Forget everything you knew about Swift Rings

(here's everything you need to know about Rings)

Your Ring Professors

- Christian Schwede
  - Principal Engineer @ Red Hat
  - Stand up guy
- Clay Gerrard
  - Programmer @ SwiftStack
  - Loud & annoying

Rings 201

- **Why Rings Matter**
- **What are Rings**
- **How Rings Work**
- How to use Rings
- Ninja SWIFT RING Tricks
- MOAR Awesome Stuff

Swift 101

Looking for more general intro to Swift?

- Swift 101: https://youtu.be/vAEU0Ld-GIU
- Building webapps with Swift: https://youtu.be/4bhdqtLLCiM
- Stuff to read: https://www.swiftstack.com/docs/introduction/openstack_swift.html

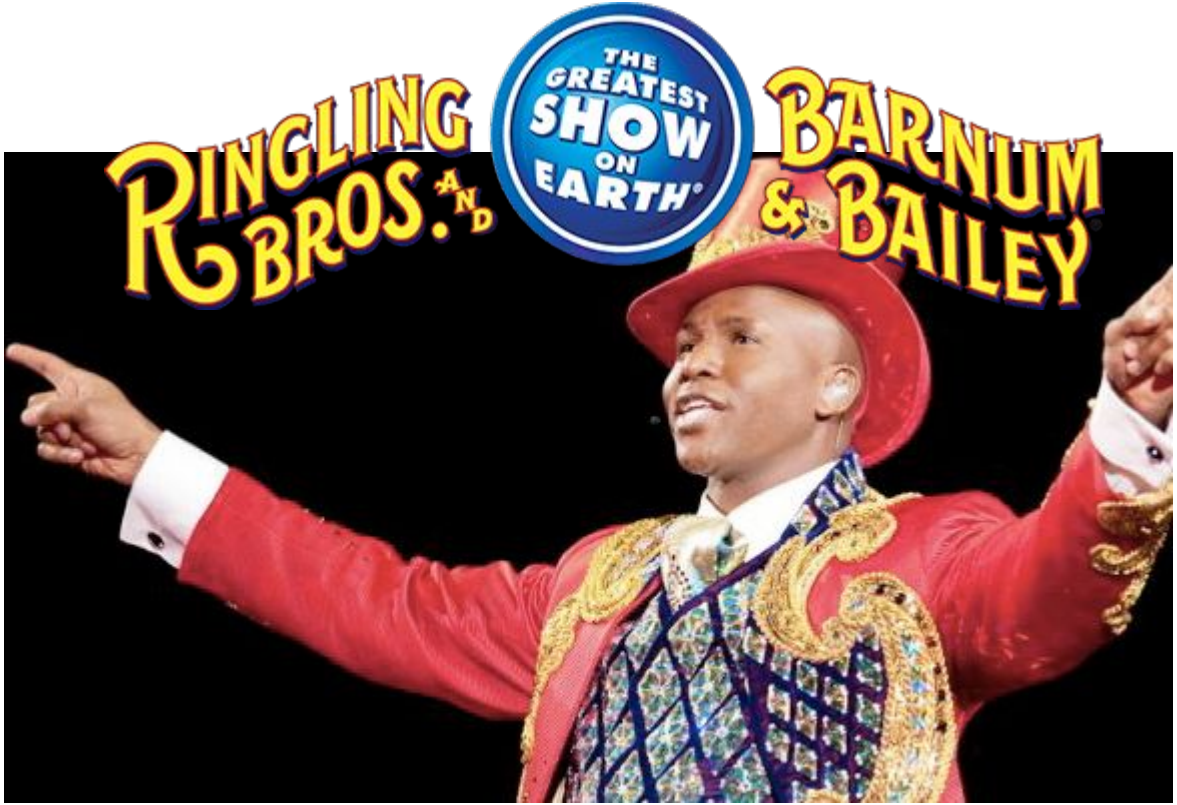
OR SIX

~~One Ring~~ To Rule Them All

Swift Developers

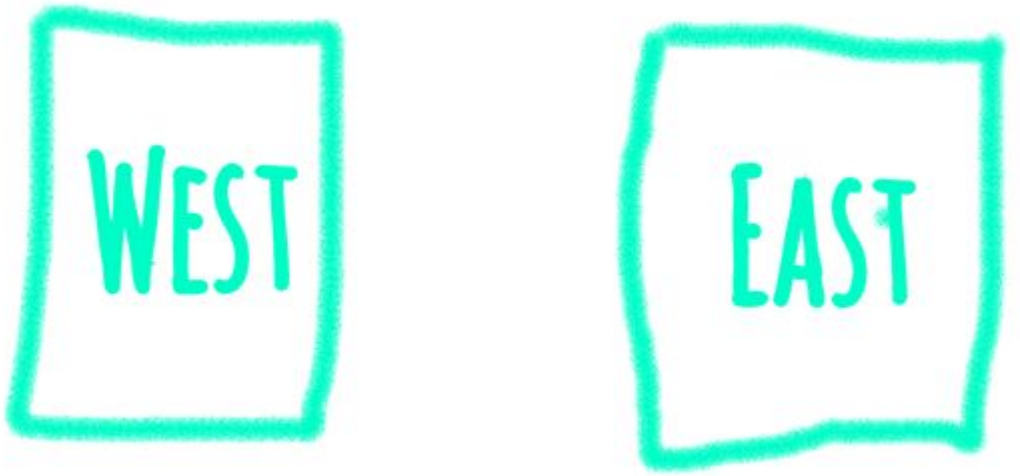
Can be a wild ride

Ring Masters



Ring Features

- Devices & servers
- Zones
- Regions
  - Multi-Region
  - Cross-Region
  - Local-Region
- Storage Policies



