

Sharding Architectures

International PHP Conference 2008

Who the f*** is talking?

- ✧ David Soria Parra <dsp@php.net>
- ✧ PHP since 7 years
- ✧ Software Engineer at Sun Microsystems
- ✧ Sun Microsystems Web Stack
- ✧ PHP Source Committer
- ✧ Various other OSS Projects (Mercurial, etc)

Agenda

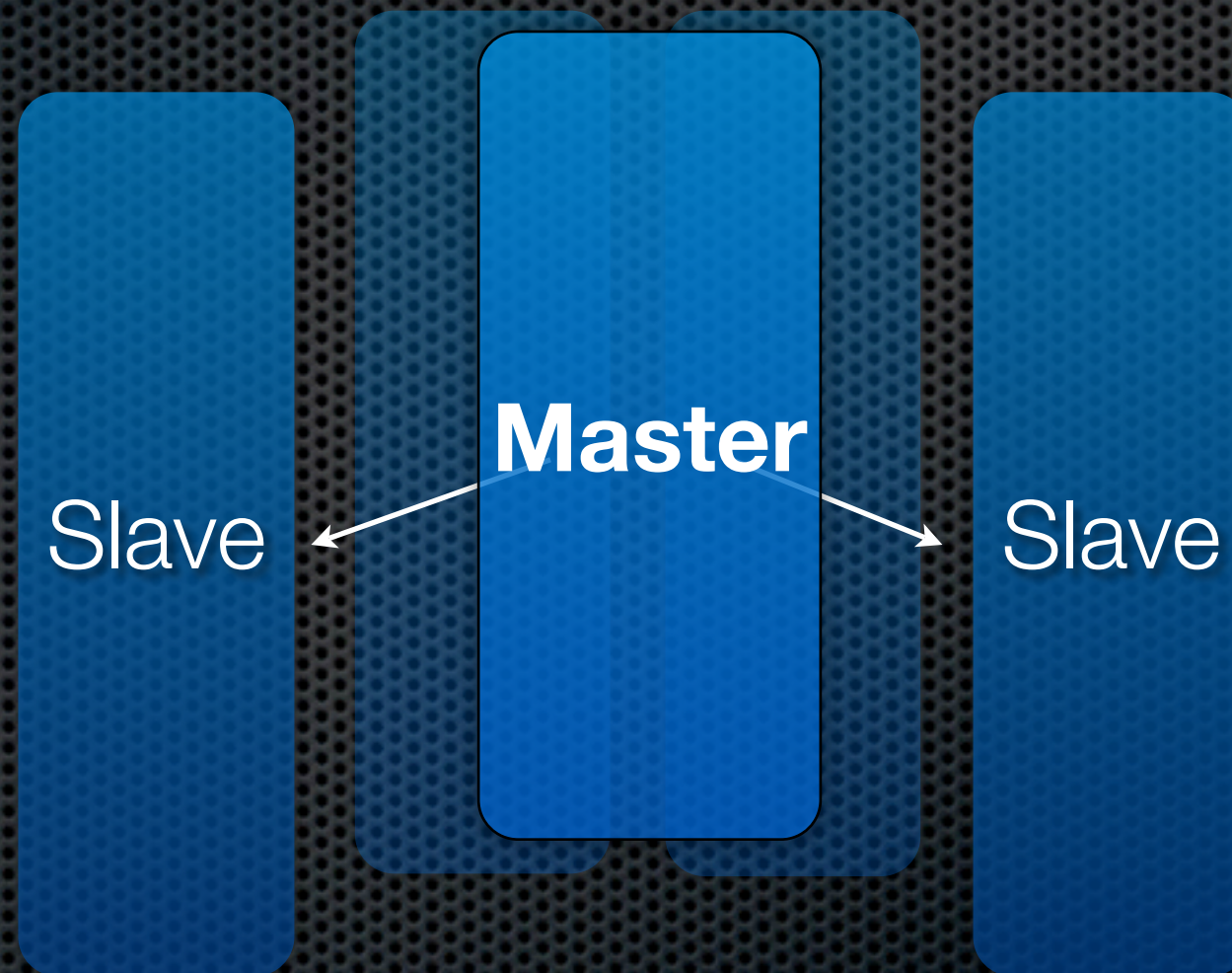
- ✦ Problem
- ✦ Possible Solutions ...and what not
- ✦ What is sharding?
- ✦ Implementation
- ✦ Optimization
- ✦ Migration

