MySQL @ tumblr

Bob Patterson Jr
Database Engineer @ tumblr
Percona Live 2014

MySQL at Tumblr

MySQL at Tumblr

- → 300+ machines
- → 71 Terabytes
- → ~300 Billion rows of data
- → 1¾ engineers

365 days ago

- → 213 machines
- → 40 Terabytes
- → 179 billion rows
- → 3 ¼ engineers

Layout

Layout

- → 81 shards
- → 8 function pools
- → 5 node classes
- → 3 hardware classes

Hardware Classes

- PCI flash machines
- Hardware Raid Hard Drives
- Web Class nodes

Database Classes

- Master
- Active Slave
- Standby Slave
- Backup Slave
- Util Database nodes (non user facing db)

Shards

- 1 master
- 1-2 Standby Slaves
- 0-1 Backup Slaves
- Depends on how hot the shard is
- PK is a clustered index with Tumblelog ID first

Shards

- posts
- notes
- likes
- tags
- themes
- .....

Functional Pools

- 1 master
- 0-8 Active slaves (taking reads)
- 2 Standby Slaves
- All of our non-sharded Data

Functional Pools

- Primary DB - original database small tables
- Tumblelogs and Users database
- Tags
- Post Ref (powers Dashboard)
- odds and ends (non-user facing)

Jetpants

What is Jetpants?

- → MySQL management suite
- → Command Line Tool
- → Ruby automation library
- → Developed by Tumblr (created by Evan Elias)
- → Open sourced in June 2012 - https://github.com/tumblr/jetpants

Ranged Sharding at Tumblr

Data moves around

- → Shards can be split into N new shards
- → Splits are non-disruptive
- → Shards can me merged to join under-utilized shards (new feature)

Ranged Based Sharding

- min and max id range
- Ranges can be any size
- Easy to generate config file
- Each Shard Contains the same tables
- Different sliver of the overall dataset

Benefits of Ranged Sharding

- → Low shard count
- → we do not pre-allocate thousands of small shards
- → Shards created on demand
- → Shard splits (and now shard merges) defrag the data

Pitfalls of Ranged Sharding

→ Uneven load on your shards

→ Uneven usage of disk capacity

Jetpants Command Line Tools

- → jetpants pools
- → jetpants summary
- → jetpants clone_slave
- → jetpants shard_split
- → jetpants promotion
- → jetpants shard_merge

jetpants clone slave

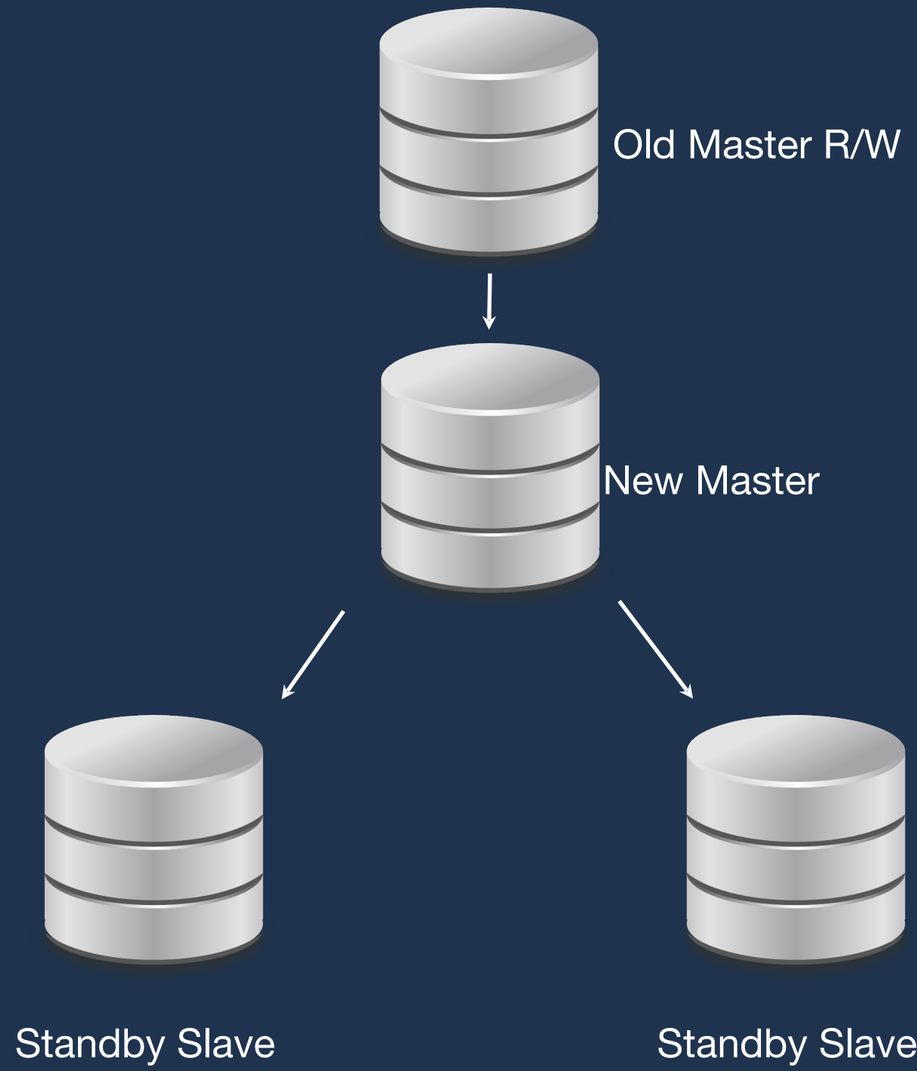
- → Utilizes a standby slave
- → Stops mysqld and copies over files
- → Uses tar, nc and compression with qpress (you can set your own compression)
- → Can create multiple slaves by chaining them with tee and fifo on intermediate nodes (uses full up and down link)

jetpants removing masters

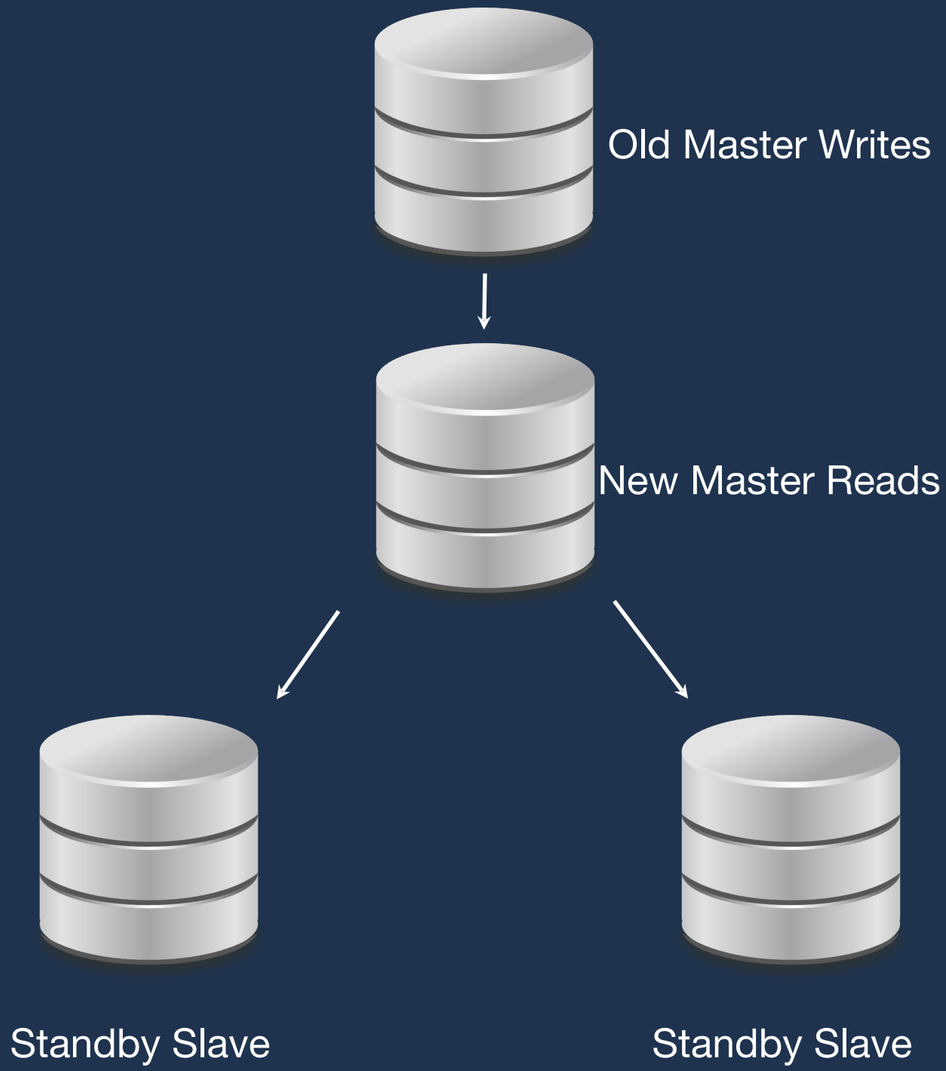
- → Lockless
- → Point reads to new master
- → Point writes to new master
- → No auto_increment tables