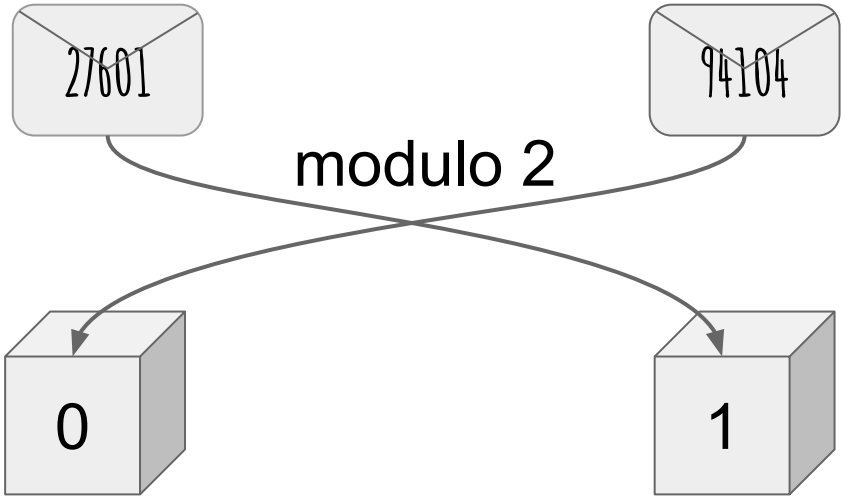
Swift's Rings use Simple Concepts

Consistent Hashing introduced by Karger et al. at MIT in 1997

The Same Year HTTP/1.1 is specified in RFC 2616

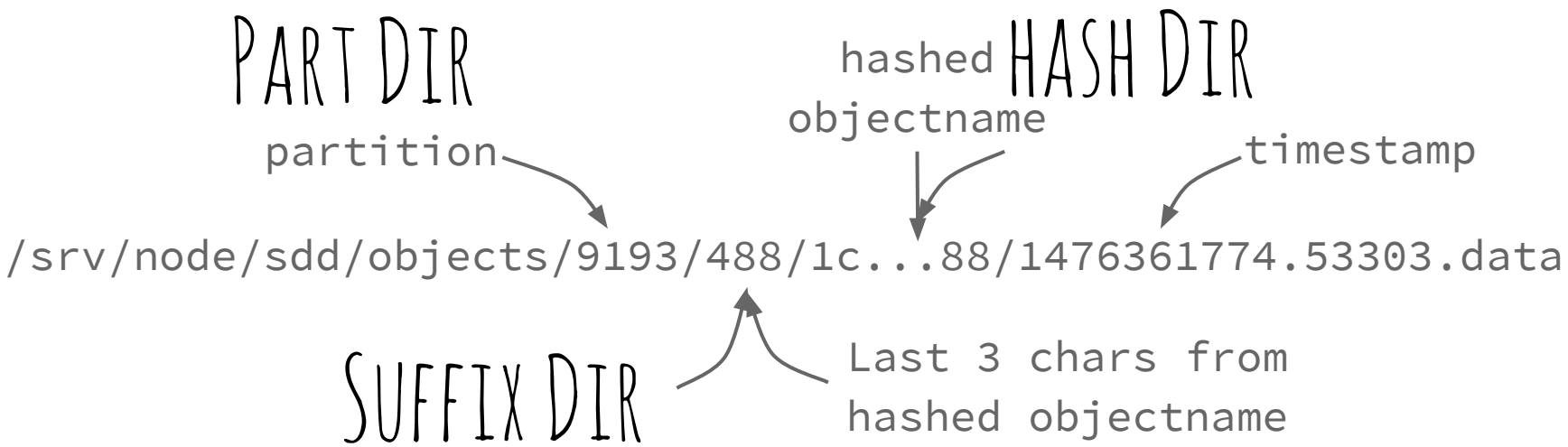
Consistent what?

- Just remember the distribution function
- No growing lookup tables!
- Easy to distribute!



Partitions in Swift

- Object namespace is mapped to a number of **partitions**
- Each partitions holds one or more objects



replica2part2dev_id

	Replica # 1	Replica # 2	Replica # 3
Part # 0	Device # 0	Device # 1	Device # 3
Part # 1	Device # 3	Device # 0	Device # 1
Part # 2	Device # 3	Device # 4	Device # 2
Part # 3	Device # 2	Device # 0	Device # 1
Part # 4	Device # 1	Device # 4	Device # 3
Part # 5	Device # 0	Device # 2	Device # 4
Part # ...	...	...	**...**

How to lookup partition

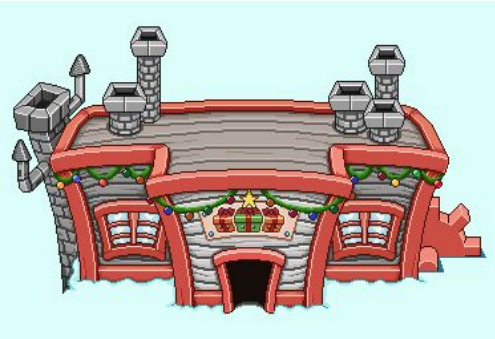
Primary

`get_nodes(part)`

Part # 2	Device # 3	Device # 4	Device # 2

Handoff

`get_more_nodes(part)`



What makes a good ring

A good ring has good

- Dispersion
- Balance
- Low overload (some, but not too much!)

