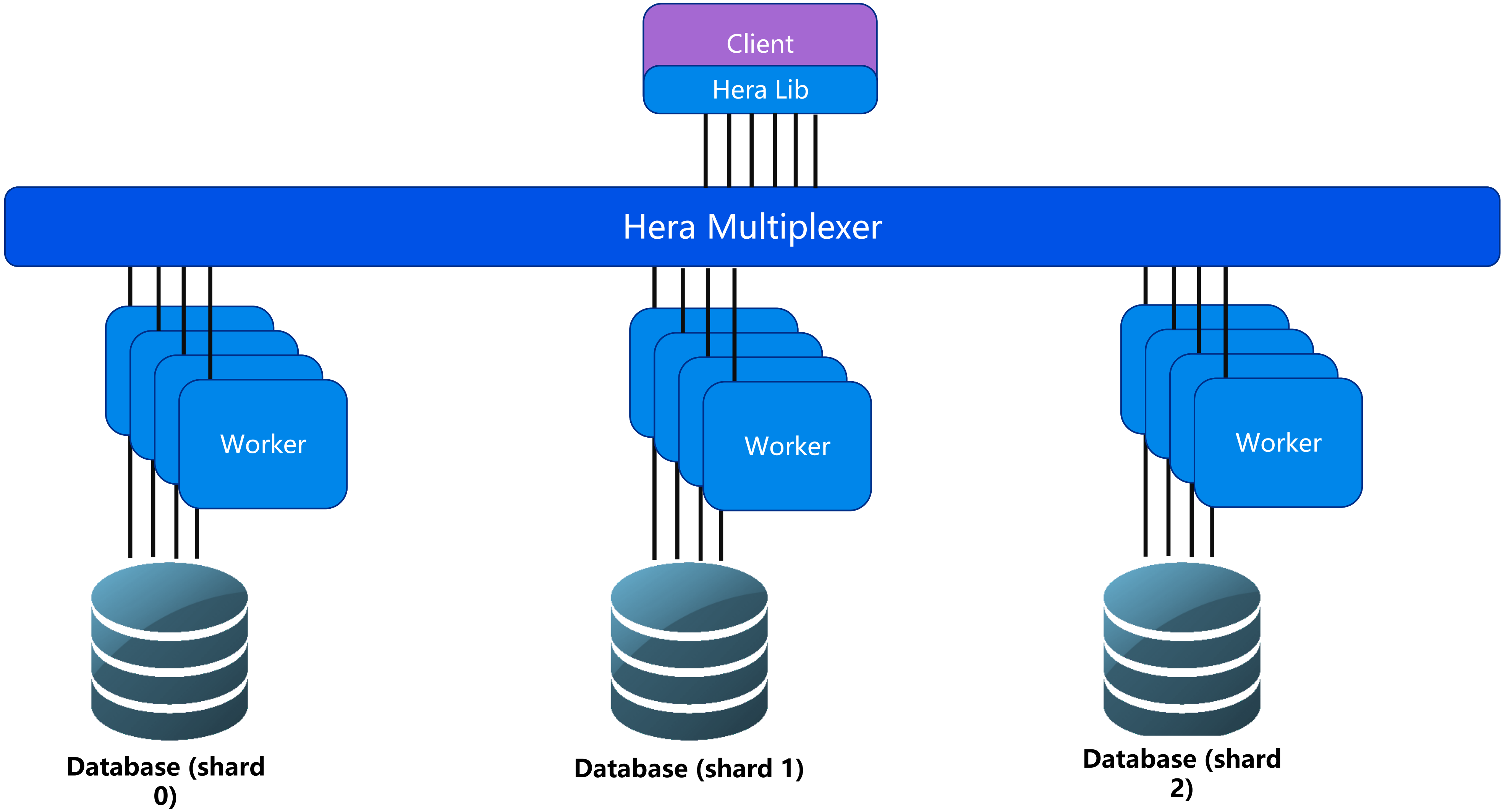
- **Bad batch jobs**
  - **Poorly coded**
  - **Retries immediately on failure**
- **Admins and DBAs then**
  - **Find vendor**
  - **Use triggers to block**

New

- **Trigger on overload**
- **Evict vendor causing excessive load**
- **Throttle based on load**
- **Will have Pull Request soon**