jetpants removing masters

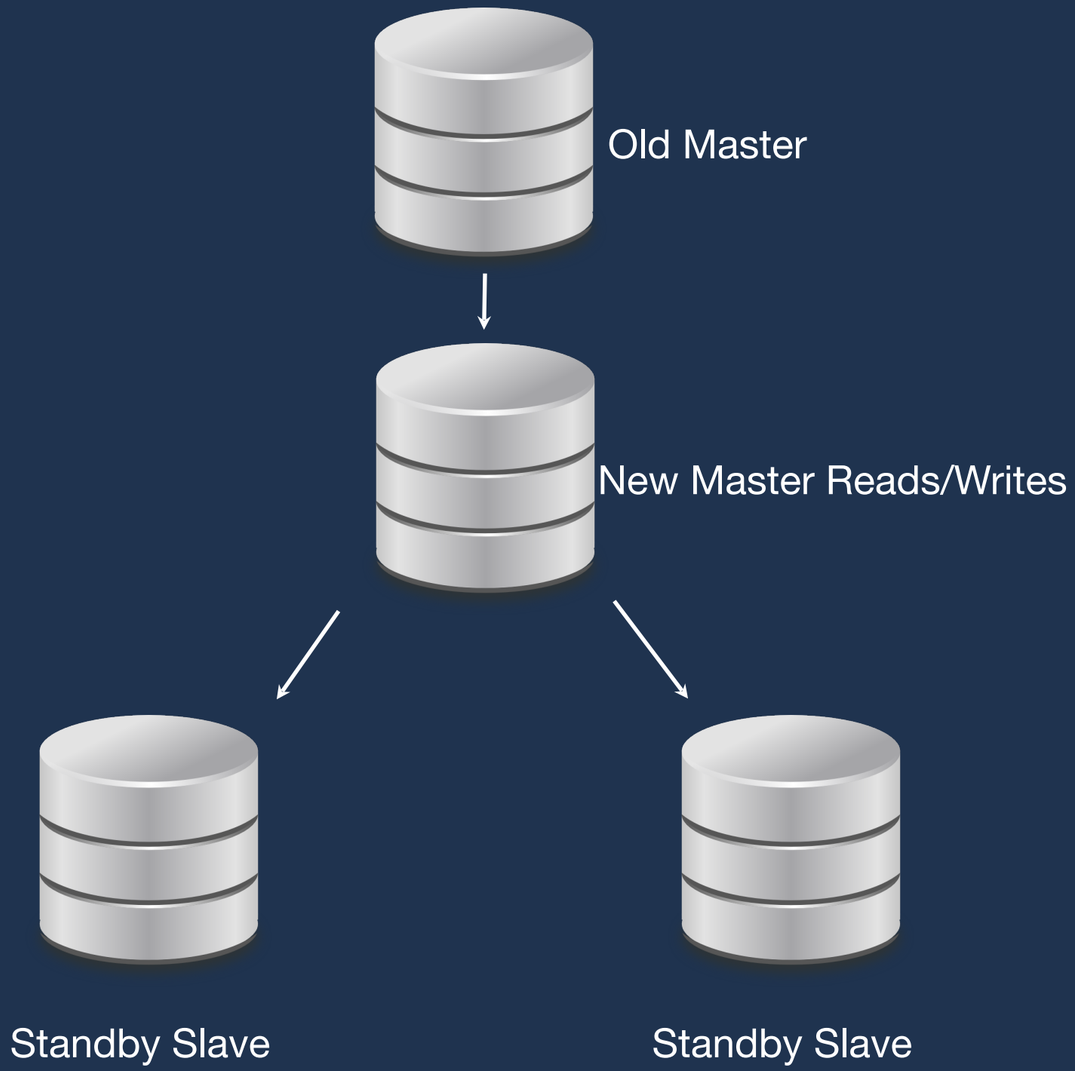
jetpants removing masters



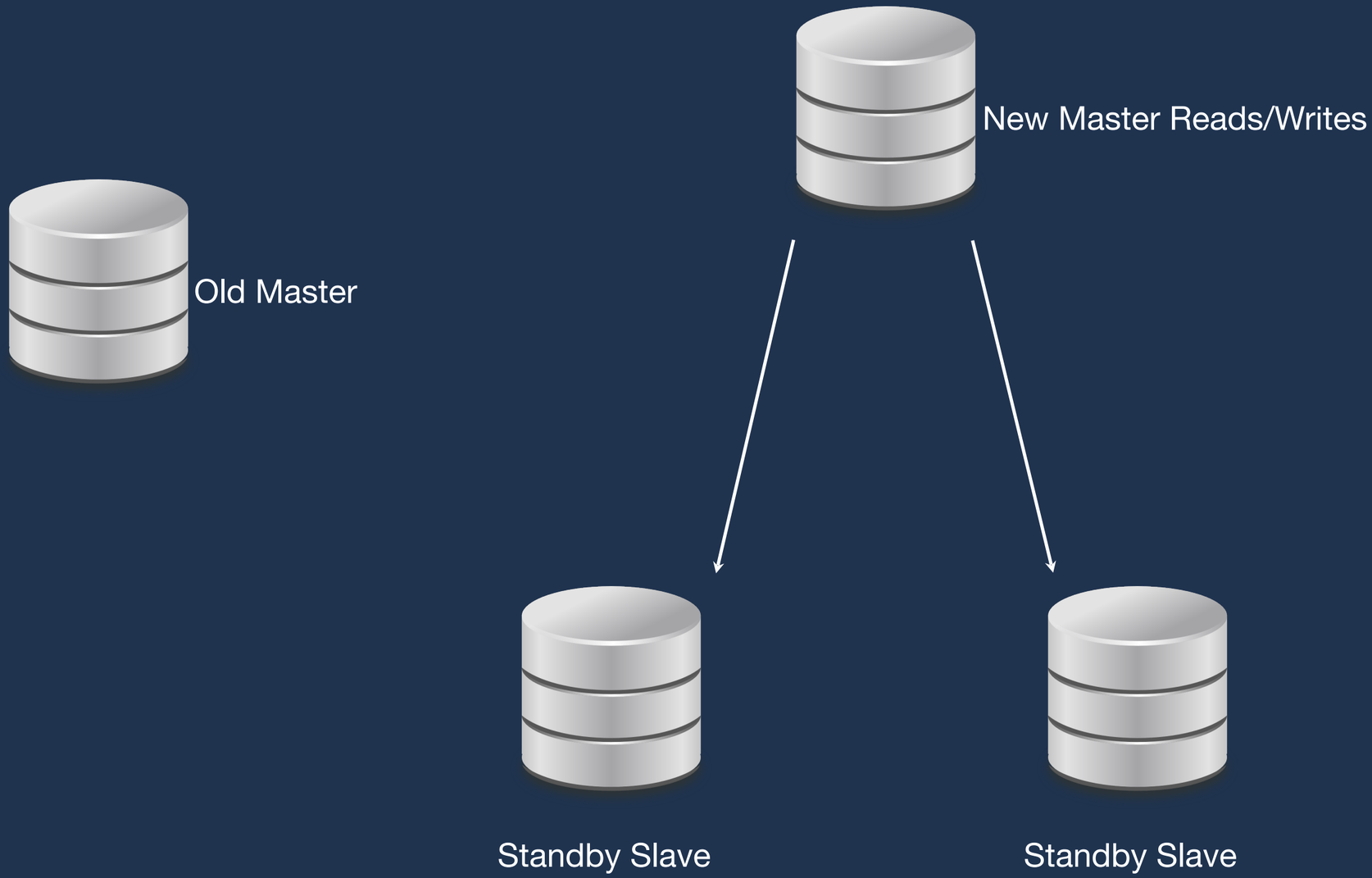
jetpants removing masters



jetpants removing masters



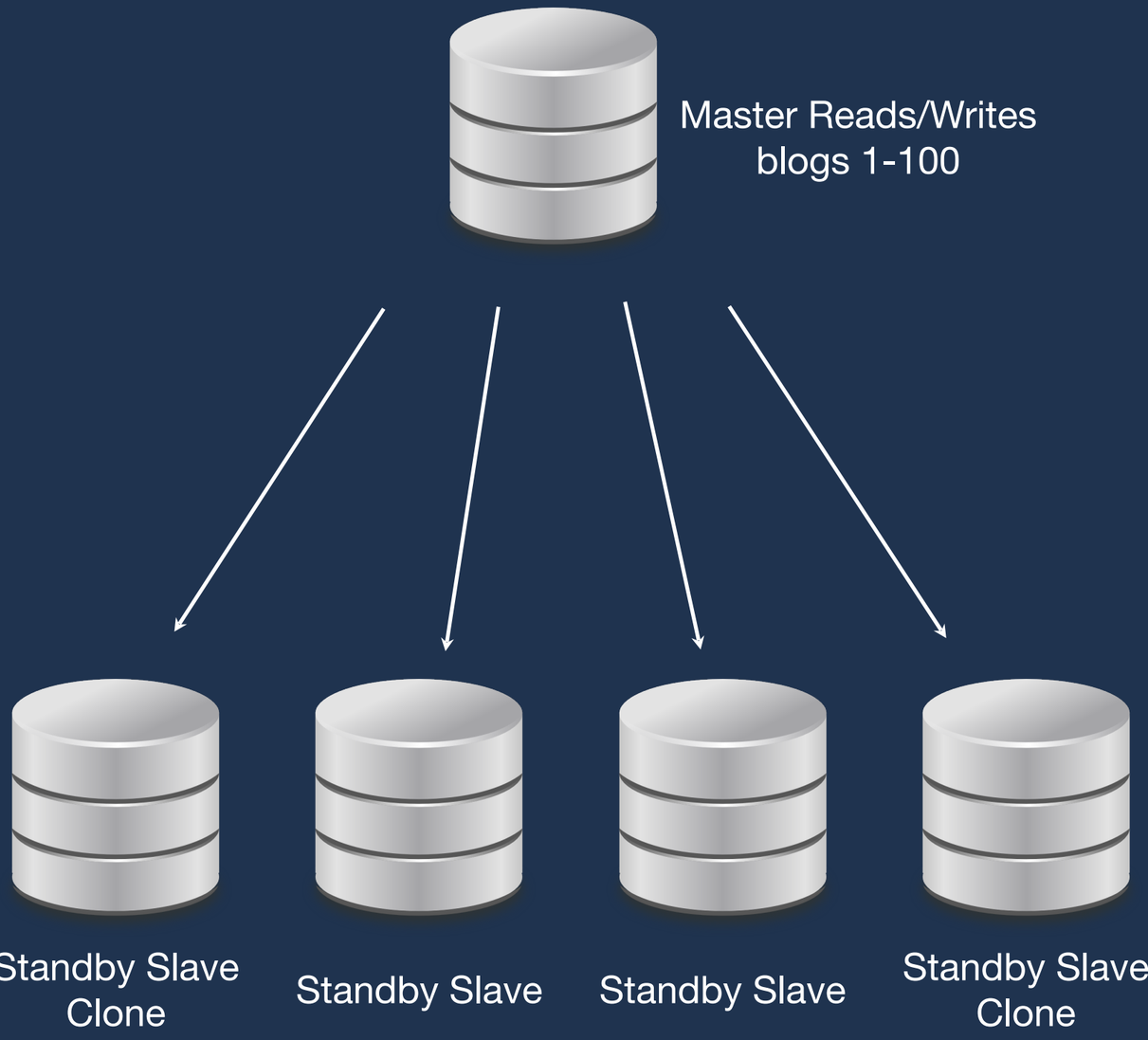
jetpants promotions