Problem

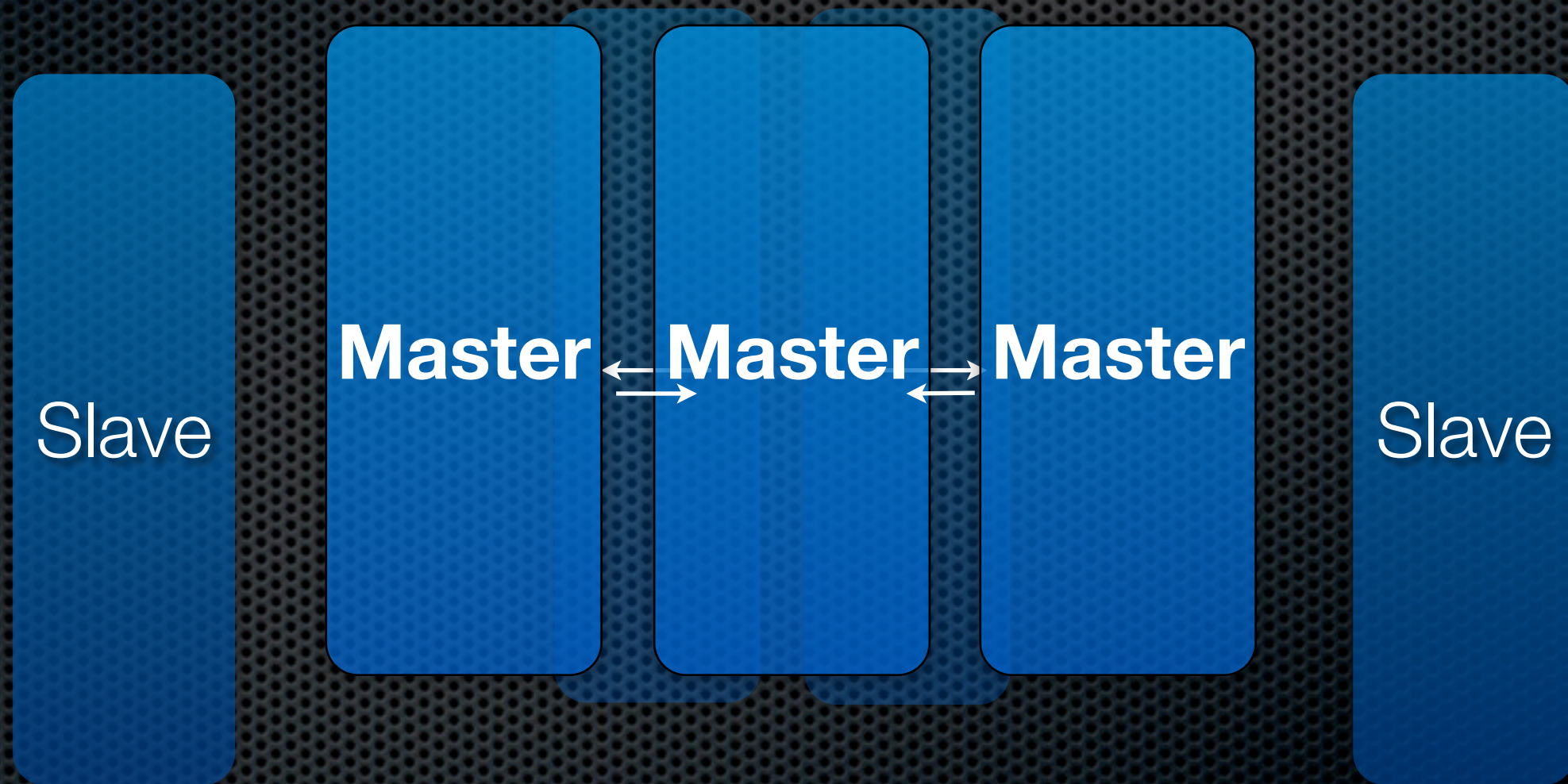
- ✧ Scalability
 - ✧ What kind of scalability?
- ✧ High read and write traffic
- ✧ Huge amount of data