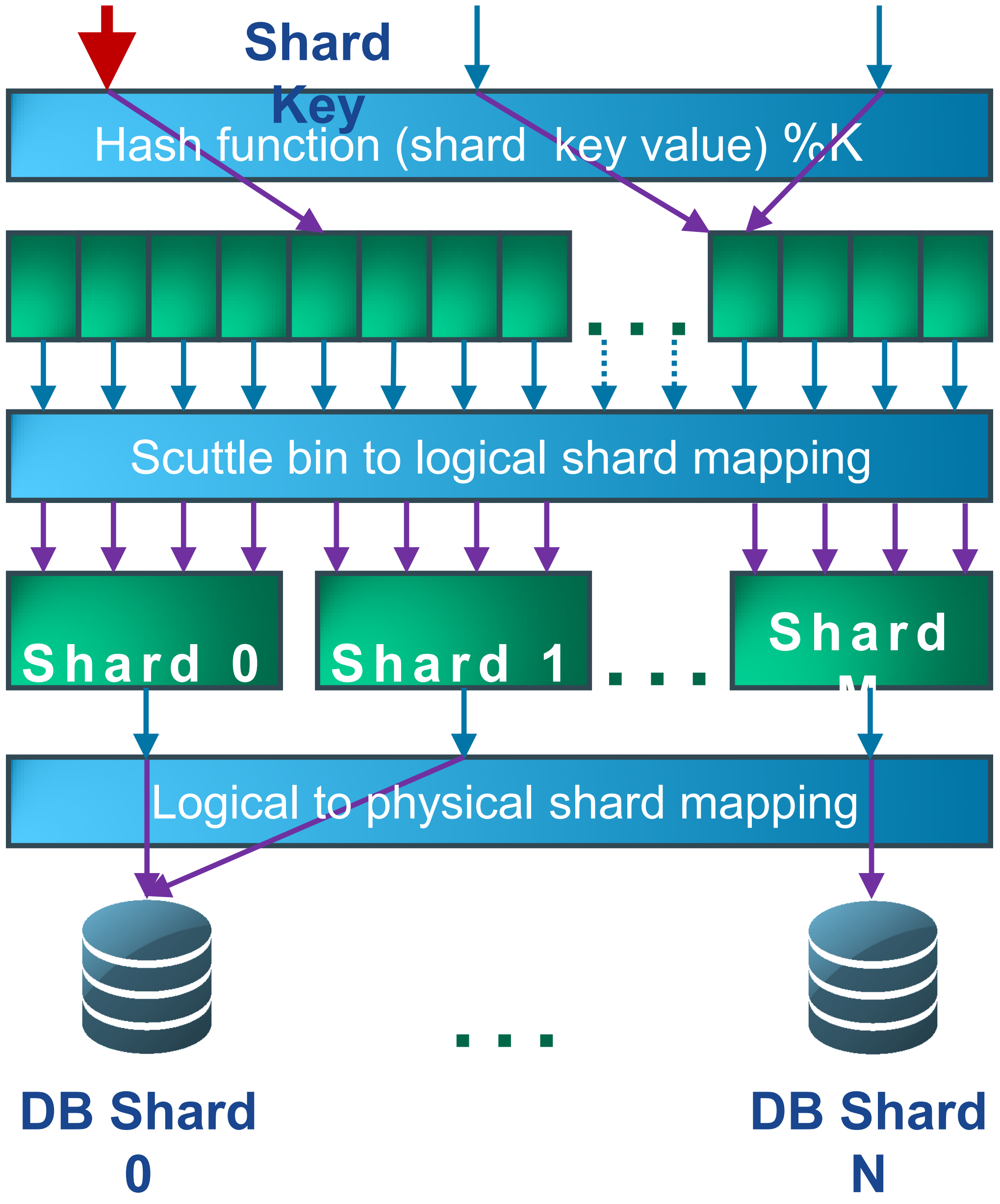
Sharding on Key-Value

Sharding for Scale



Sharding: Key-Value

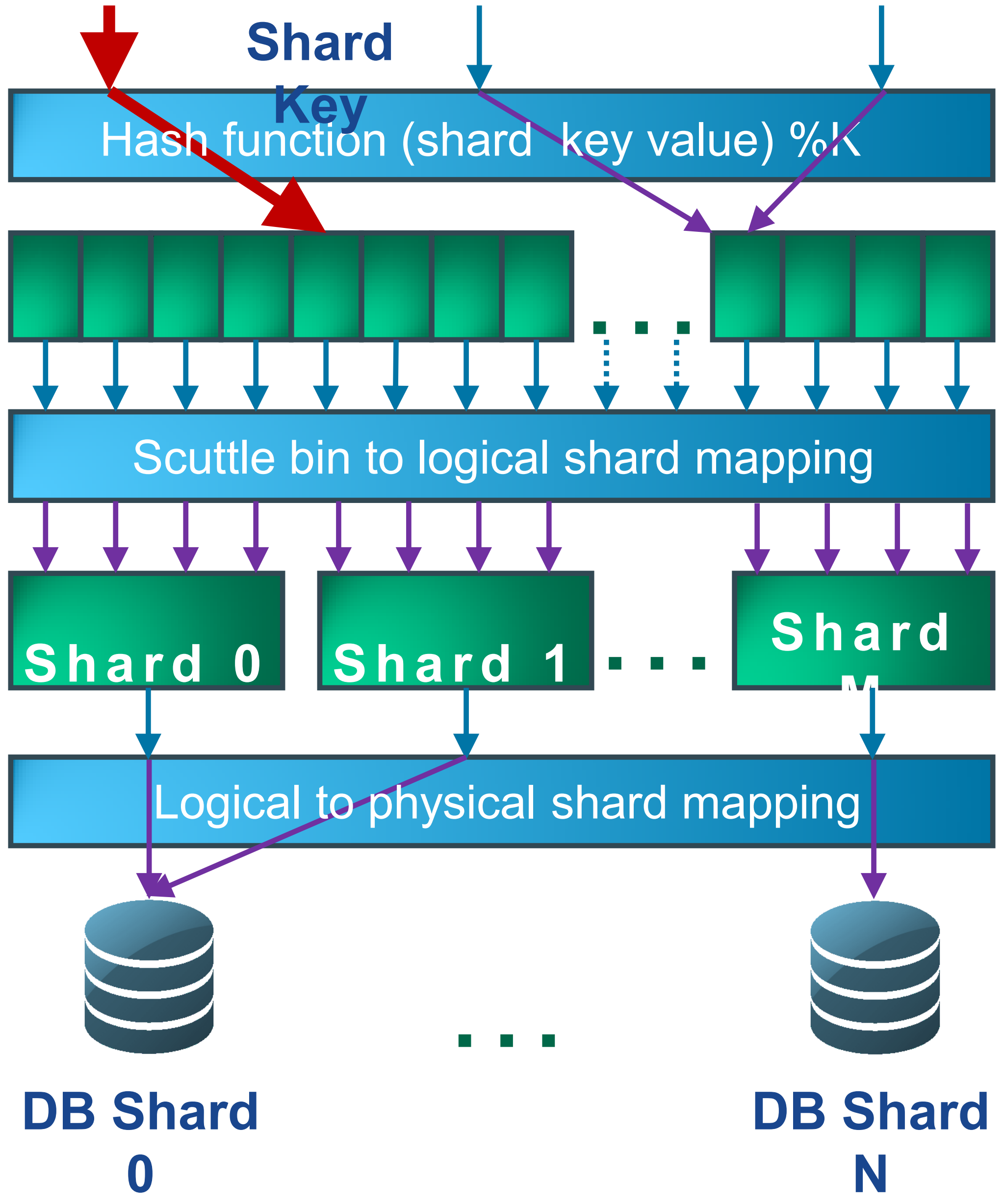
1. **Shard Key: account_id=2000**


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

Sharding: Key-Value

1. **Shard Key: account_id=2000**

2. **Murmur3Hash(2000)%1024=280**



Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

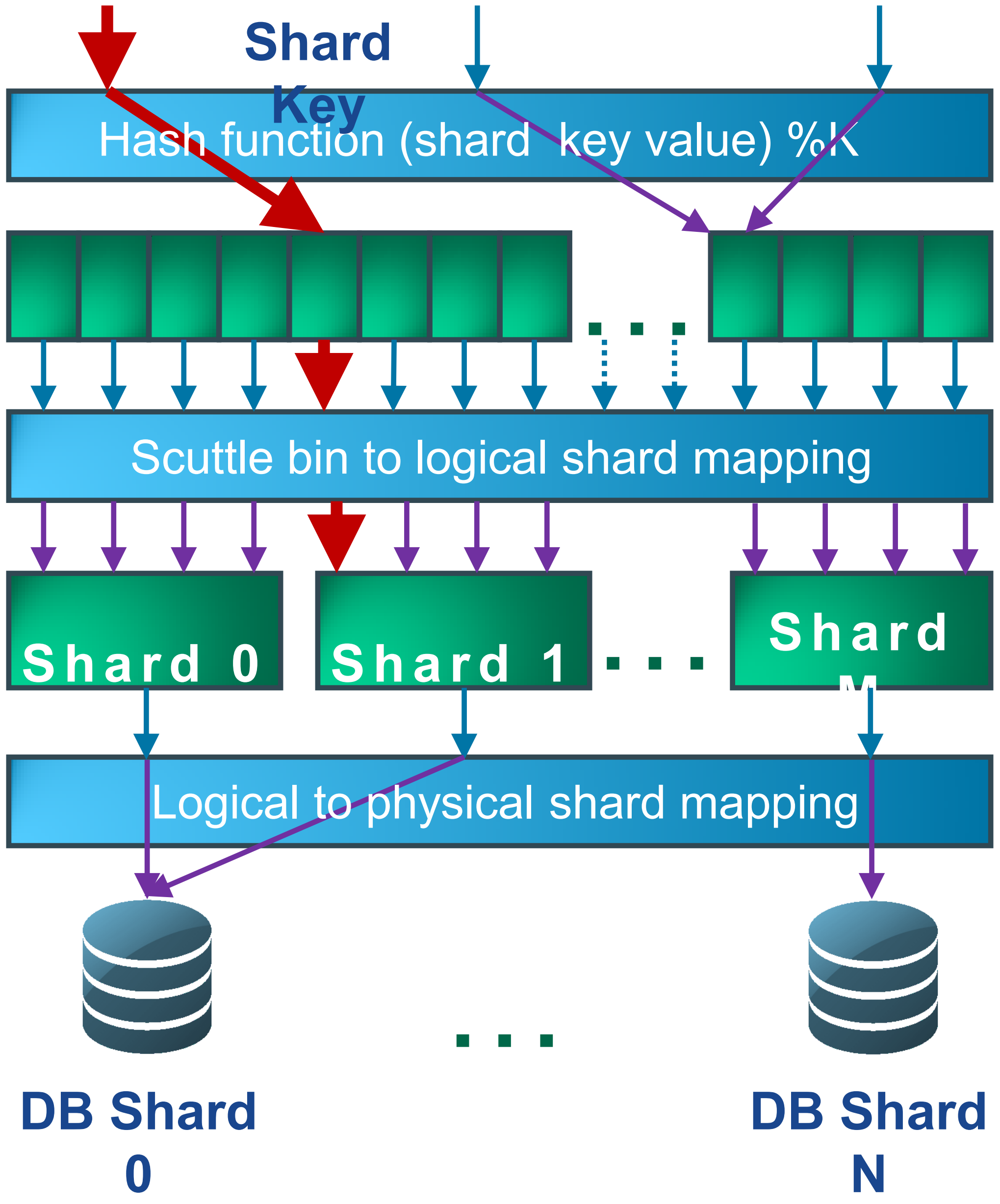
Sharding: Key-Value

1. Shard Key: account_id=2000

2. Murmur3Hash(2000)%1024=280

3. select shard_id from hera_shard_map where scuttle_bin = 280;

```
shard_id
----------
1
```