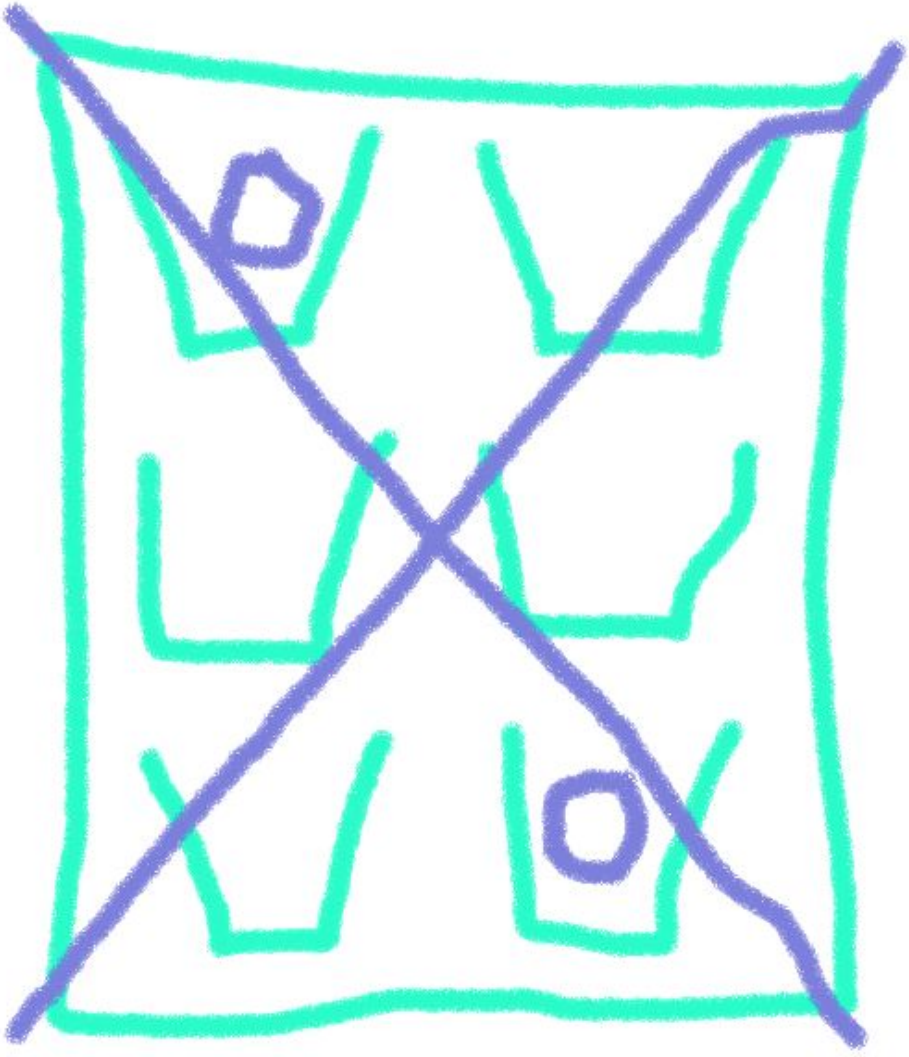
PC LOAD LETTER

Fundamental Constraints

- **Devices (disks)**
- **Servers**
- **Zones (racks)**
- **Regions (datacenters)**

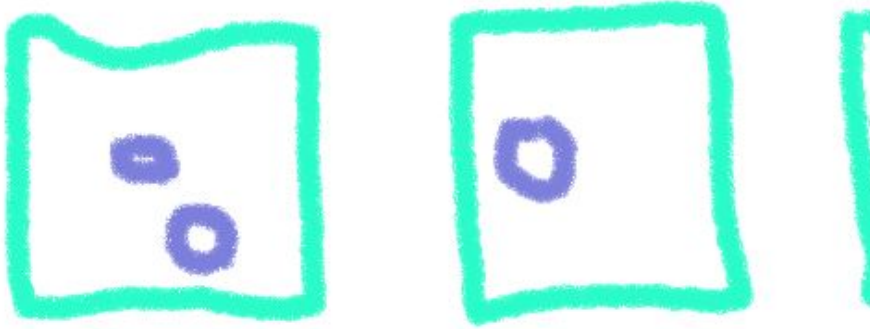
A Failure Domain fails together

These are tiers



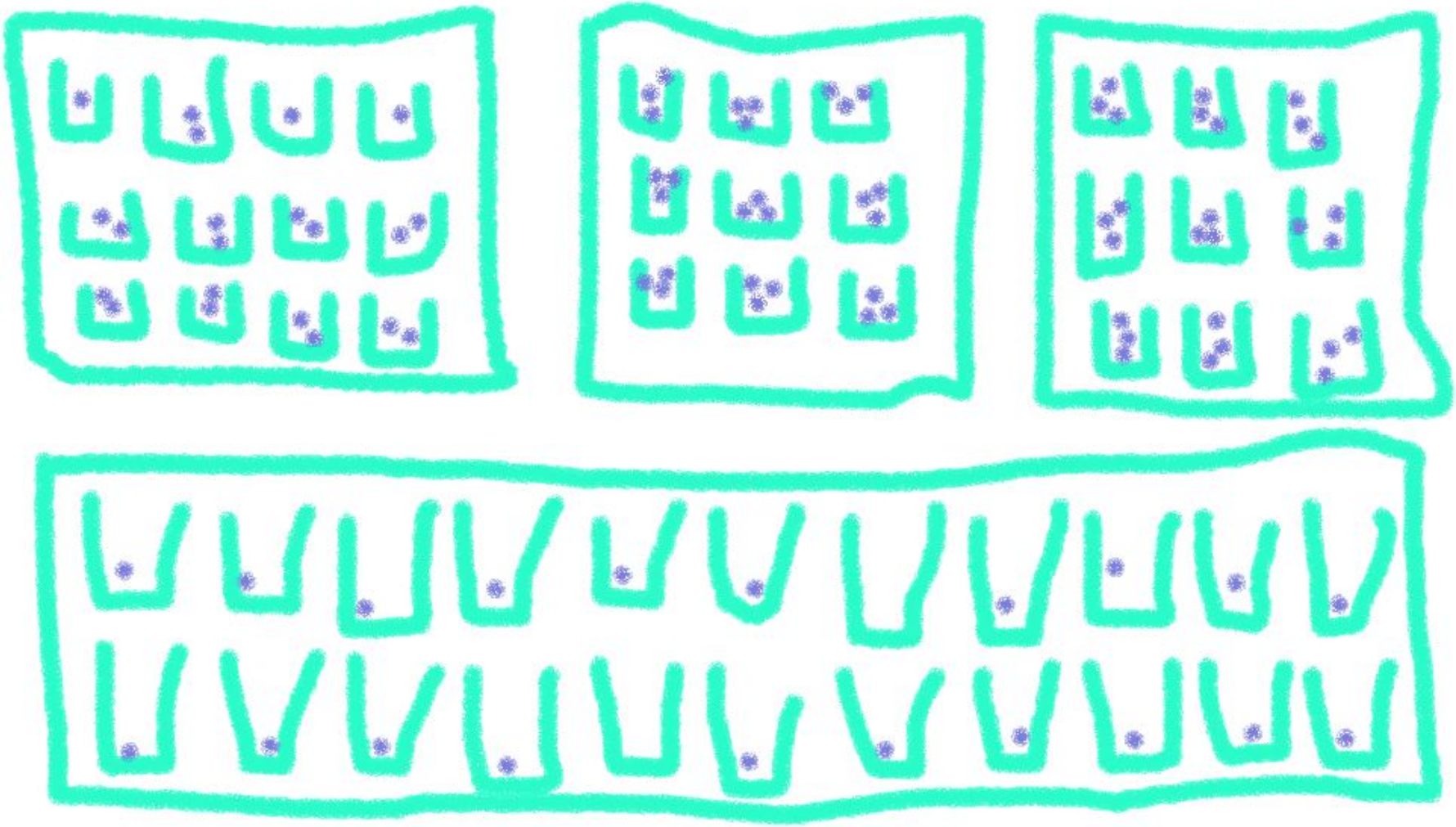
Dispersion

Measurement that the Failure Domain of each Replica of a Part is unique as possible



