Elliptics Network

Evgeniy Polyakov

<zbr@ioremap.net>
<zbr@yandex-team.ru>

Distributed hash table
Key/value storage
How to handle huge dataset?

Can existing solutions scale?
Distributed hash table

Consistent hashing
Map and routing table
Elliptics network architecture

Frontend
Core
Backend
Frontends

HTTP

Bindings

POHME LF S

Command Line

500 - Internal

mount -t pohmelfs /mnt

root@main.google.com:~ # df -h
system message: stats are bad

root@main.google.com:~ # reboot
system message: going to meet Kenny, bastards
Frontends: HTTP

500 - Internal
Frontends: bindings
Frontends: command line

```
root@main.server# dnet_stat
system message: stats are really bad
root@main.server# reboot
system message: going to meet Kenny, bastards!
```
Frontends: POHMEFLFS

```
morning~ root@server# mount -t pohmel 213.180.204.3 /mnt
```
IO backends

libeblob

Files


distribute


Databases
Eblob random read performance: SAS

- 2 sas shelves (14 disks raid10 each, ext4)
- 1 Tb of data
- ~ 100 millions of objects
- Eblob: 5000 rps
- Eblob: 3500 rps within 100 ms
- Eblob: 4000 rps within 200 ms
- Filesystem: 600 rps within 200 ms
- Filesystem: 800 rps within 300 ms

FS contains about 30 millions of objects actually
Eblob random read performance: SATA

- 2 sata raids (4-disks raid10 each, ext4)
- 370 Gb of data
- 30 millions of objects
- Eblob: 1000 rps
- Eblob: 900 rps within 100-150 ms
- Filesystem: 200 rps within 200 ms
Elliptics network: core

Transactions, versions

Data replication

Fault Tolerance

Data deduplication
IO models

Write always succeed
Multiple copy reading
Eventual consistency
Future plans

* Fast Recovery
* POHMELEFS
* Distributed locks transactions

World Domination
Kill all Humans
wtf

Questions ?