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

Sharding by Key-Value

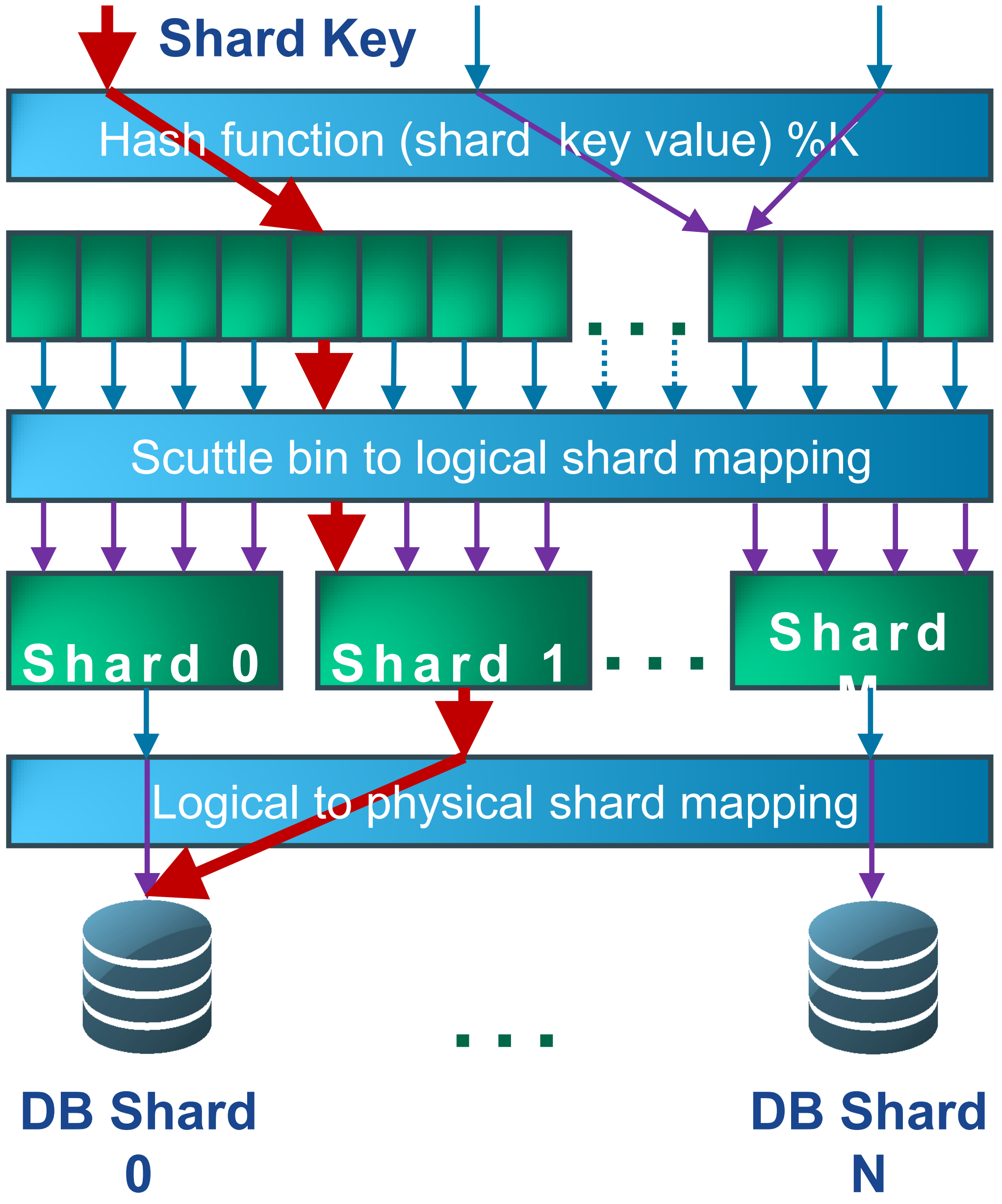
1. Shard Key: account_id=2000

2. Murmur3Hash(2000)%1024=280

3. select shard_id from hera_shard_map
   where scuttle_bin = 280;

```
shard_id
----------
    1
```