Solutions

Master/Slave replication



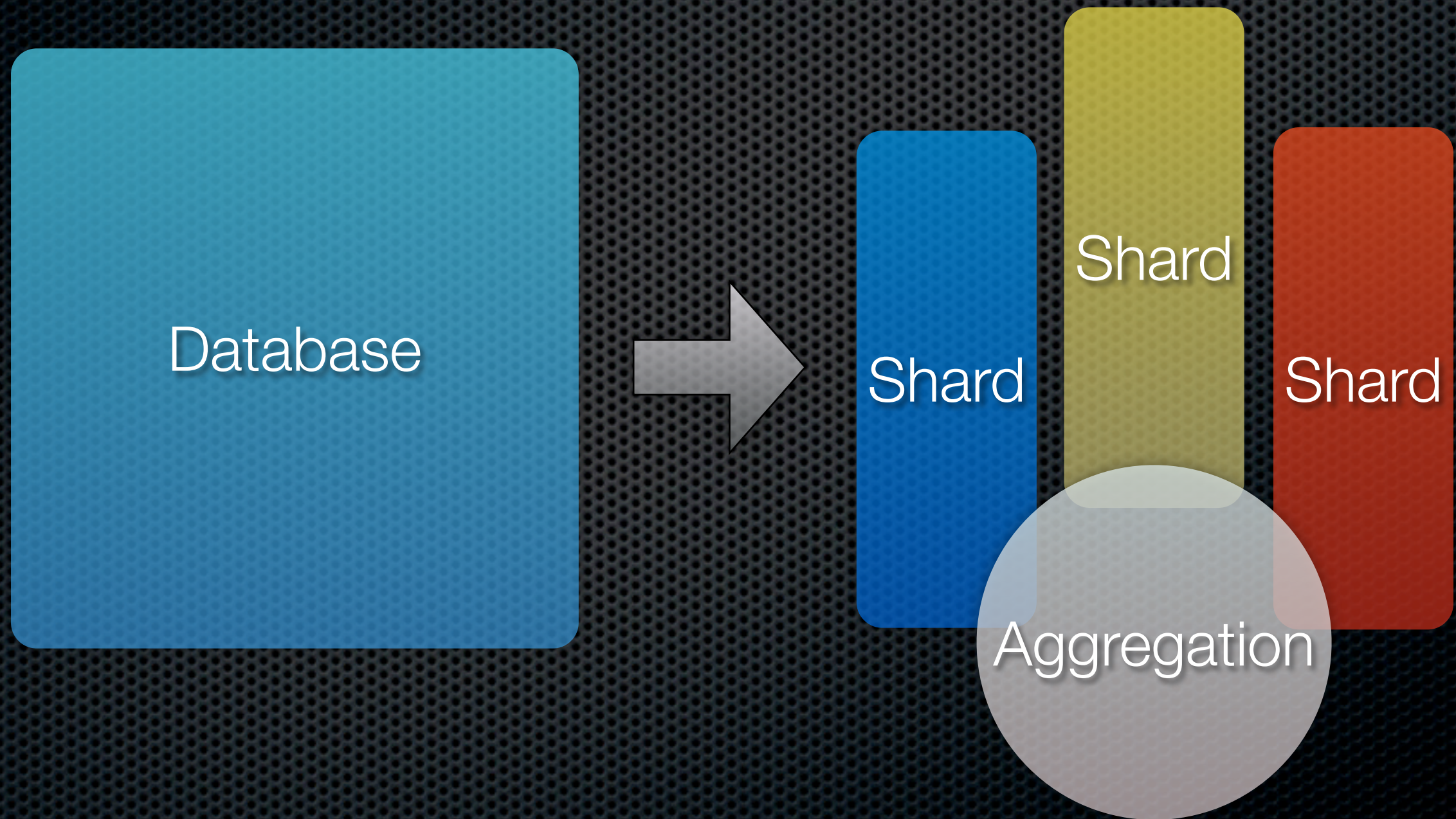
Master/Master replication



Master/Master Master/Slave

- ✧ Master / Master
 - ✧ unstable / no official support
 - ✧ replication not distribution
- ✧ Master / Slave
 - ✧ no write optimization
 - ✧ replication not distribution

Sharding



What is sharding?

- ✦ Application side **implementation**
- ✦ Application side **aggregation**
- ✦ **Splitting** data across databases
- ✦ Parallelization

Sharding is **not**...

- ✧ sharding is...
 - ✧ **NOT** replication / backup
 - ✧ **NOT** clustering
 - ✧ **NOT** just spreading

Implementation


```
graph LR; PHP[PHP application] --- S0[shard 0]; PHP --- S1[shard 1]; PHP --- S2[shard 2];
```