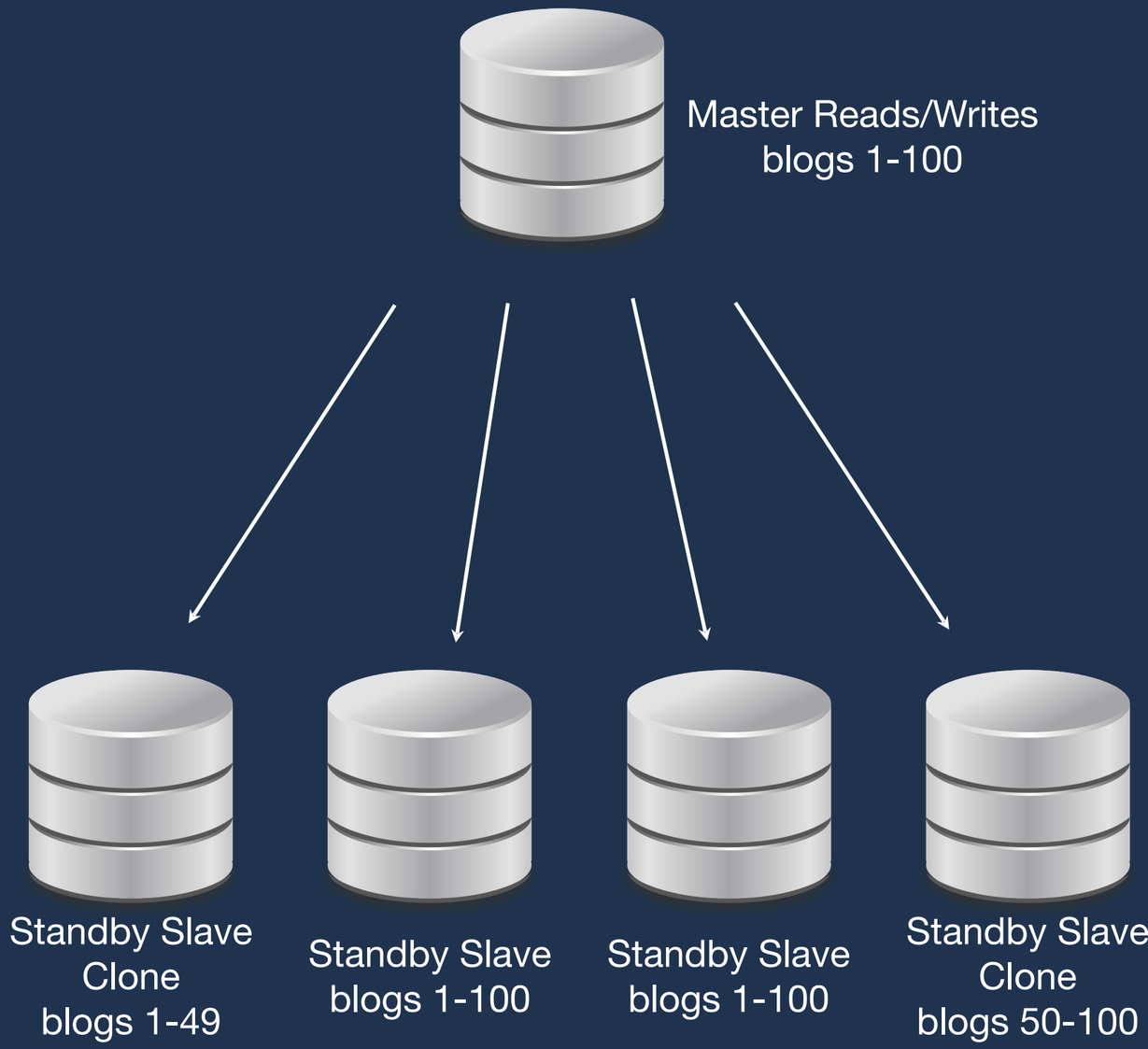
jetpants shard splits

- → Clone one new slave for each new shard
- → Export dataset that will live on new shards
- → Drop the tables and recreate tables
- → Import data
- → Replay replication stream from parent shard
- → Remove data from other shards