4. Connections to Logical DB
TWO_TASK_1=LOAN_SH1
TWO_TASK_1='tcp(loan-sh1: 3306)/loan'

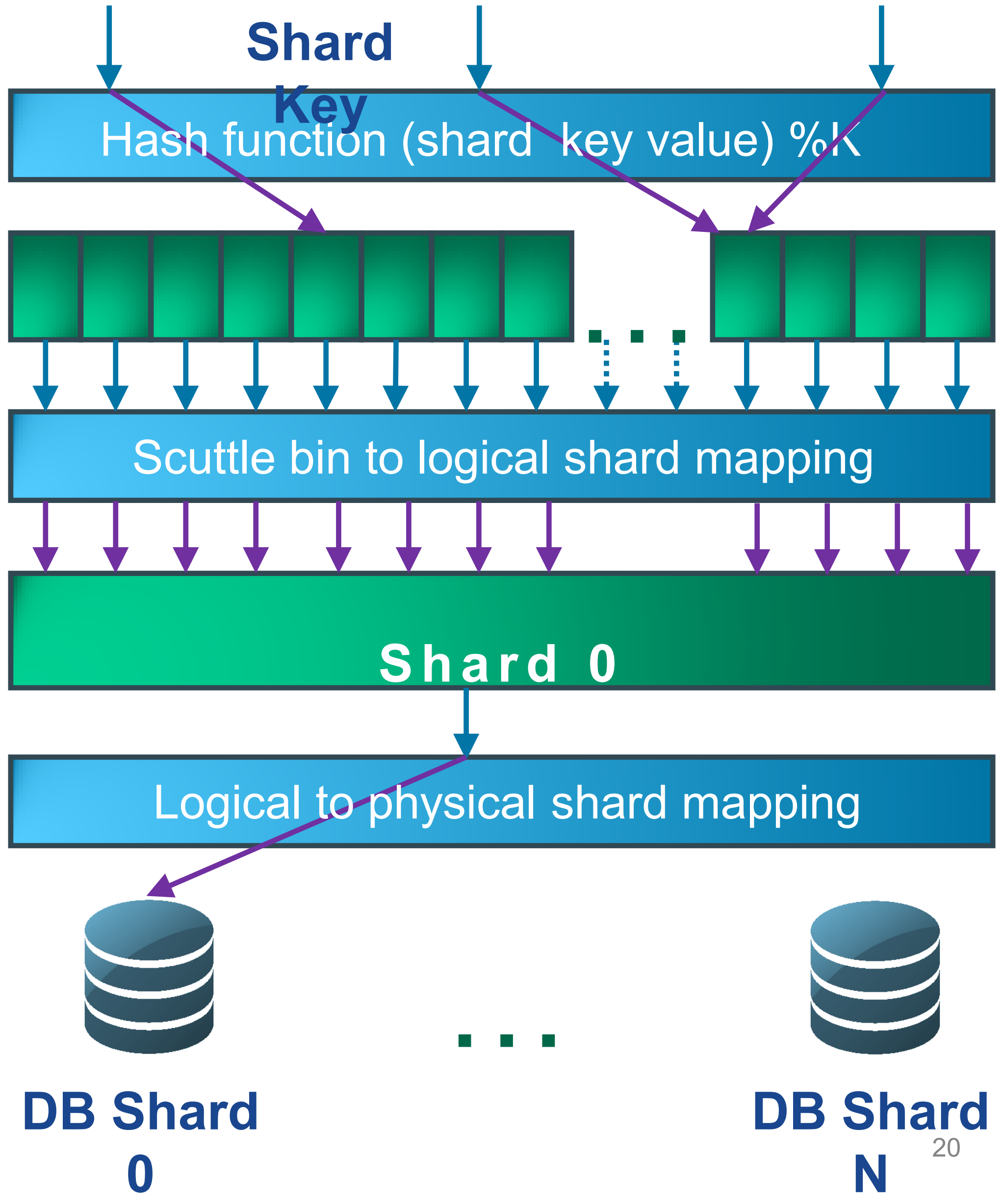


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

Sharding - Data Movement

Sharding: Legacy App Conversion

- All queries directed to shard 0
- Logs queries that don't bind to shard key

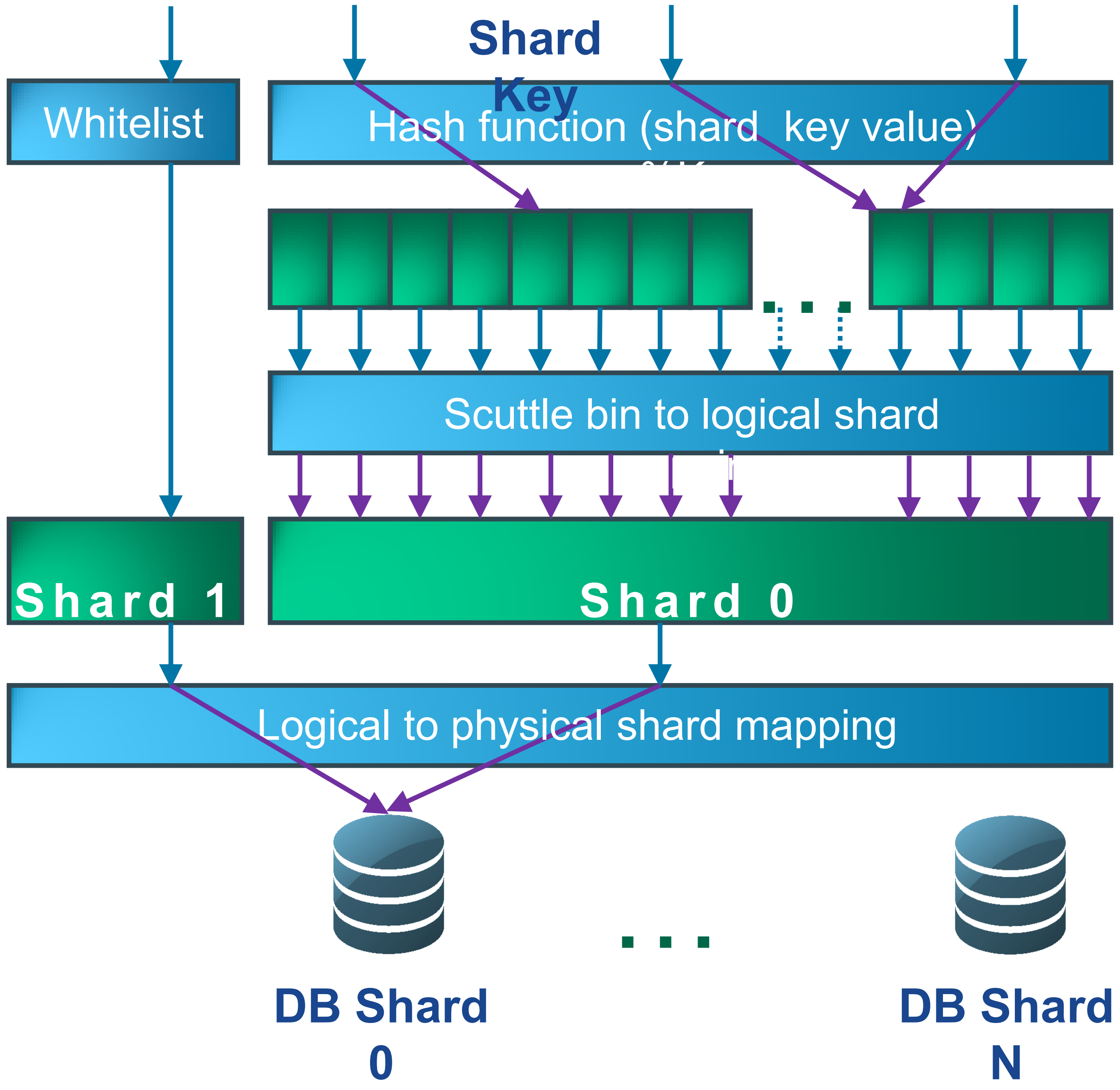

Shard Key
Hash function (shard key value) %K
. . .
Scuttle bin to logical shard mapping
Shard 0
Logical to physical shard mapping
. . .
DB Shard 0
DB Shard N

Sharding: Legacy App Conversion

- **Whitelist sends one value to a specific shard**
- **Limits risk of failures to 1 value**

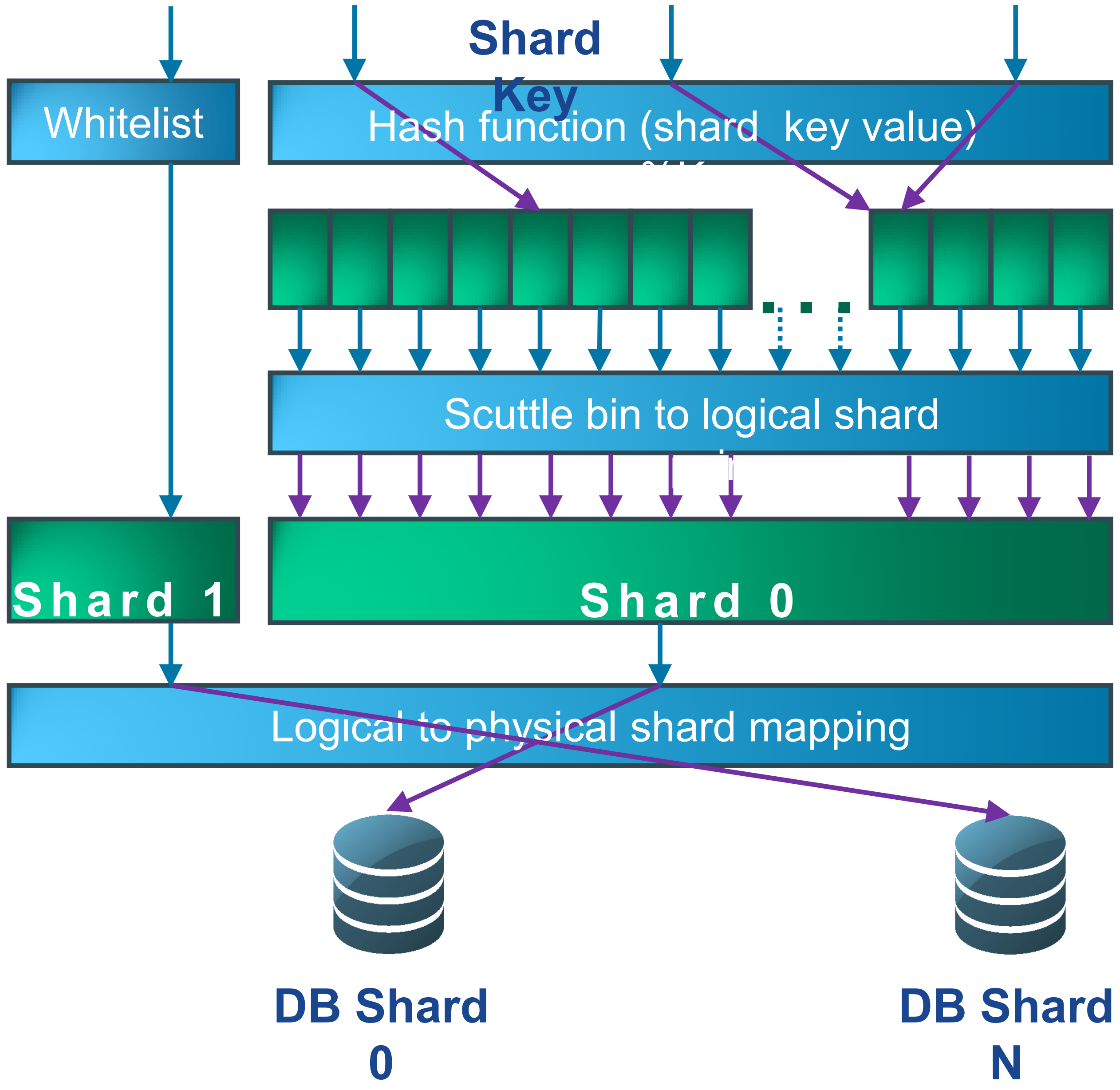
- **Hera uses Shard 1, but same DB**
- **Fast rollback on errors**

- **Repeat for larger sets**


Shard Key
Whitelist
Hash function (shard key value)
Scuttle bin to logical shard
Shard 1
Shard 0
Logical to physical shard mapping
DB Shard 0
DB Shard N