Fundamental Constraints

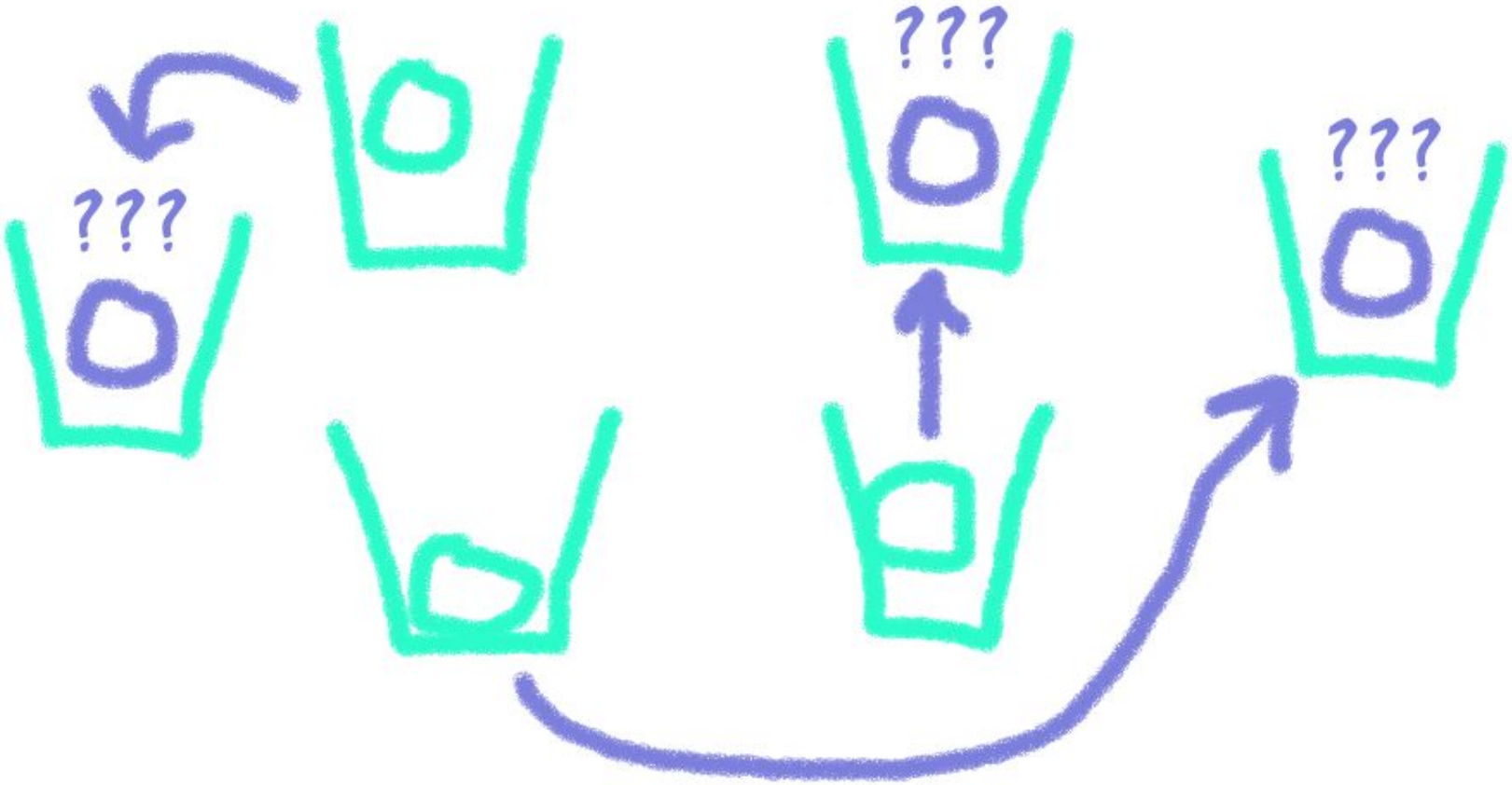
BALANCE



The Rebalance Process

"RINGS ARE NOT PIXIE DUST THAT MAGIC DATA OFF OF HARD DRIVES"