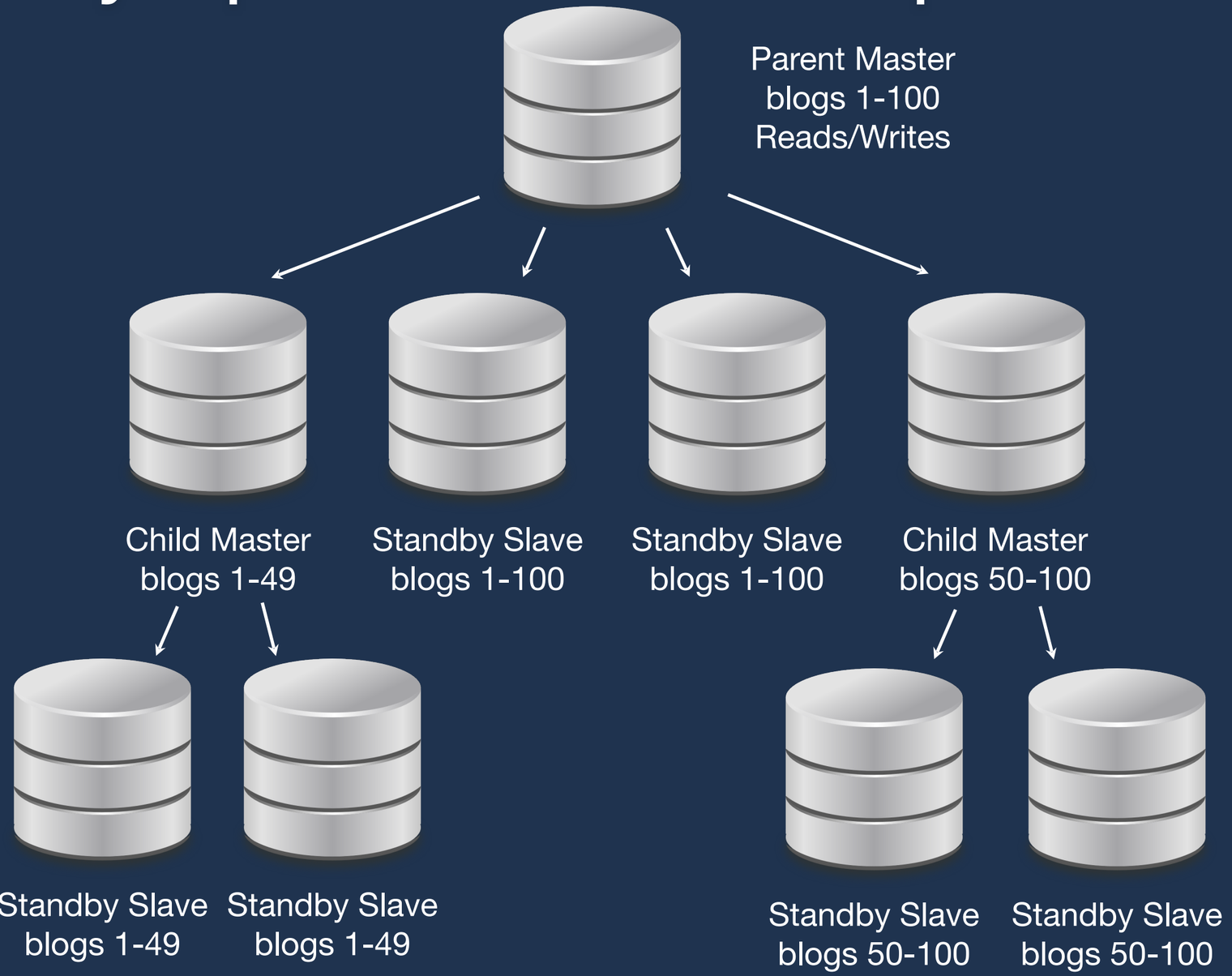
jetpants shard splits



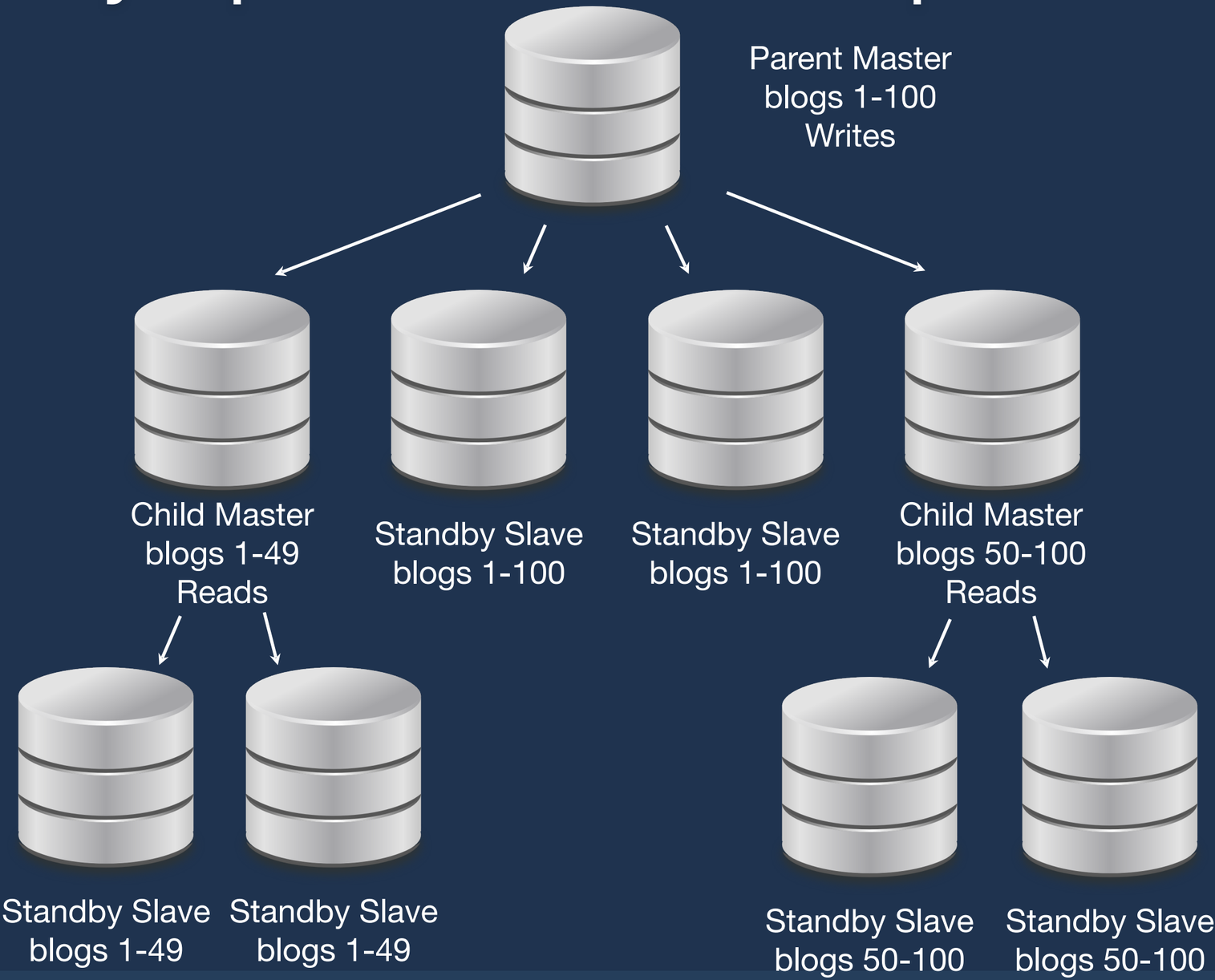
jetpants shard splits



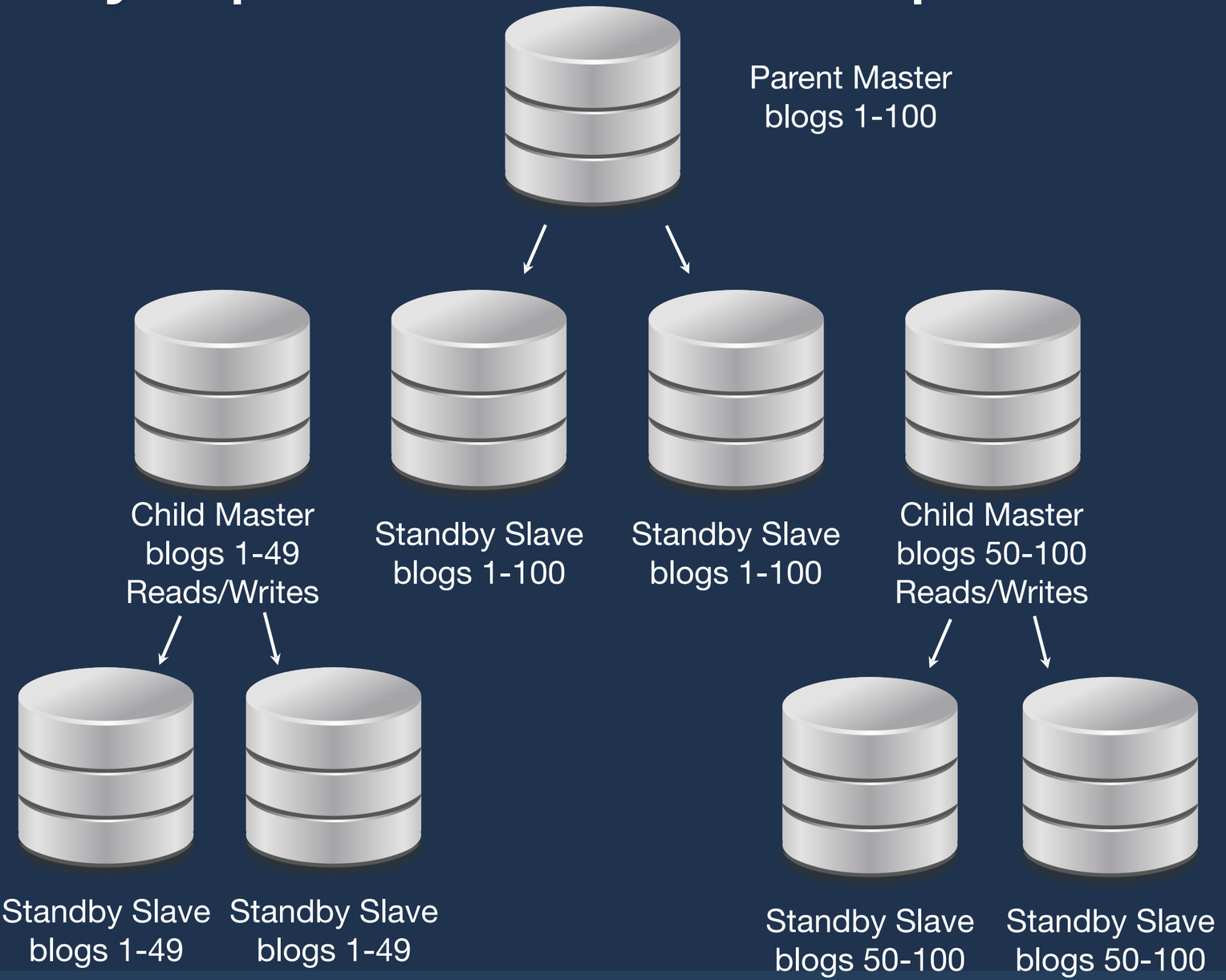
jetpants shard splits



jetpants shard splits



jetpants shard splits



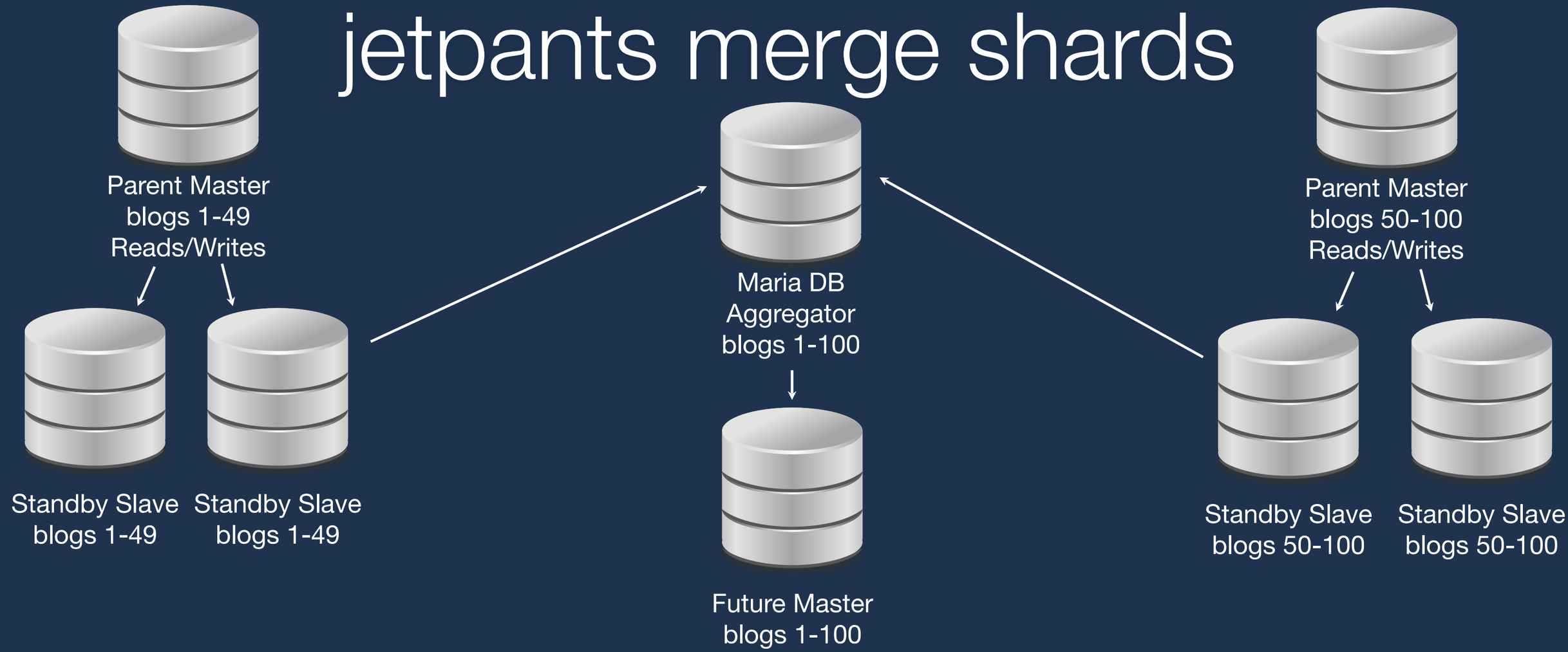
jetpants shard splits

jetpants merge shards