PHP
application

shard 0

shard 1

shard 2

user ID: 9
→

PHP
application

shard 0

shard 1

shard 2

user ID: 9

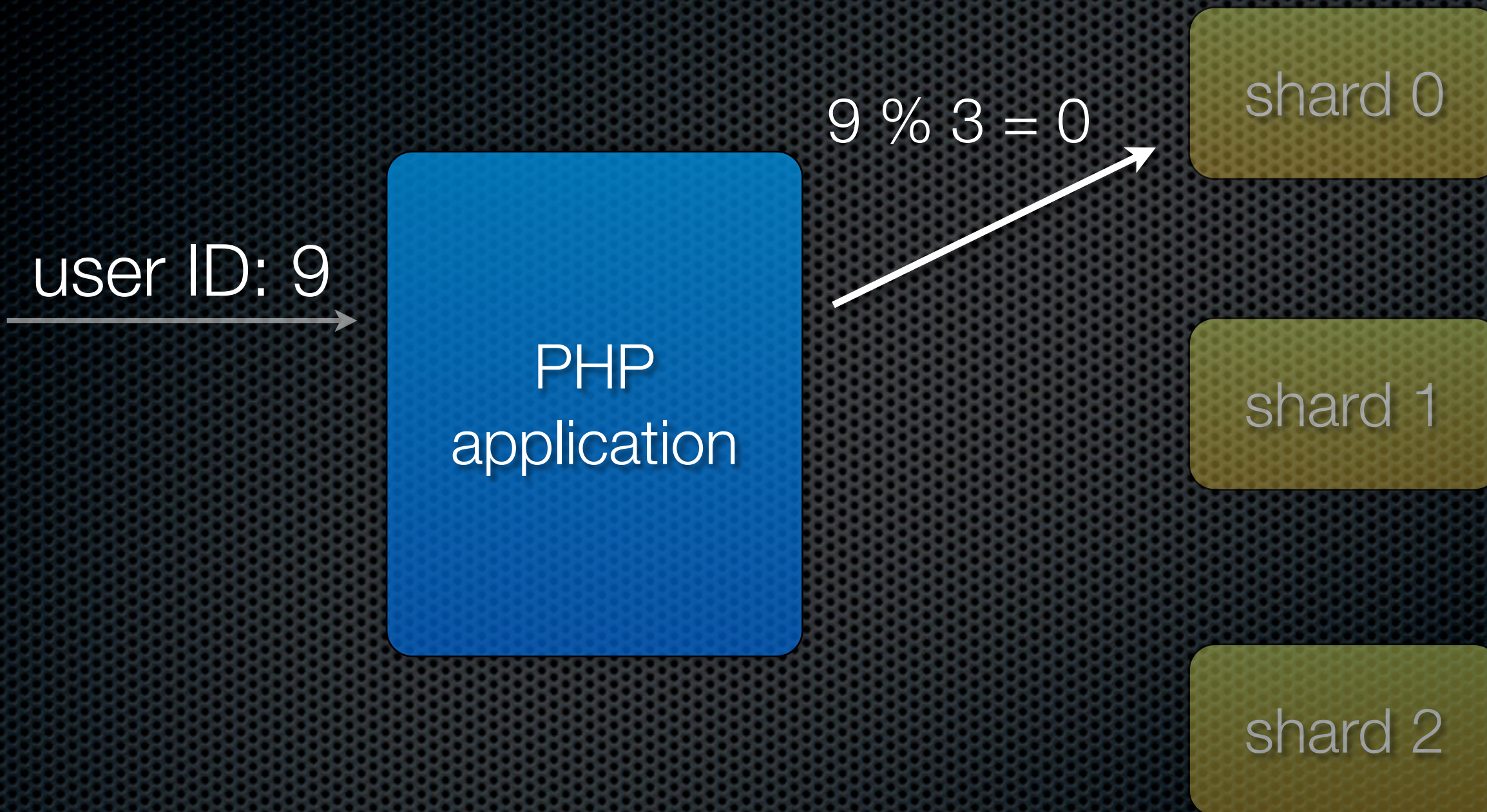
PHP
application

$$9 \% 3 = 0$$

shard 0

shard 1

shard 2



user ID: 9



```
graph LR; User[user ID: 9] --> PHP[PHP application]; PHP -- "9 % 3 = 0" --> Shard0[shard 0];
```