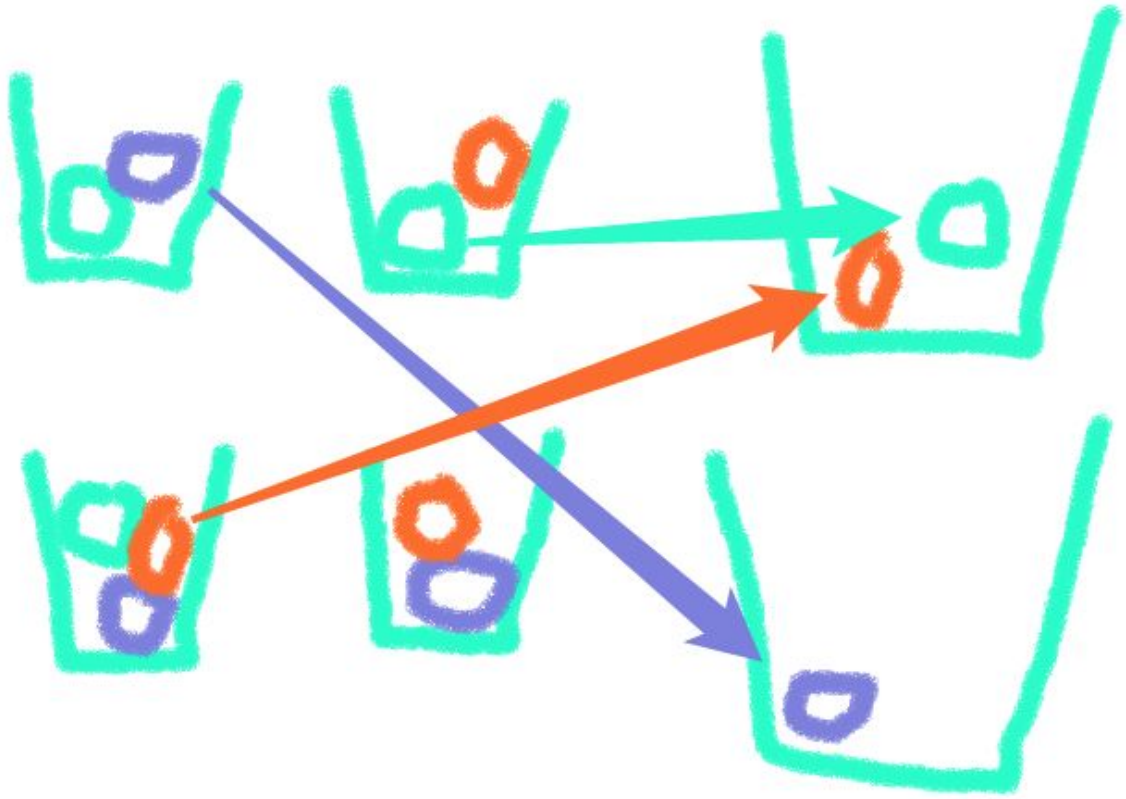
-- DARRELL



Fundamental Constraints

min_part_hours

Only move one replica of a partition per rebalance



Monitoring Replication Cycle

- Only rebalance after a full replication cycle
- swift-dispersion-report is your friend

```
Queried 8192 objects for dispersion reporting, ...
There were 3190 partitions missing 0 copy.
There were 5002 partitions missing 1 copy.
79.65% of object copies found (19574 of 24576)
```

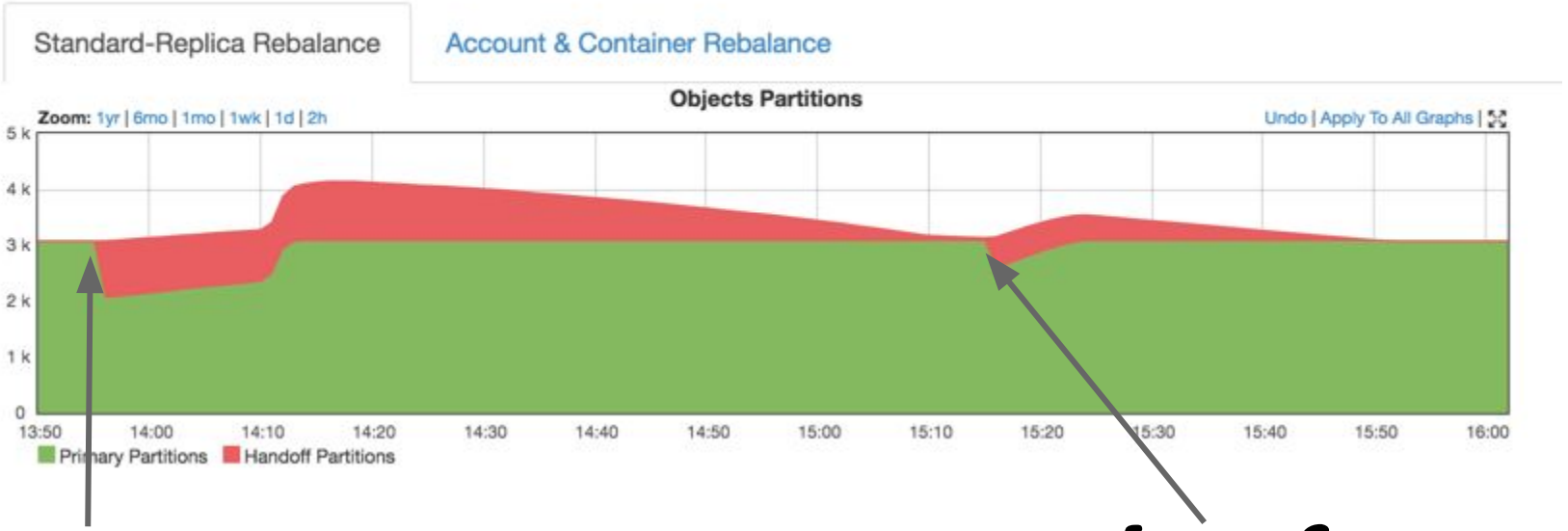
Hostname	Role	Zone	Storage
ss-rax-geo-syd-node1	Swift Node Account & Container, Standard-Replica **(Default)**	SYD Zone 1	12G 1K
ss-rax-geo-syd-node2	Swift Node Account & Container, Standard-Replica **(Default)**	SYD Zone 1	12G 1K
ss-rax-geo-syd-node3	Swift Node Account & Container, Standard-Replica **(Default)**	SYD Zone 1	12G 1K
ss-rax-geo-dfw-node2	Swift Node Account & Container, Standard-Replica **(Default)**	DFW Zone 1	9G 3G 2K
ss-rax-geo-dfw-node1	Swift Node Account & Container, Standard-Replica **(Default)**	DFW Zone 1	9G 3G 2K
ss-rax-geo-dfw-node3	Swift Node Account & Container, Standard-Replica **(Default)**	DFW Zone 1	9G 3G 2K

PATITIONS ASSIGNED

GB USED

STARTING TO FILL!