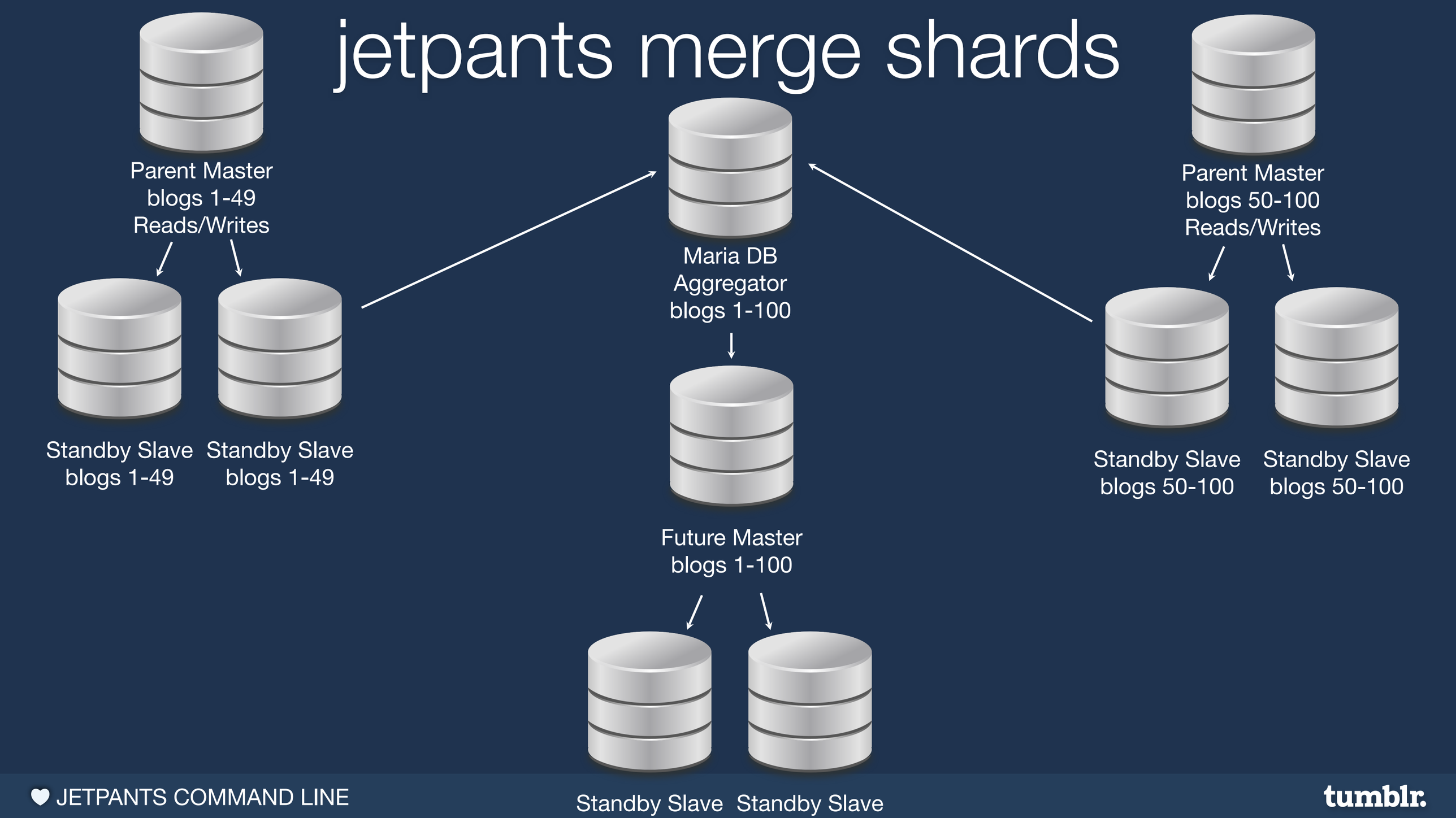
- → Mariadb as an aggregator node (for its multi-source replication)
- → Export data from each shard
- → Import data into aggregator node and new master at same time
- → Clone new slaves from new master
- → Pt-query-checksum during testing and a validator for each live run