PHP
application

$$9 \% 3 = 0$$

shard 0

shard 1

shard 2

user ID: 9



PHP
application

$$9 \% 3 = 0$$

shard 0

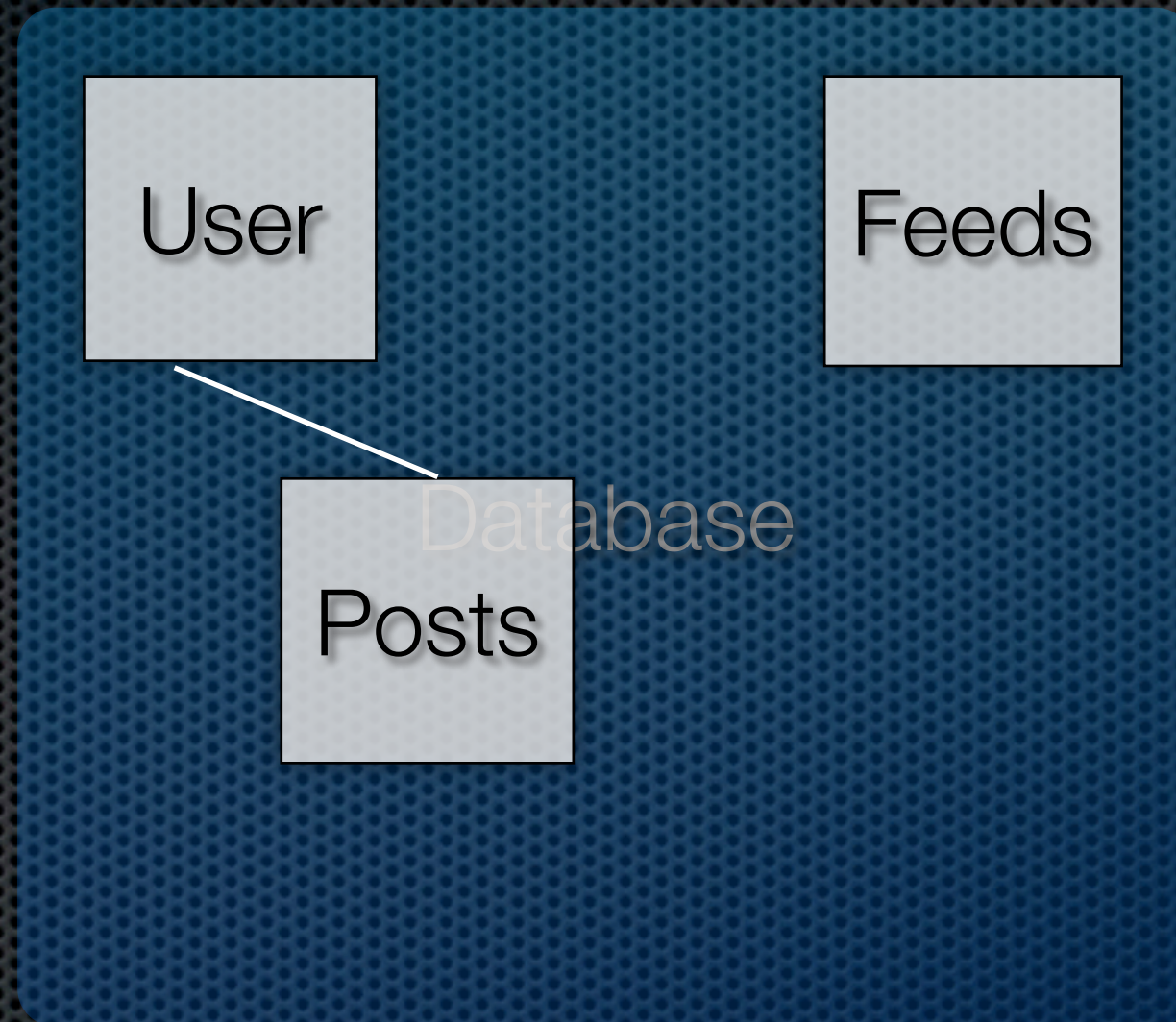
shard 1

shard 2

Shard

- ✦ Tables with a lot of writes
- ✦ Tables with a lot of reads
- ✦ Related data on one shard