Rebalance Status



RING PUSH

Primary Partitions

Handoff Partitions

FIRST CYCLE FINISHED

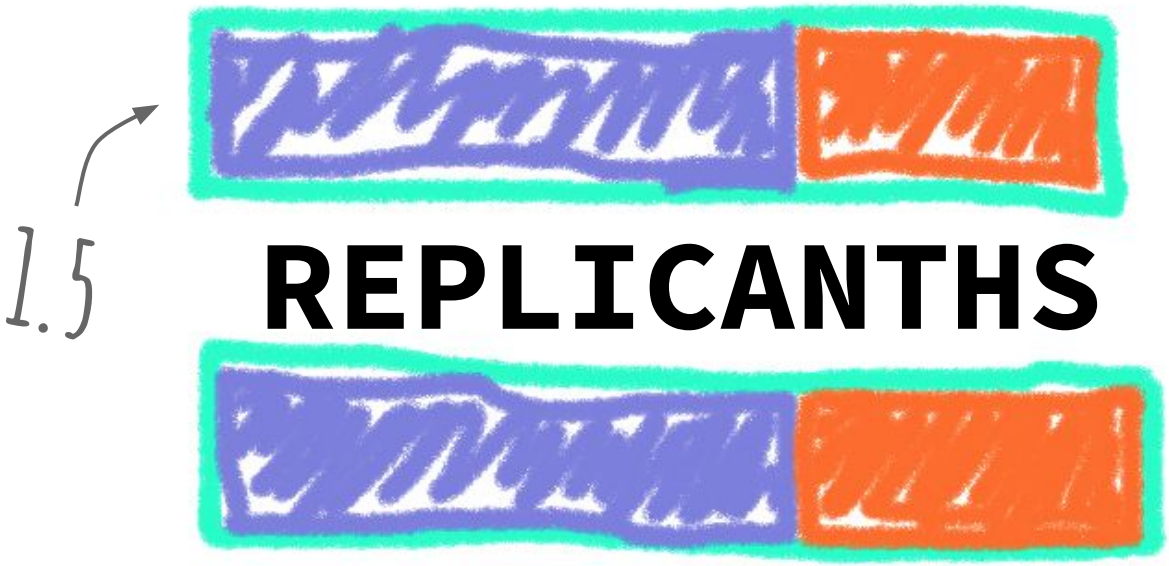
OVERLOAD

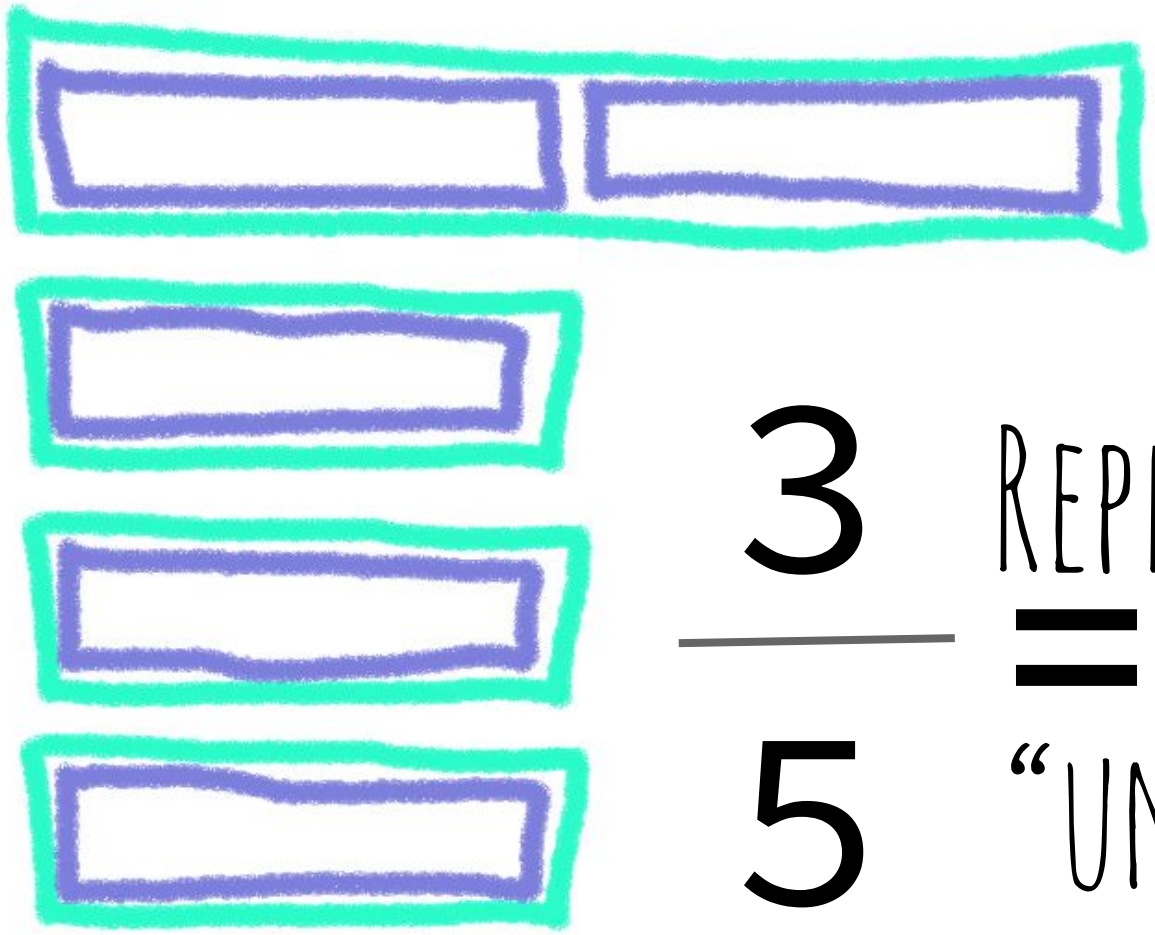
Balance vs. Dispersion

9.99999

FIGHT!