jetpants merge shards

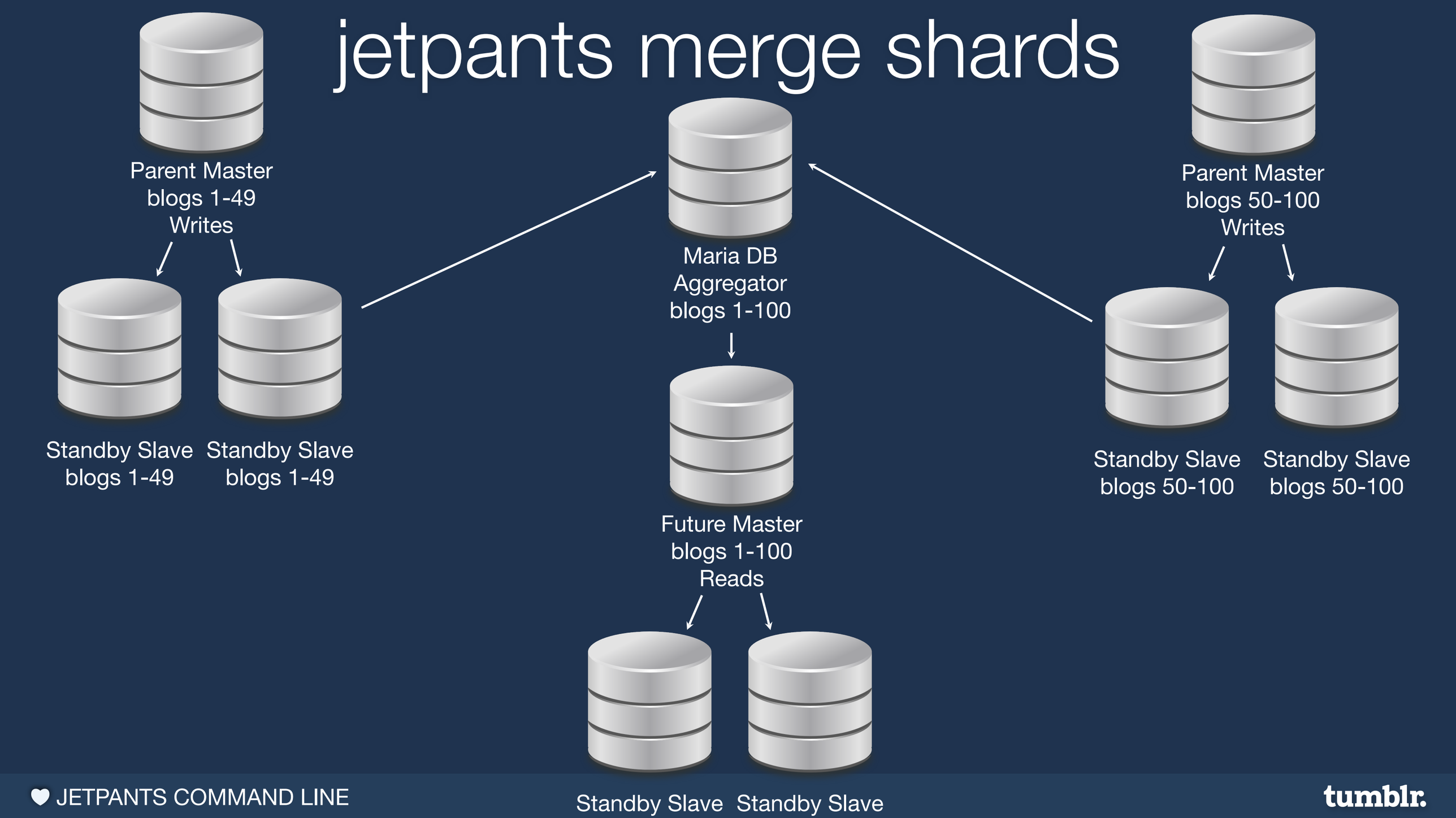
jetpants merge shards



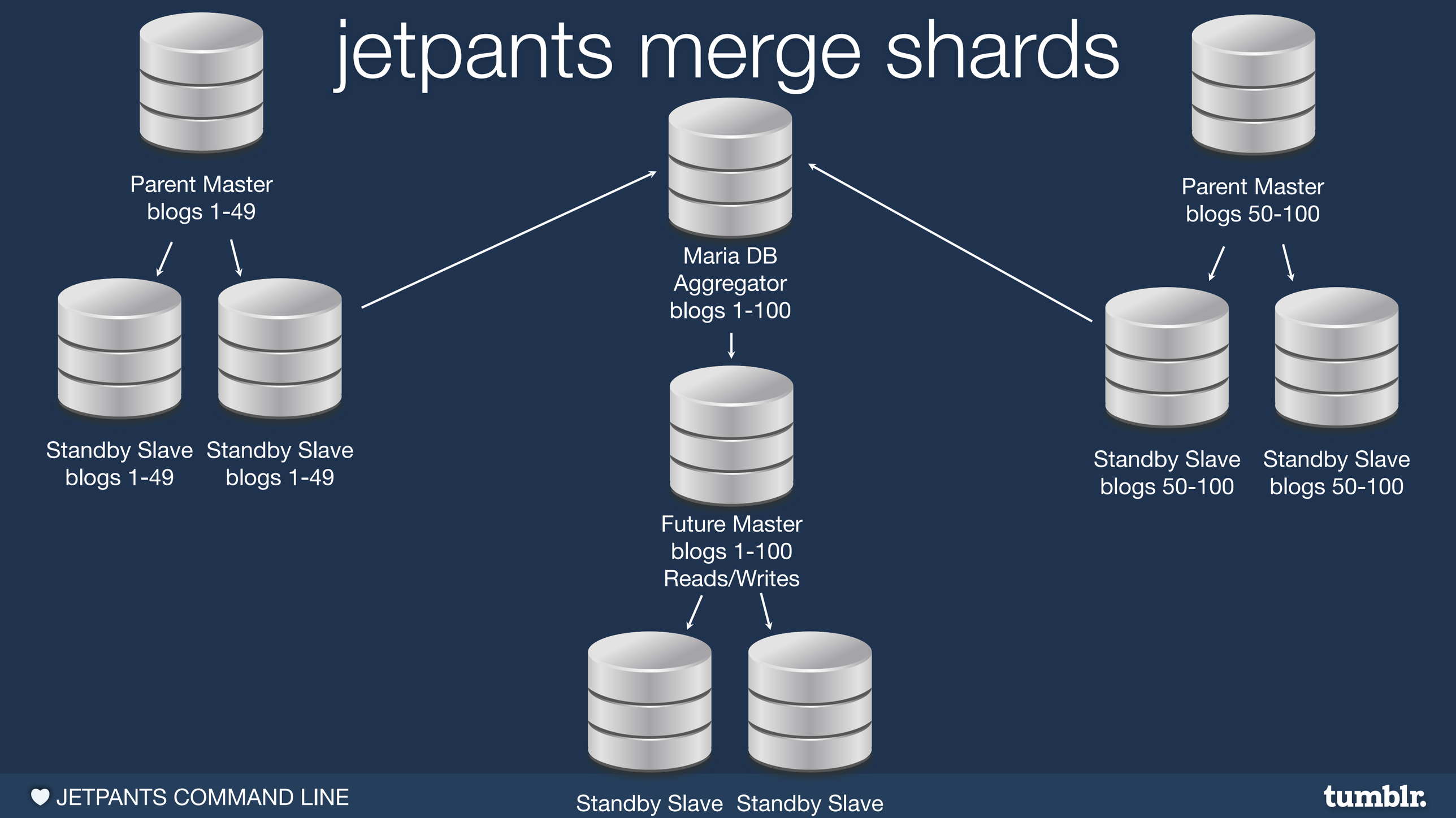
jetpants merge shards



jetpants merge shards



jetpants merge shards



jetpants merge shards

jetpants alter table

- → Wraps Percona's pt-online-schema-change
- → Tracks alters via jetpants
- → Does live alter table on master
- → No downtime
- → Shard command to do the alter on all shards

jetpants capacity plan

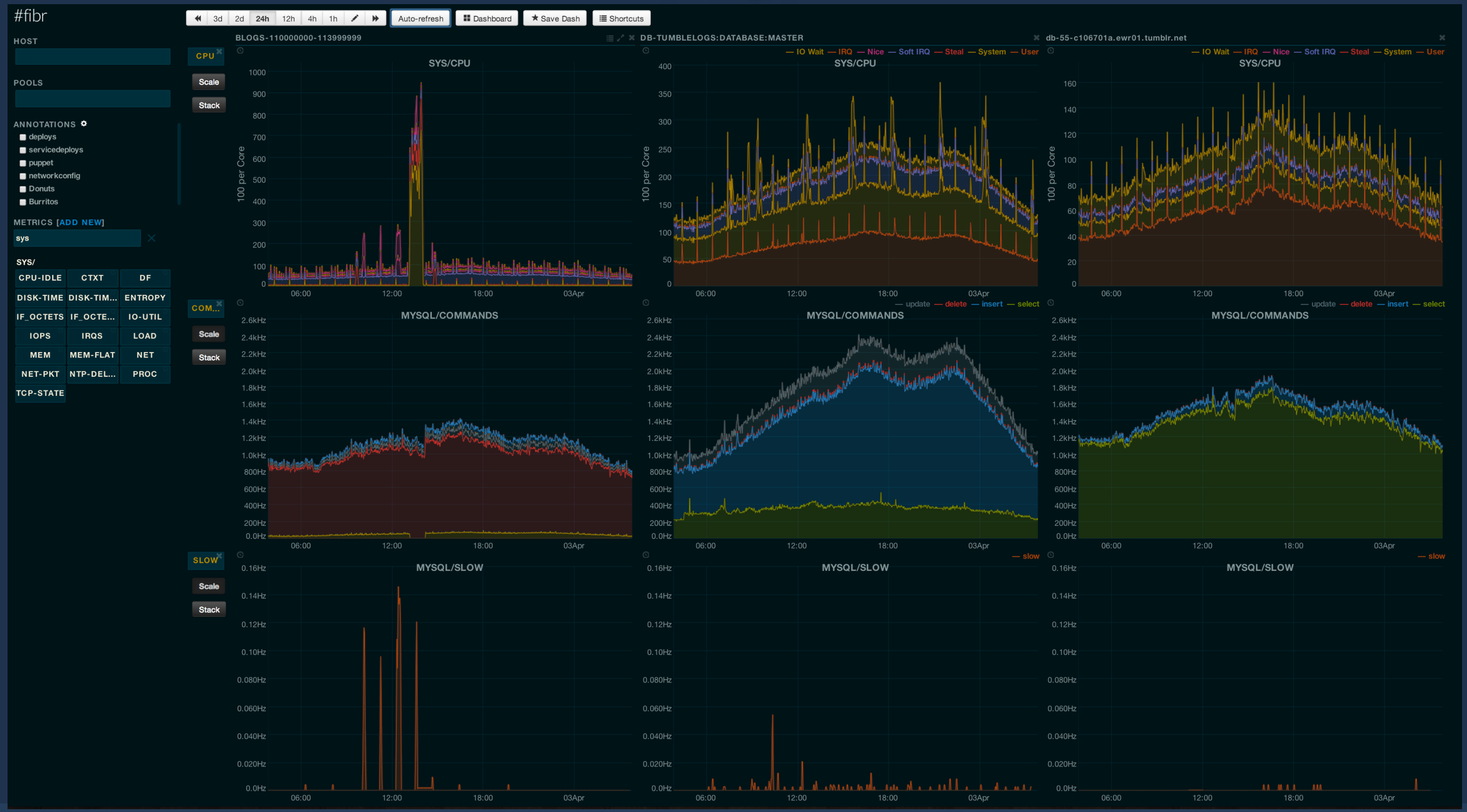
- → Tracks data usage on all pools
- → Calculates growth rates
- → Finds outliers for abnormally high growth
- → Rollups of all your hardware usage and states
- → Emails reports every morning