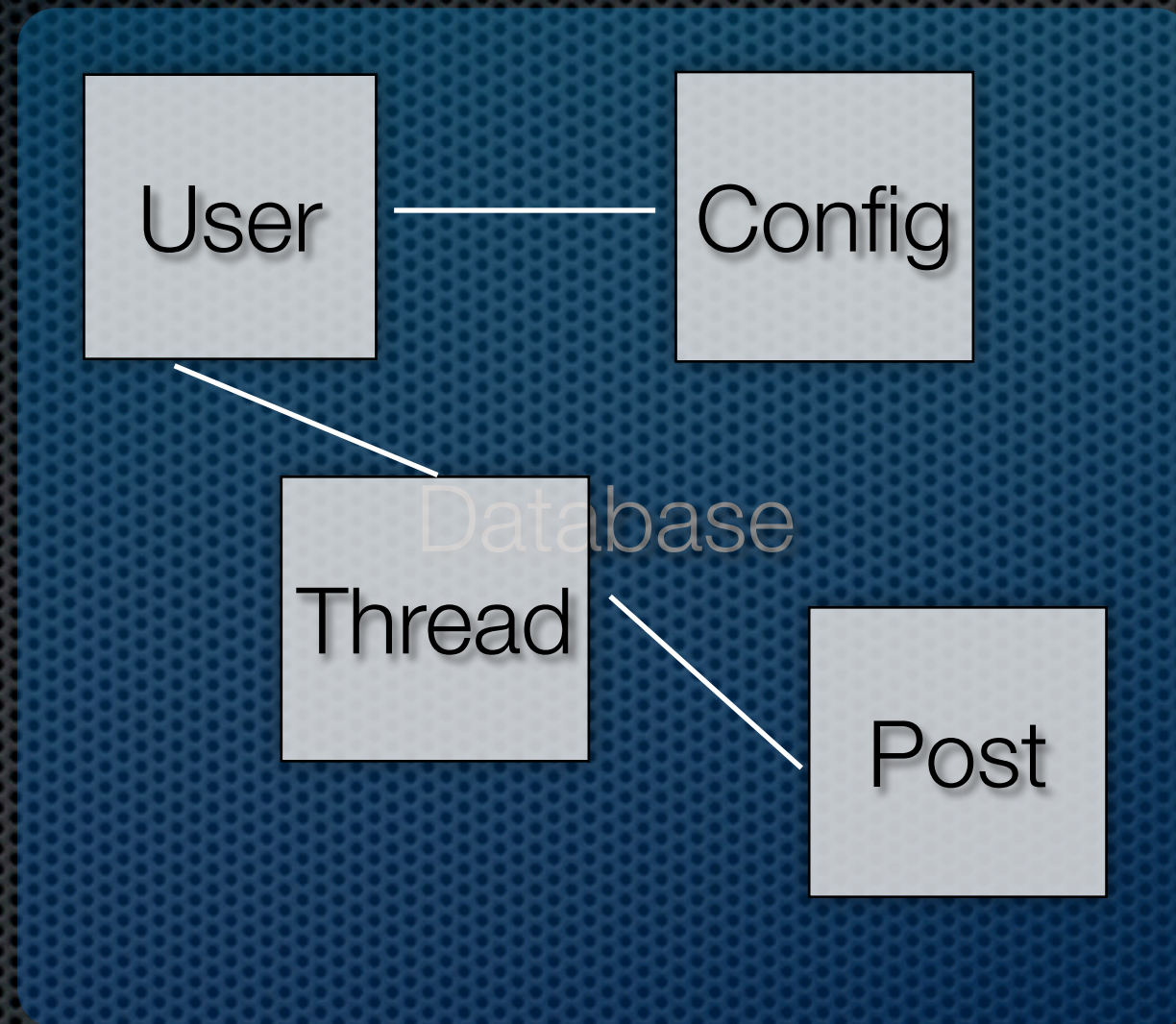
Splitting: Tables



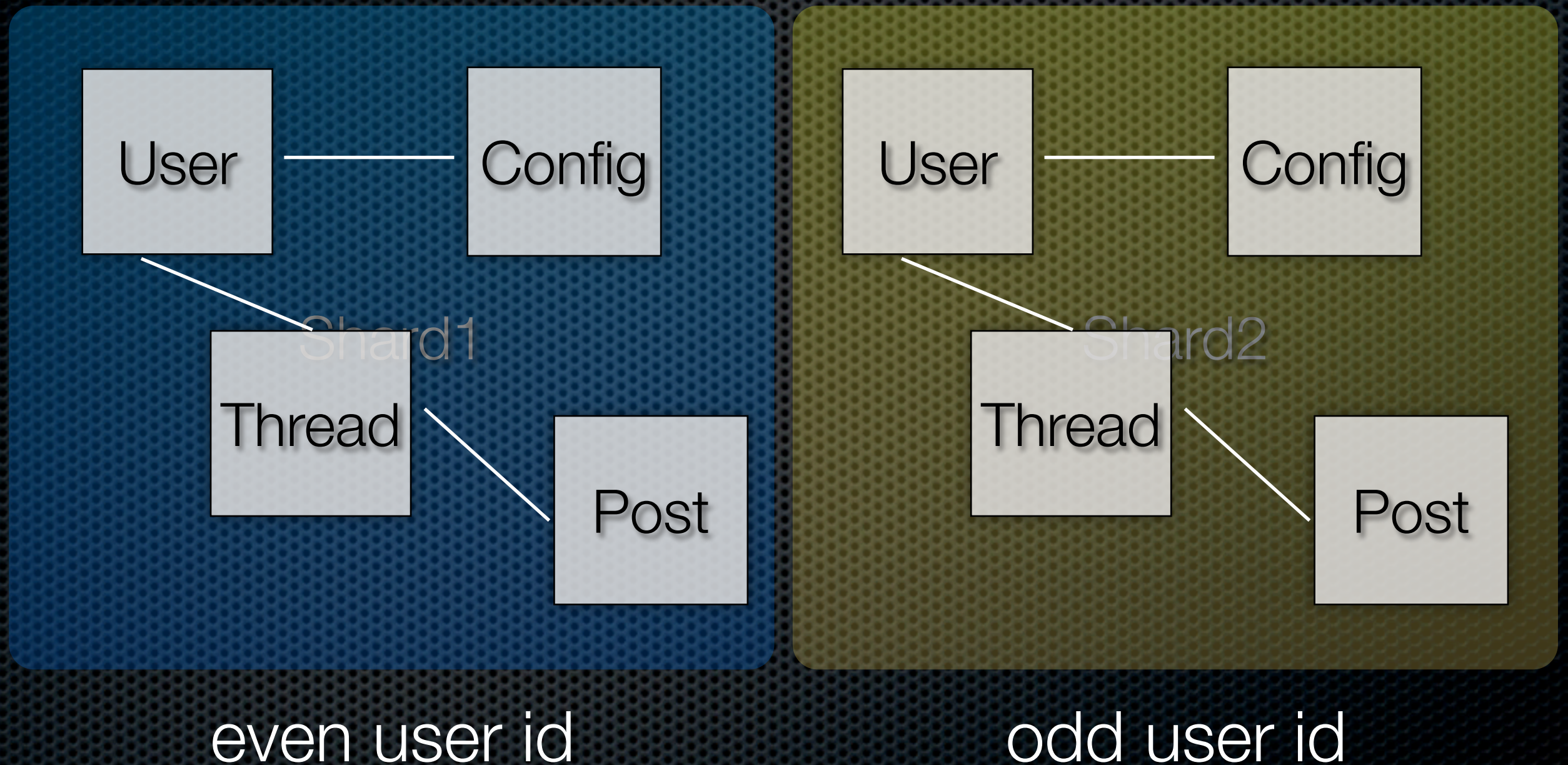
Splitting: Tables



Splitting: Data



Splitting: Data



Data distribution

Criteria

- ✦ Fast lookup
- ✦ Few database connections
- ✦ Inexpensive reorganization
- ✦ Equal distribution