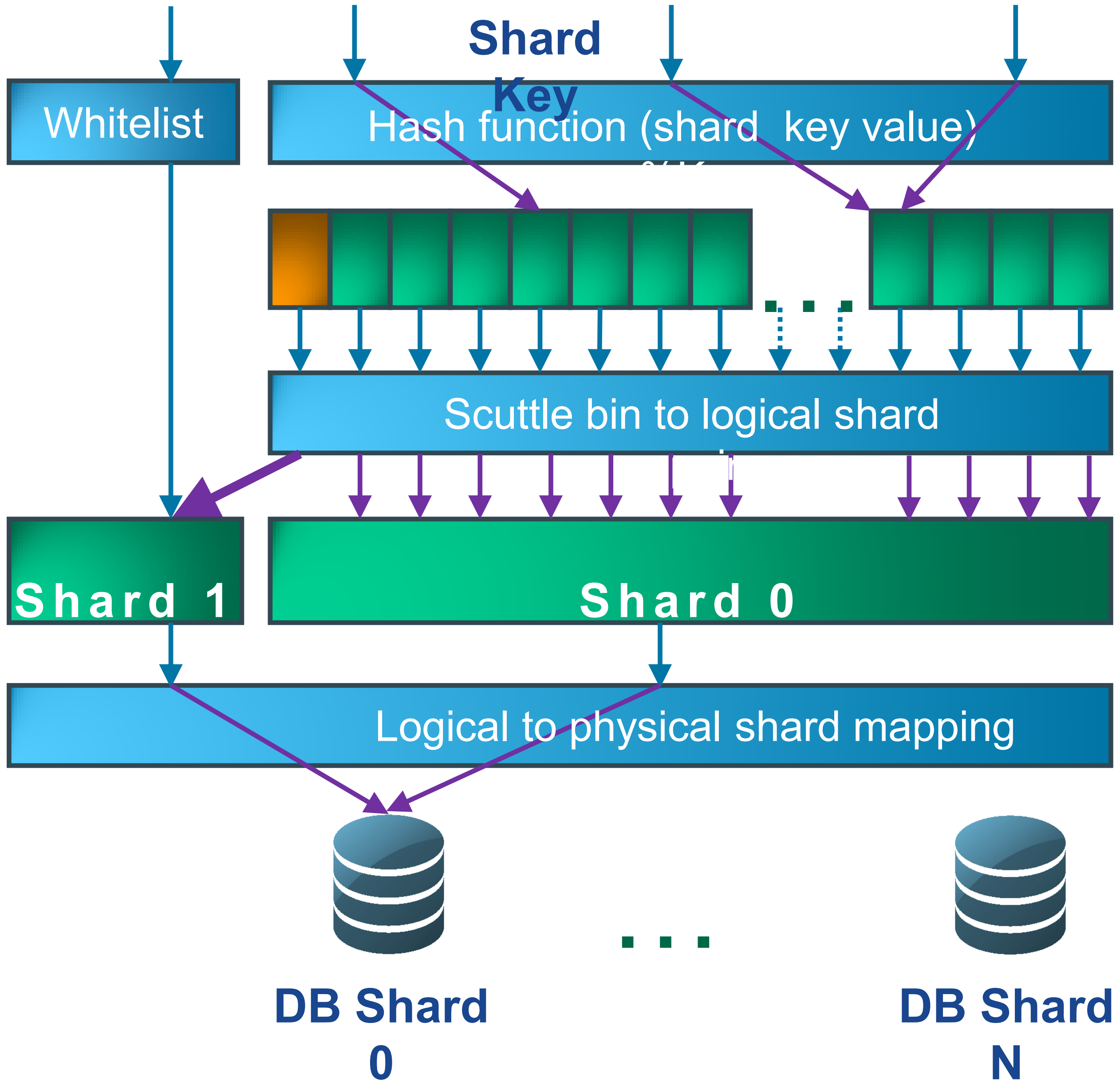
Sharding: Physical Whitelist

- **Validates permissions and data copy**


Shard Key
Whitelist
Hash function (shard key value)
Scuttle bin to logical shard
Shard 1
Shard 0
Logical to physical shard mapping
DB Shard 0
DB Shard N

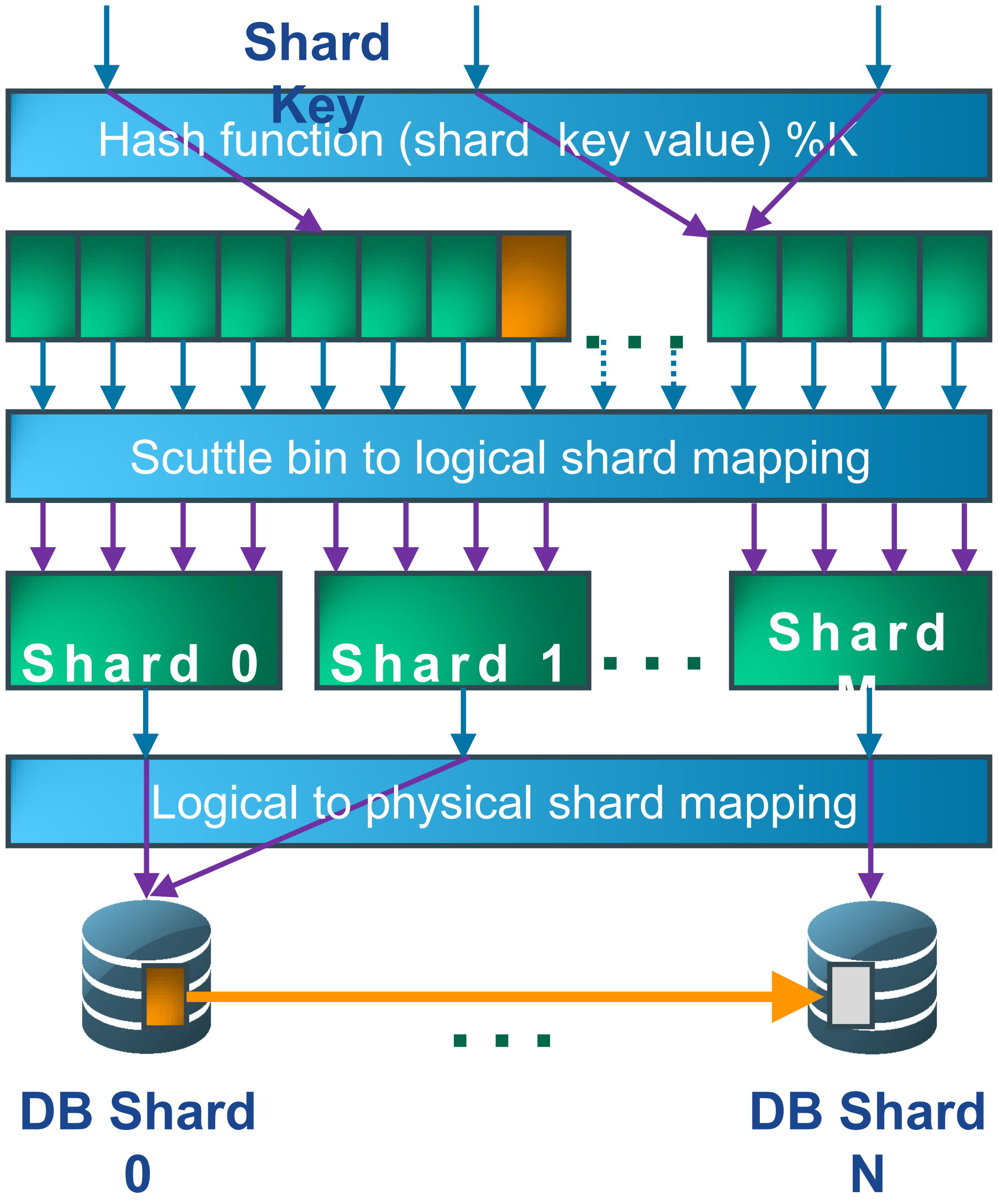
Sharding: Logical Move for One Scuttle Bin

- If successful, do a physical data move next


Shard Key
Whitelist
Hash function (shard key value)
Scuttle bin to logical shard
Shard 1
Shard 0
Logical to physical shard mapping
DB Shard 0
DB Shard N