jetpants capacity plan

```
Notices

The following pools have to many standby slaves :

db-tumblelogs

Warning Mounts

blogs-10000000-10999999

Usage and Time Left
 --------- The 'GB per day' and 'Days left' fields are using a growth rate that is calulated by taking
 --------- a exponically decaying avg

                    pool name    Current Data Size GB per day  Days left (until critical)
                    ---------    ----------------- ----------  ---------
                    postsreff              238.82       4.11     237.55
                      primary              344.42       1.91     456.27
                blogs-1-999999              593.63       0.56        N/A
          blogs-999999-infinity               48.12       3.33     350.53

Day Over Day

                    pool name      today  1 day ago 2 days ago 7 days ago 14 days ago
                    ---------      -----  --------- ---------- ---------- -----------
                dbm-postsreff       4.24       4.17       3.78       4.55        4.74
                      primary       1.69       2.07       2.29       0.56        0.53
                blogs-1-999999      0.53       0.58       0.58       0.68        1.01
          blogs-999999-infinity     3.61       3.33       2.87       2.81        0.00

New Outliers
--Compare the last 3 days in 2 hour blocks to the same 2 hour block 7, 14, 21, 28 days ago

                    Pool Name              Start Time                End Time      Usage  Prev Weeks
                    ---------              ----------                --------      -----  ----------
                      primary      09/20/2013 00:02:19      09/20/2013 02:04:19       2.25        0.26

Hardware status

            status        databasenode      database55node     utildatabasenode               total
  ------------------  ------------------  ------------------  ------------------  ------------------
             allocated                   0                 247                   6                 253
                 spare                   0                   6                   2                   8
               claimed                   0                   0                   0                   0
           maintenance                   0                   7                   0                   7
           provisioned                   0                   1                   0                   1
          provisioning                   0                   3                   0                   3

Total Unallocated nodes - 0
```

Other Tools

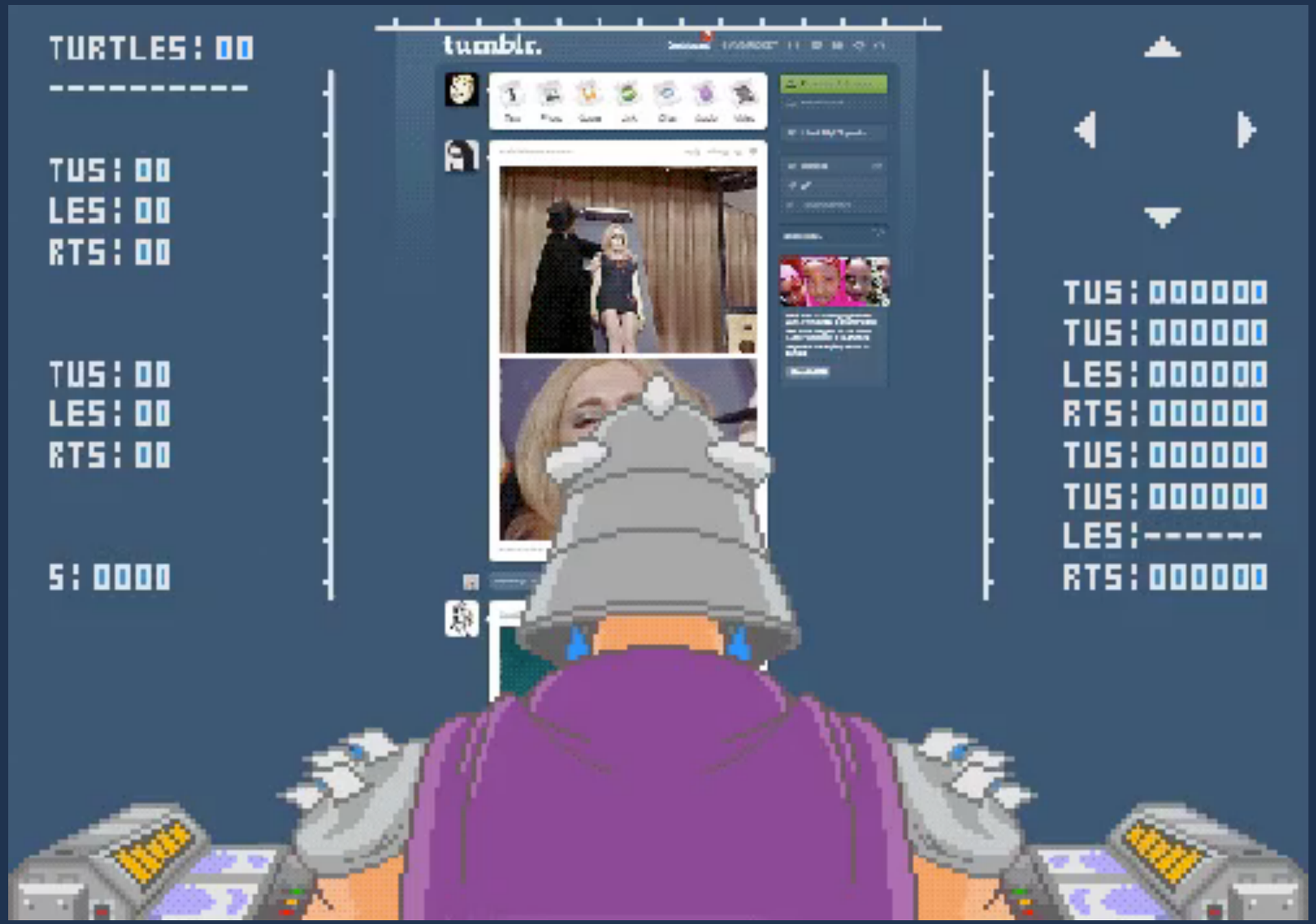
Fibr


#fibr
HOST
POOLS
ANNOTATIONS
deploys
servicedeploys
puppet
networkconfig
Donuts
Burritos
METRICS [ADD NEW]
sys
SYS/
CPU-IDLE
CTXT
DF
DISK-TIME
DISK-TIM...
ENTROPY
IF_OCTETS
IF_OCTE...
IO-UTIL
IOPS
IRQS
LOAD
MEM
MEM-FLAT
NET
NET-PKT
NTP-DEL...
PROC
TCP-STATE
3d
2d
24h
12h
4h
1h
Auto-refresh
Dashboard
Save Dash
Shortcuts
BLOGS-110000000-113999999
DB-TUMBLELOGS:DATABASE:MASTER
db-55-c106701a.ewr01.tumblr.net
CPU
Scale
Stack
SYS/CPU
IO Wait
IRQ
Nice
Soft IRQ
Steal
System
User
100 per Core
COM...
MYSQL/COMMANDS
update
delete
insert
select
SLOW
MYSQL/SLOW
slow
06:00
12:00
18:00
03Apr

Also

- → OpenTSDB
- → percona toolkit
- → Collins (github.com/tumblr/collins)
- → pt-online-schema-change
- → Leasing api
- → ...

Dashboard


TURTLES: 00
TUS: 00
LES: 00
RTS: 00
TUS: 00
LES: 00
RTS: 00
S: 0000
tumblr.
TUS: 000000
TUS: 000000
LES: 000000
RTS: 000000
TUS: 000000
TUS: 000000
LES:------
RTS: 000000

Post Creation

- → User Creates a post
- → Post inserted into blog's shard
- → Post inserted into Post Ref
- → Last Post Time table updated (blog associated)

What is Post Ref?