The decimal fraction of one replicas worth of partitions

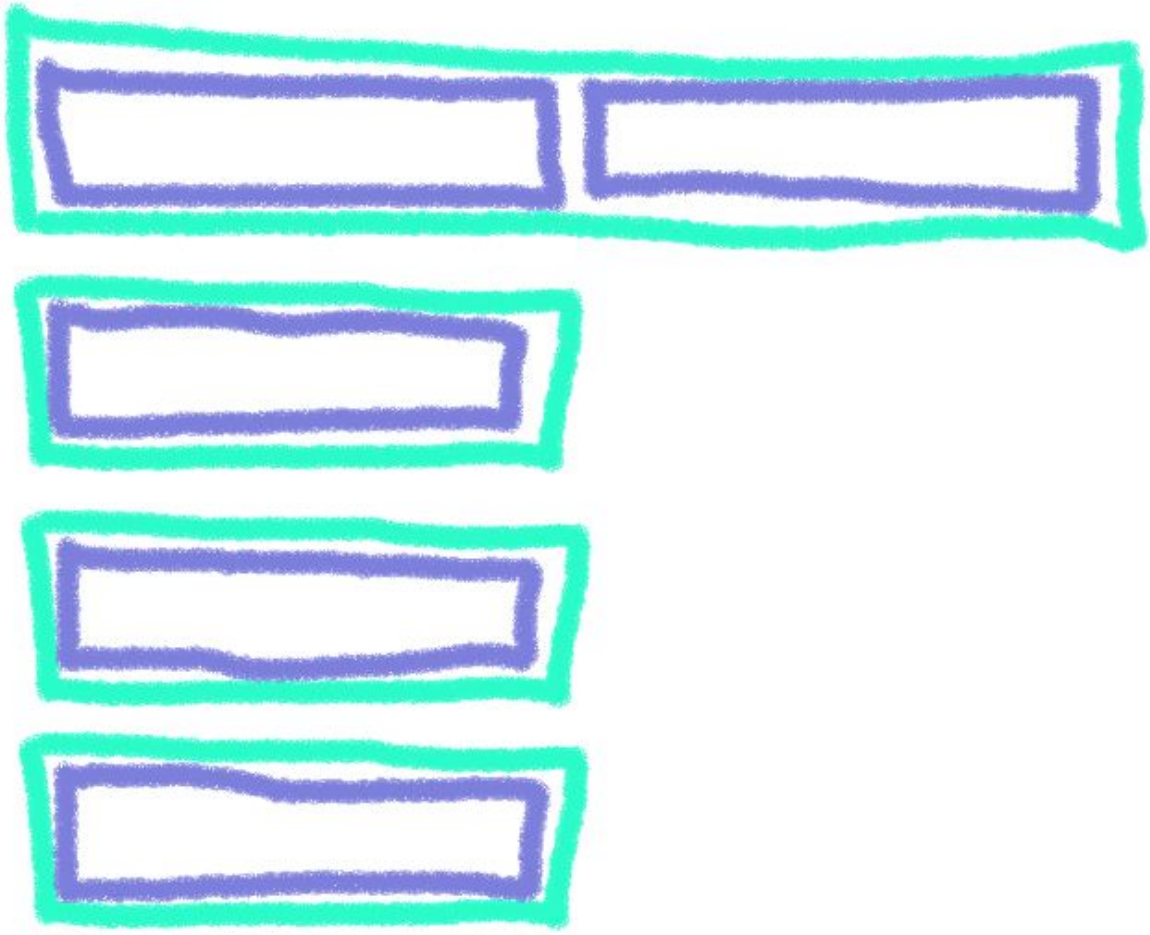


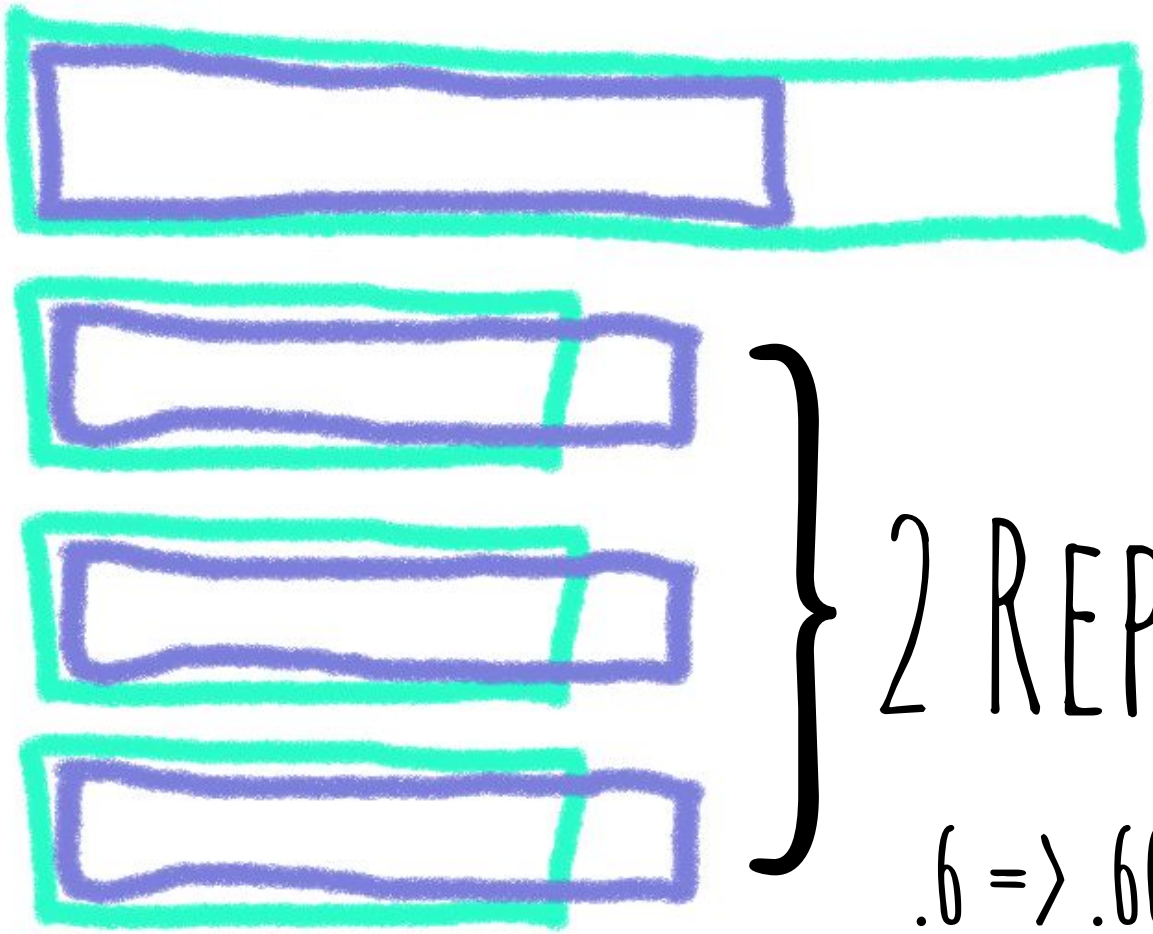


$$\frac{3}{5} = .6$$

Replicas

"units"




2 Replicas
.6 => .66~11%

Overload

Too Much => DRIVES FILL UP

Not Enough => CORRELATED DISASTER

Just use 10% (Hopefully it was cat pics?)