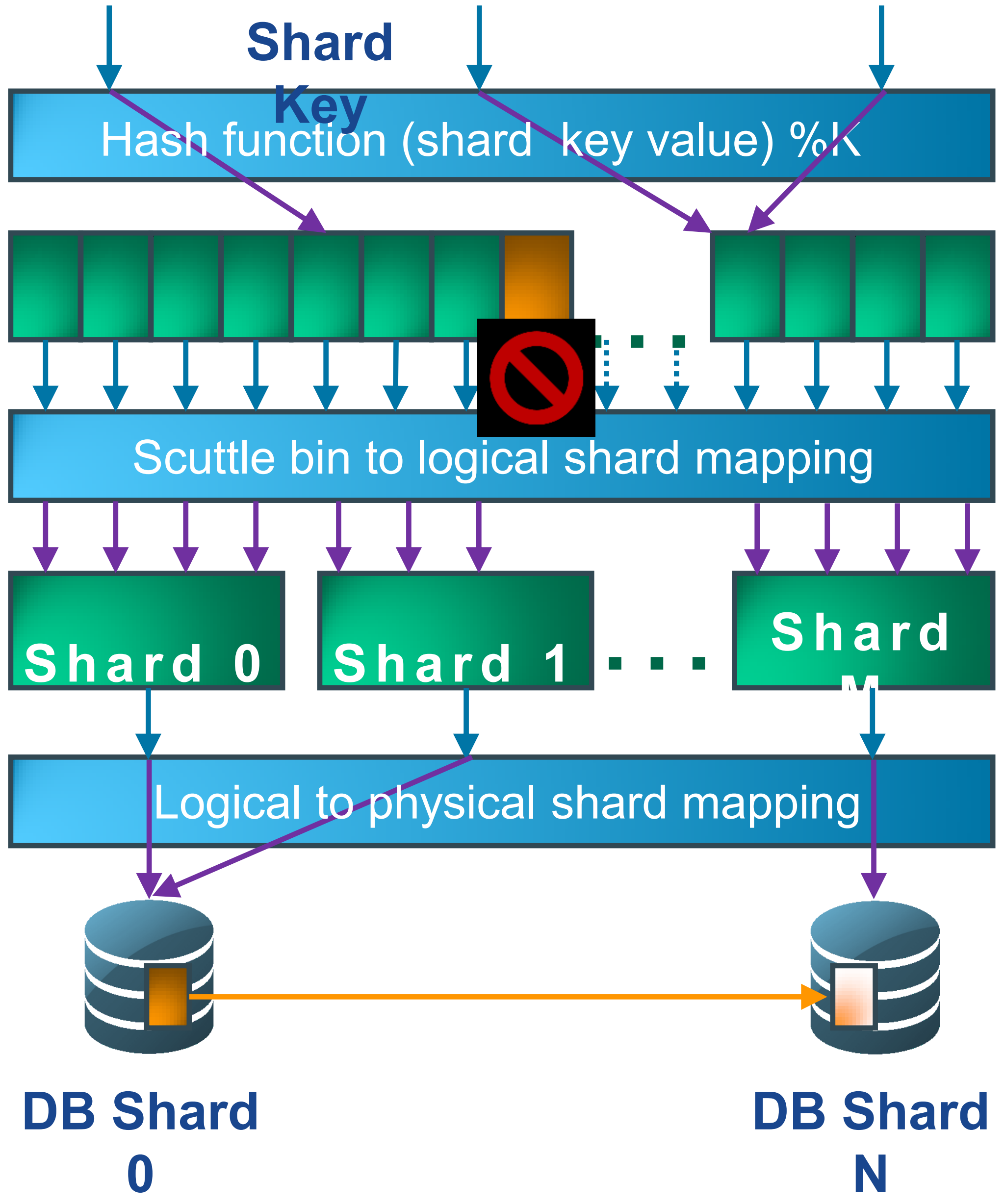
Sharding: Moving Scuttle Bin

- **Start data copy**
- **Typically, tables are partitioned by scuttle bin**


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

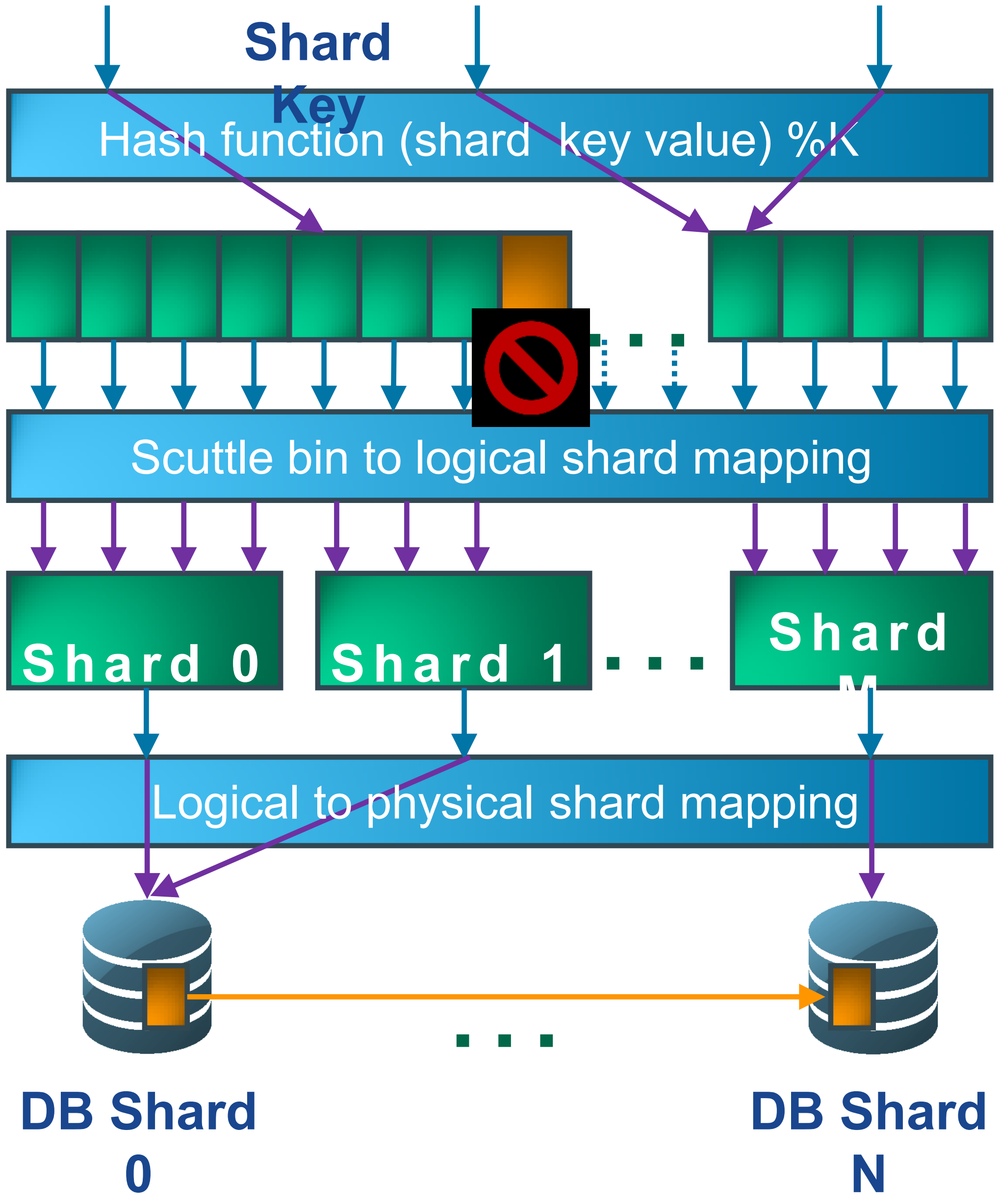
Sharding: Moving Scuttle Bin

- **Start data copy**
- **Typically, tables are partitioned by scuttle bin**
- **Block writes**