- 3 column table
- tumblelog id, timestamp, post id
- every public post on tumblr
- spread across 3 pools, ~3.3 TB

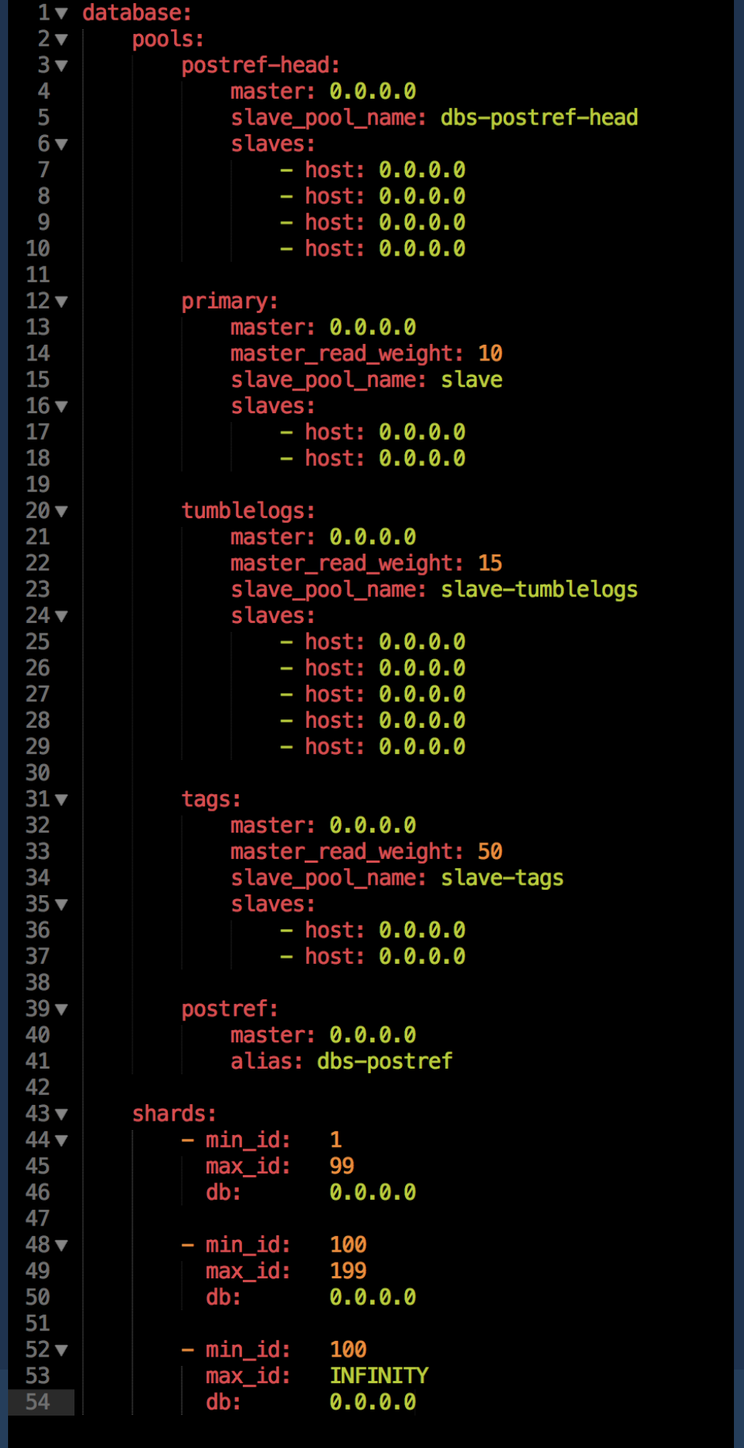
Loading the Dashboard

- → Retrieve the blogs you are following
- → Find the most recent x blogs that posted
- → Query Post Ref for x number of posts
- → Fetch posts
- → Is there a next page

Config

Config

- → Generated by jetpants
- → uses Collins to determine state
- → anyone can create their own config hooks in jetpants



```yaml
database:
    pools:
        postref-head:
            master: 0.0.0.0
            slave_pool_name: dbs-postref-head
            slaves:
                - host: 0.0.0.0
                - host: 0.0.0.0
                - host: 0.0.0.0
                - host: 0.0.0.0

        primary:
            master: 0.0.0.0
            master_read_weight: 10
            slave_pool_name: slave
            slaves:
                - host: 0.0.0.0
                - host: 0.0.0.0

        tumblelogs:
            master: 0.0.0.0
            master_read_weight: 15
            slave_pool_name: slave-tumblelogs
            slaves:
                - host: 0.0.0.0
                - host: 0.0.0.0
                - host: 0.0.0.0
                - host: 0.0.0.0
                - host: 0.0.0.0

        tags:
            master: 0.0.0.0
            master_read_weight: 50
            slave_pool_name: slave-tags
            slaves:
                - host: 0.0.0.0
                - host: 0.0.0.0

        postref:
            master: 0.0.0.0
            alias: dbs-postref

    shards:
        - min_id:   1
          max_id:   99
          db:       0.0.0.0

        - min_id:   100
          max_id:   199
          db:       0.0.0.0

        - min_id:   100
          max_id:   INFINITY
          db:       0.0.0.0
```

♥ CONFIG

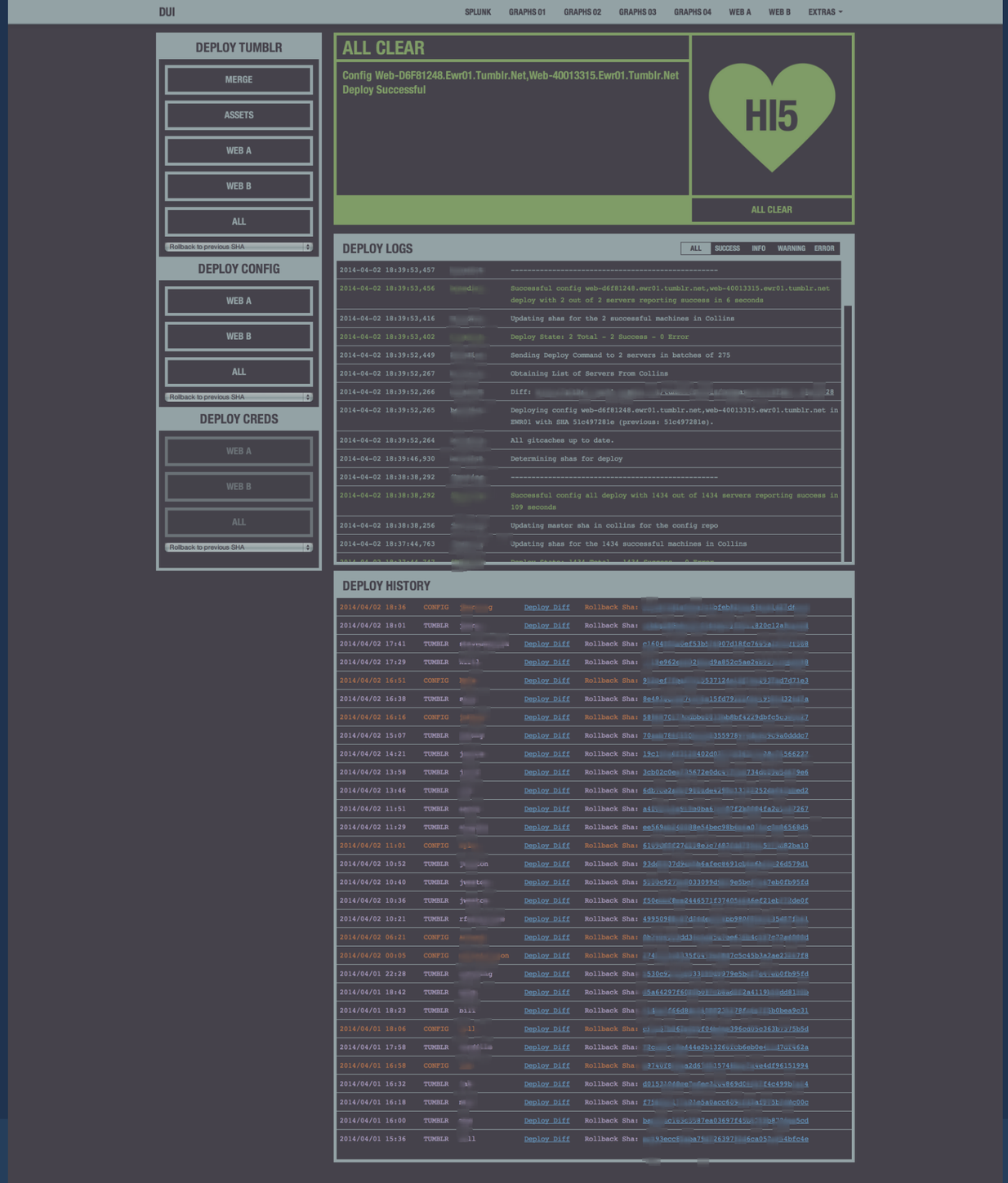
tumblr.

Deploys

Deploys

- → Internal deploy interface (cli and web)
- → Func does actual pushing of deploys
- → using internal git and gitcache boxes

DUI



DEPLOYS

tumblr.

Memcache

Memcache

- → 8½ TB of cache
- → 3.4 Million get/second
- → 200,000 sets/second
- → 23 Billion Items

Memcache

- → Write through caching
- → item caching for the most part
- → heavily stress multi get pre-fetching

tumblr.com/jobs

Q & A