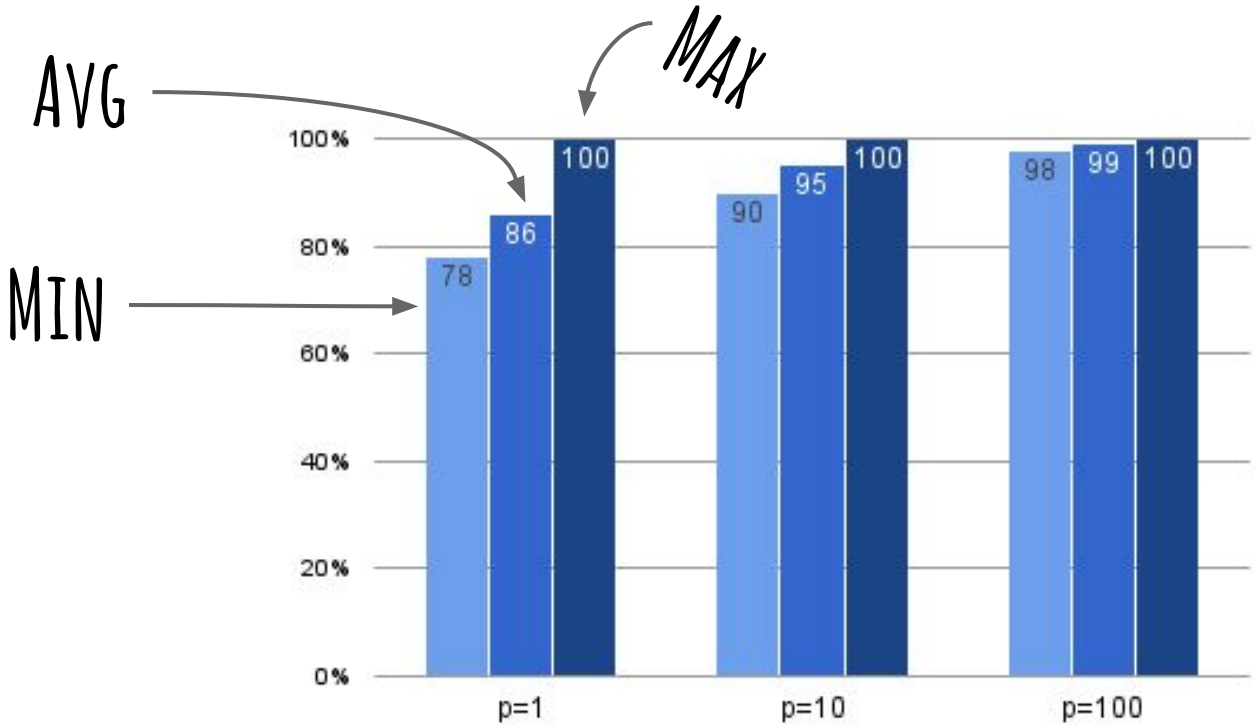
... it'll probably be fine

Partition power

Balancing the unknowns

- How to distribute objects of unknown size well-balanced?
  - Objects vary between 0 bytes and 5 GiB in size
- => Store more than one partition per disk
- => Aggregation of random sizes balances out

Disk fill level vs. partition count



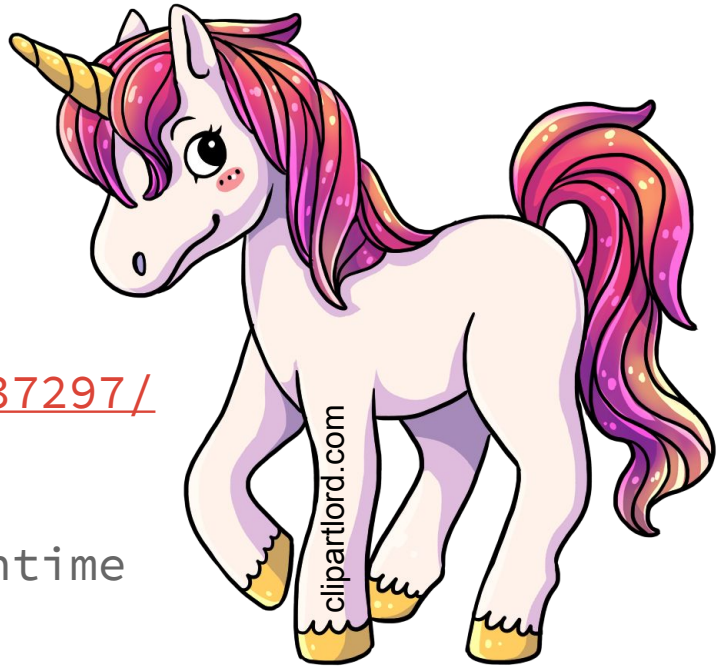
Choosing partition power

- Number of partition is fixed
- More disks => less partitions per disk
- Choose a part power with a ~ thousand partitions per disk
  - Based on today's need, not an imaginary future growth
- It is highly unlikely that your partition power is >> 20, and definitely not 32

https://gist.github.com/clayg/6879840

You became an unicorn

- Skyrocketing growth? Congrats!
- We're working on increasing partition power for you to keep your cluster balanced
  https://review.openstack.org/#/c/337297/
- Decreasing won't be possible - at least not without a serious downtime



Wrapping Up

What's a good cluster?

PARTPOWER 14 -> 2^14 = 16384

16384 PARTITIONS * 3 REPLICAS / 32 DISKS =

1536 PARTS PER DISK

MAIN DATACENTER

2ND DATACENTER

DISK WEIGHT

(64+64+60) / 3 = 62.66

ONE RACK + SWITCH

64 TB

60 TB

OVERLOAD: 4.5%

DISPERSION: 0

BALANCE: 4.65

Questions?

Thanks!

clay@swiftstack.com

cschwede@redhat.com