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

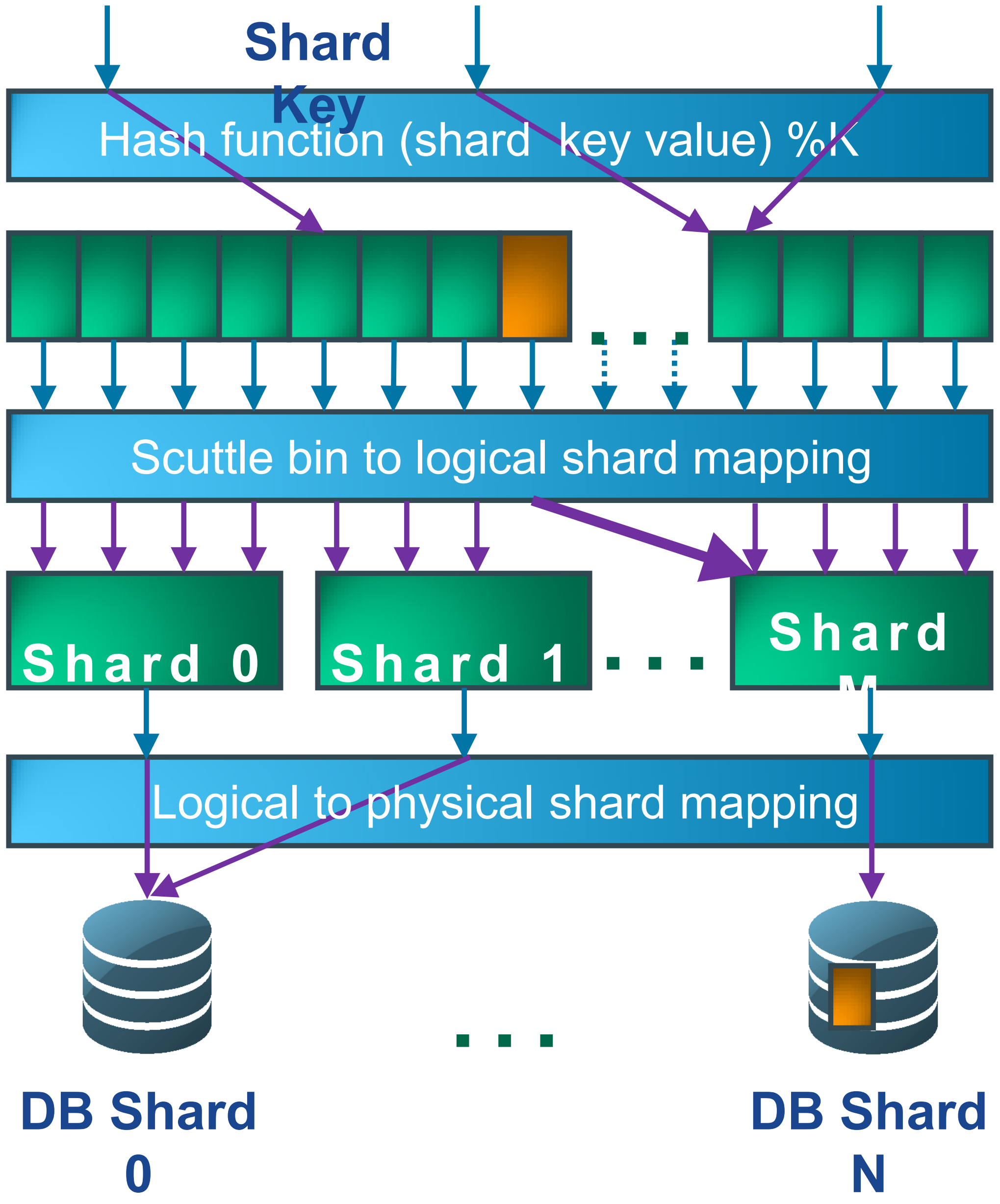
Sharding: Moving Scuttle Bin

- Start data copy
- Typically, tables are partitioned by scuttle bin
- Block writes
- Data fully copied


Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

Sharding: Moving Scuttle Bin

- Start data copy
- Typically, tables are partitioned by scuttle bin
- Block writes
- Data fully copied
- Update map
- Use scuttle bin in new location



Shard Key
Hash function (shard key value) %K
Scuttle bin to logical shard mapping
Shard 0
Shard 1
Shard M
Logical to physical shard mapping
DB Shard 0
DB Shard N

Sharding Constraints

- For DMLs (insert/update/delete), a single shard key should be passed in for each sql.
- For read-only queries, one or more shard key values can be passed in for each sql.

  *However Hera server will be able to handle multiple shard keys - scatter-gather- in phase II.
- No cross shards for dmls in the same transaction.
- Queries by ROWID will not be supported.
- No db sequence should be allowed in sharded database.
  - Client should use the ID service to get the global unique id.
- No PL/SQL should be allowed.

Sharding Compatible Query

- **Shard Key: account_id**

- **Select * from loan where loan.account_id = ?**

- **select * from loan, appfile where loan.id = ? and appfile.loan_id = loan.id *and loan.account_id = ? and appfile.account_id = loan.account_id***

- ***Update loan set loan.amount=? Where loan.account_id=?***

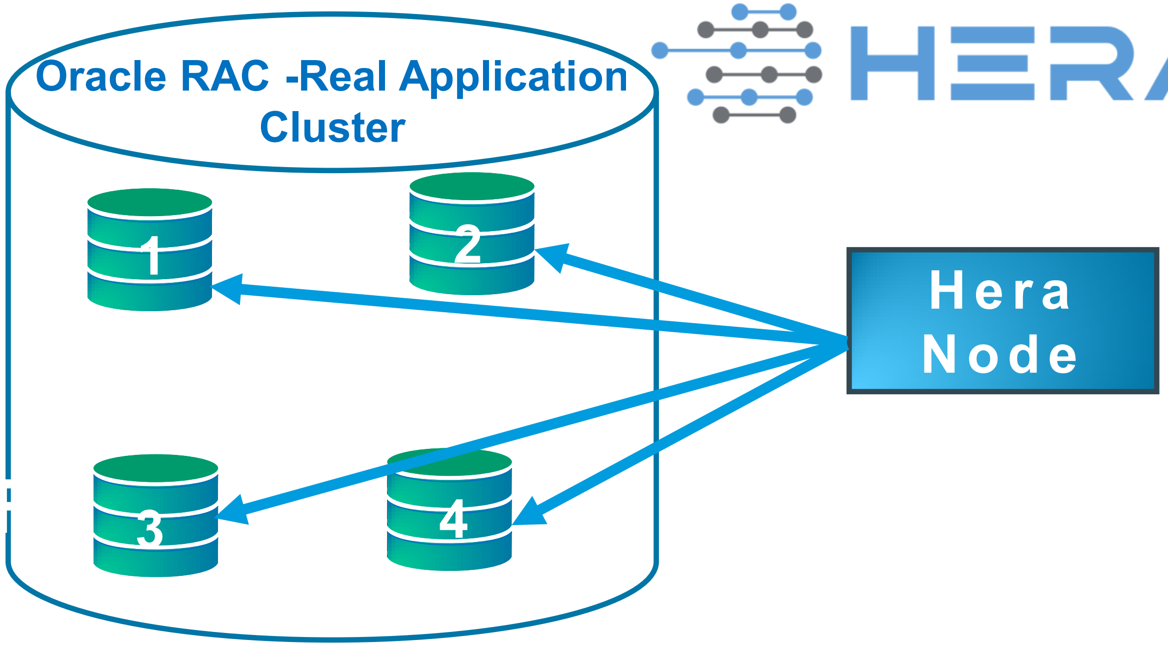
Database Maintenance

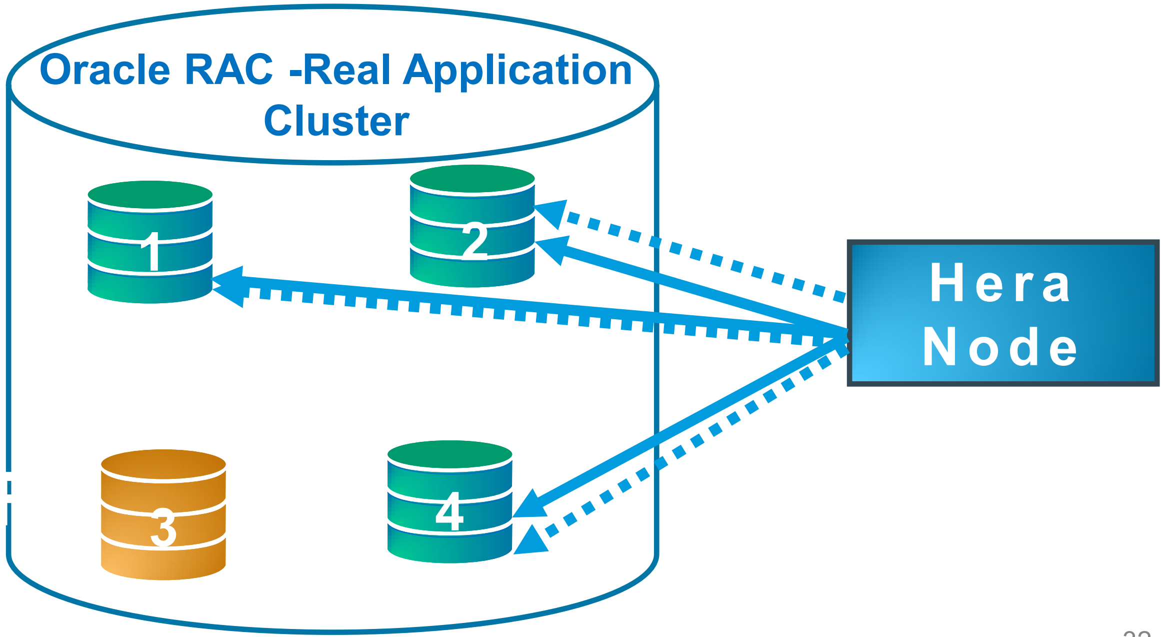
DB Maintenance

- **Enable DBA to control and prepare HERA services for RAC maintenance.**
- **Enable DBA to control and prepare OCC services for load re-distribution.**
- **Meta data table (HERA_MAINT) which stores Host Name, Service Name, RAC Instance ID, Command, Command Effective Time**