Modulo

Shard
Uid: 1-9

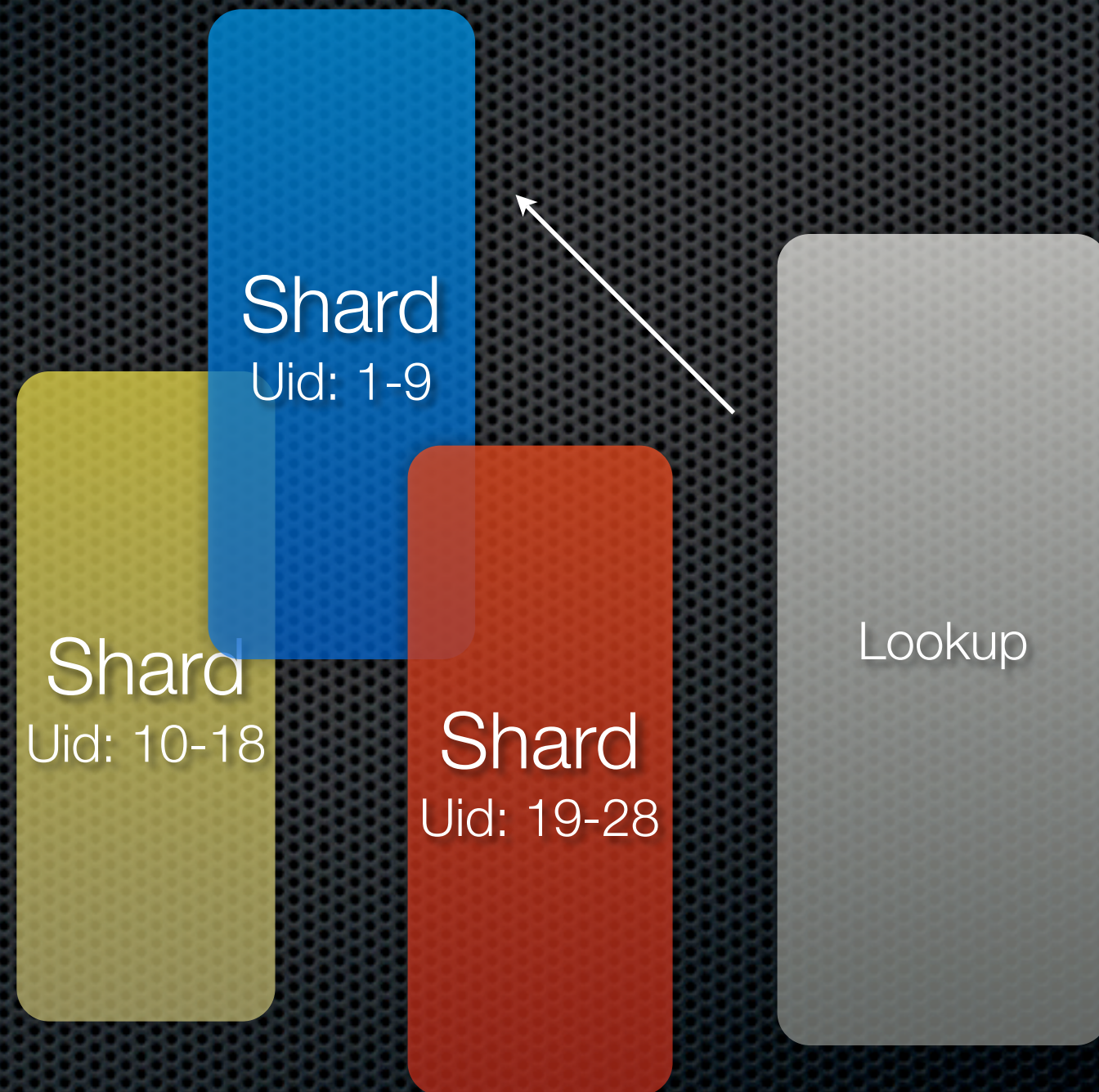
Shard
Uid: 10-19

Shard
Uid: 20-30

Modulo



Lookup table



Optimizing

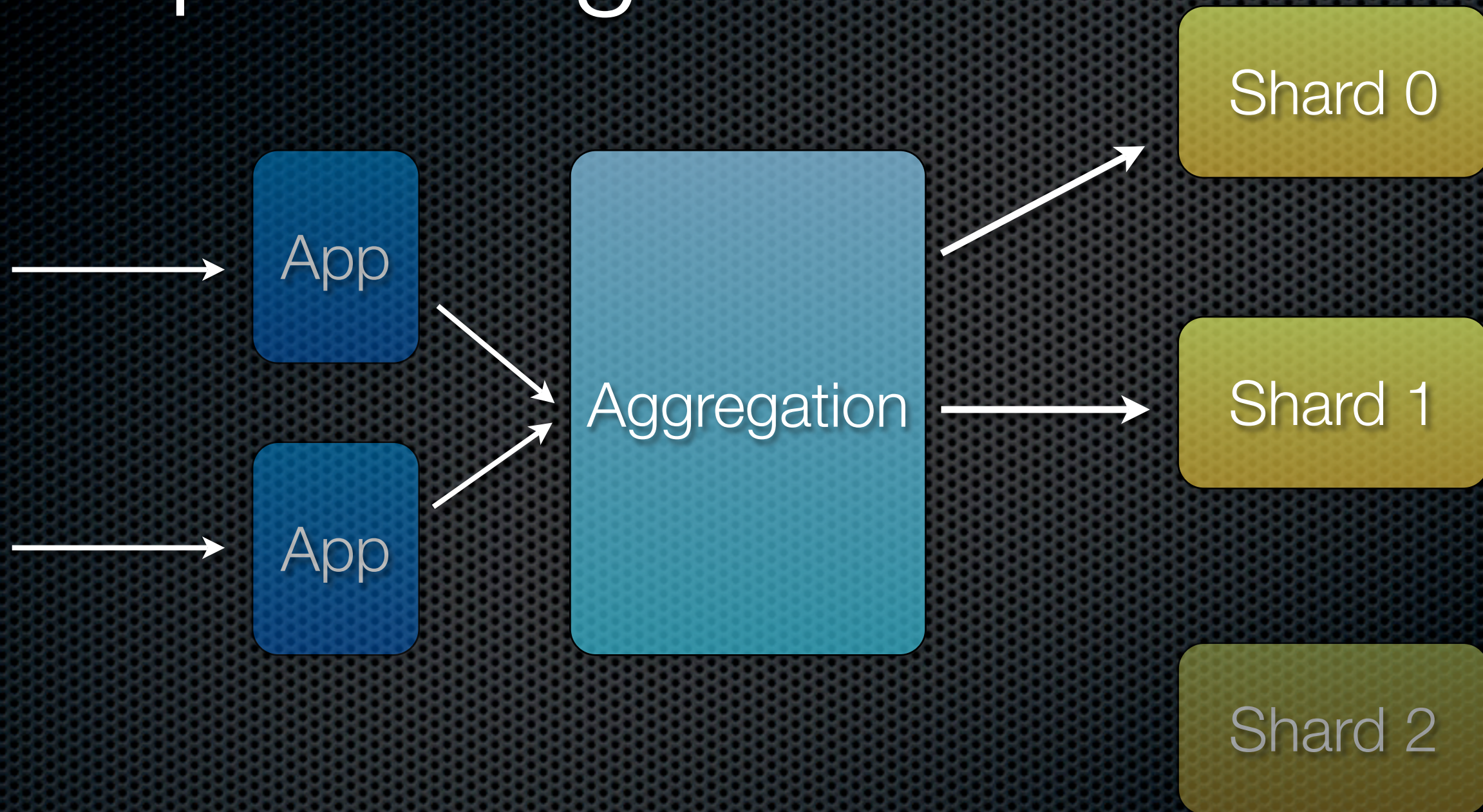
Optimizing

- ✦ caches (memcache)
- ✦ model optimization
 - ✦ normalization? no!
- ✦ connection pools
- ✦ persistent connections (using mysqli+mysqlnd)

Optimizing II

- ✧ aggregation daemons
 - ✧ persistent
- ✧ MySQL Proxy with LUA
- ✧ asynchronous queries (mysqlnd)

Optimizing III



Problems

- ✦ Joins, Unions, Intersections
- ✦ Grouping
- ✦ Selecting and projecting on groups
- ✦ Aggregation
- ✦ Primary keys
- ✦ Referential integrity (foreign keys)
- ✦ (De-)Normalization

Problems

- ✧ Global tables
 - ✧ Sharing
 - ✧ Tagging
 - ✧ Invitations
 - ✧ Search
- ✧ Lots of relations (due to normalization) between tables

Migration

- ✦ Find the bottleneck
- ✦ Refactoring of DB model needed?
- ✦ Prepare implementation
 - ✦ prepare DBA layer
 - ✦ write unit tests
 - ✦ split existing data

Migration II

- ✦ Setup shards
- ✦ Migrate your code
- ✦ Test, Test, Test
- ✦ Check if everything runs smooth
 - ✦ MySQL proxy
 - ✦ DTrace

Scale!



Some code

```
$conn->getConnection($userId)->query();
```

```
$conn->getGlobalConnection()->query();
```


Questions?

david.soriaparra@sun.com

<http://blog.experimentalworks.net>