DB Maintenance

- Preparing Oracle RAC node 3 maintenance
- DBAs remove node 3 from Oracle configs
- insert into hera_maint (inst_id, status, status_time, module, machine) values (3, 'F', [unix epoch], [hera pool name], [host]





DB Maintenance Benefits

- No interruption or impact to transactions
- No database login storm
- Staggered restart throughout restart window
- Fast restart typically used only during database cutover
- Supports Sharding and Read Replica Query Failover
- Keep controls near those who need it

Acknowledgements

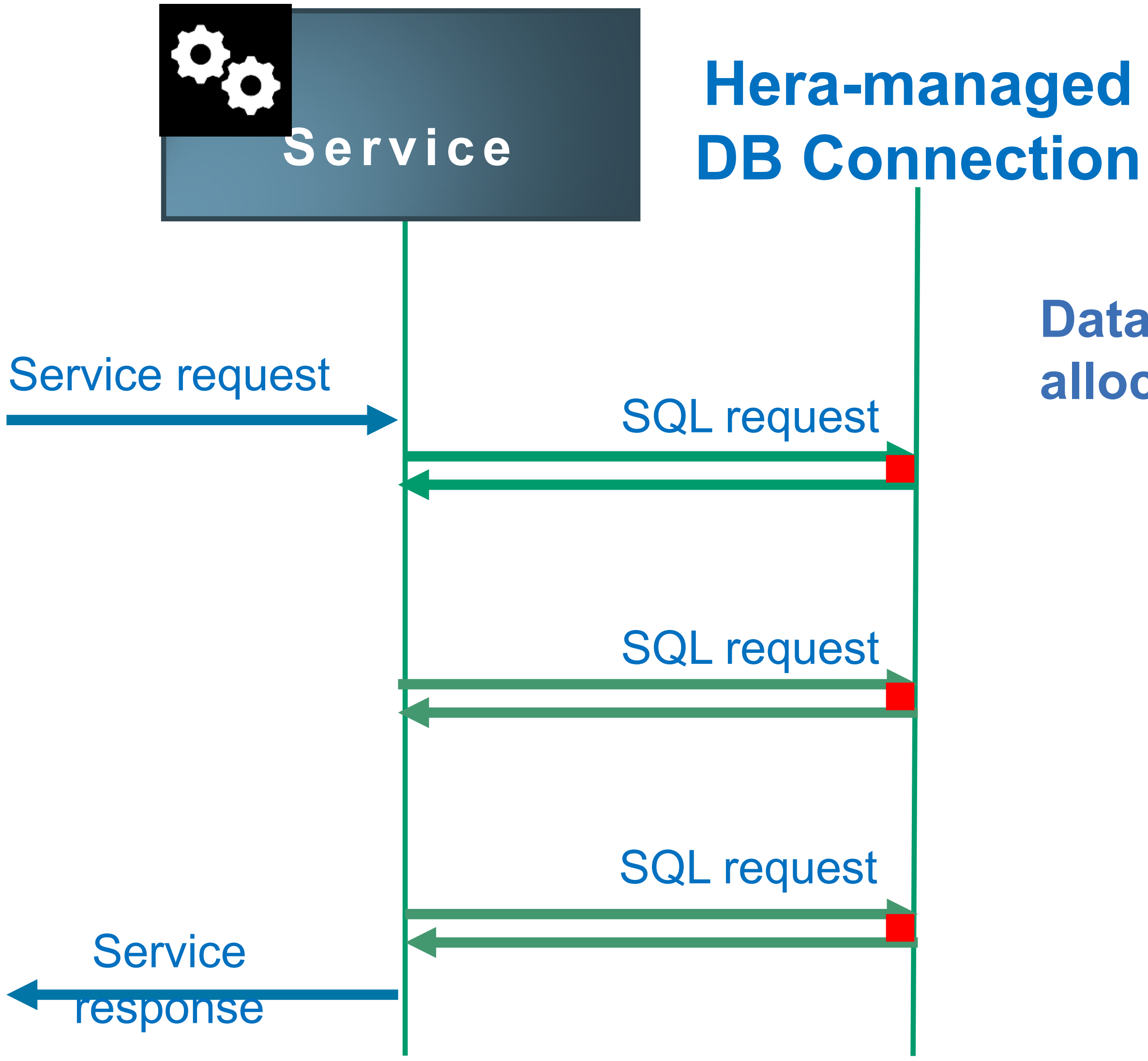
PayPal DBA & DAL Team
Petrica
Liana
Mukundan
Manoj
Sai

Shuping
Stephanie
Paresh
Kamlakar
Yaping

HERA

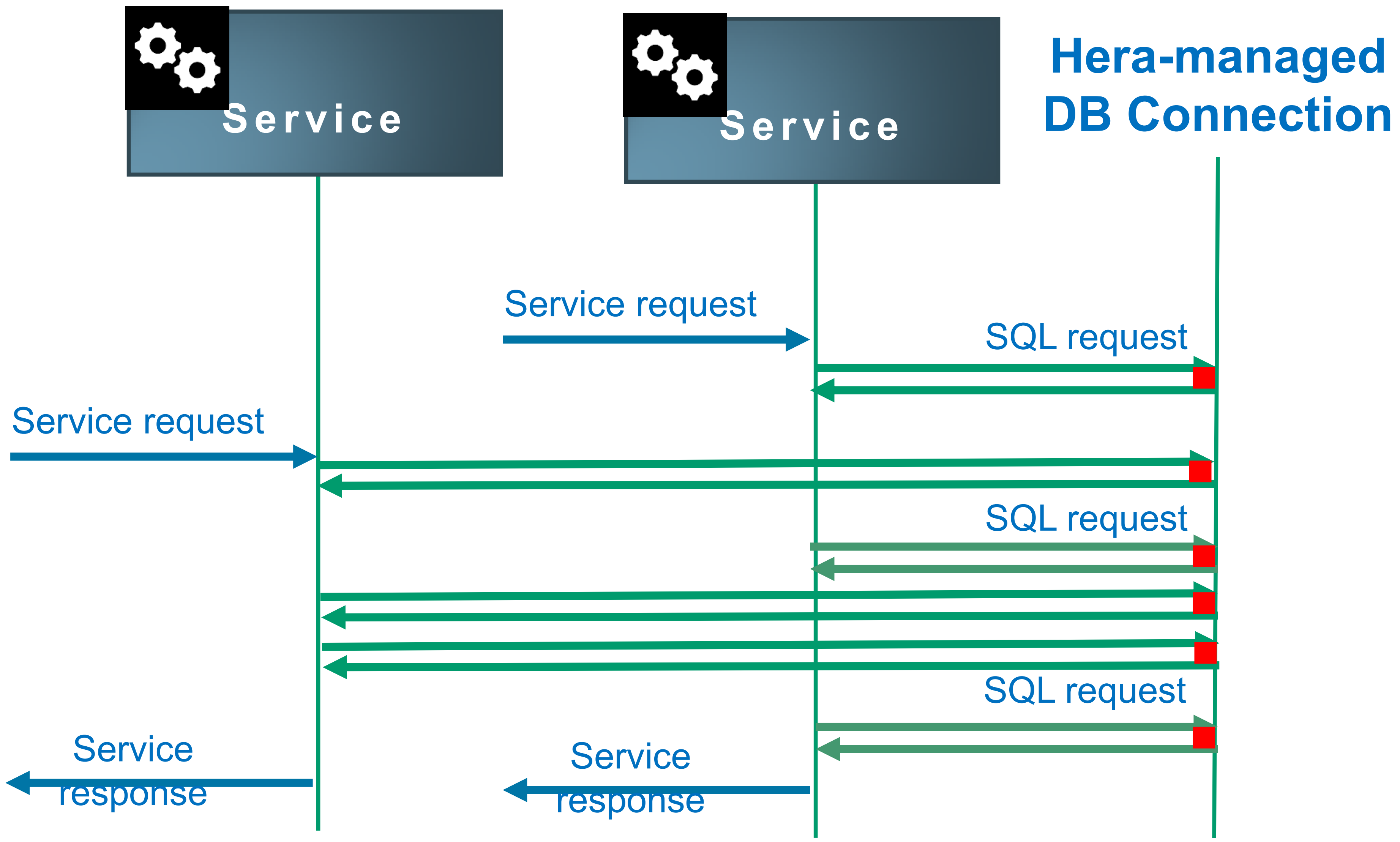
github.com/paypal/hera

Typical service request



Database connection is allocated but mostly idle

Multiplex DB Connection


Service
Service
Hera-managed DB Connection
Service request
SQL request
Service request
SQL request
SQL request
Service response
Service response