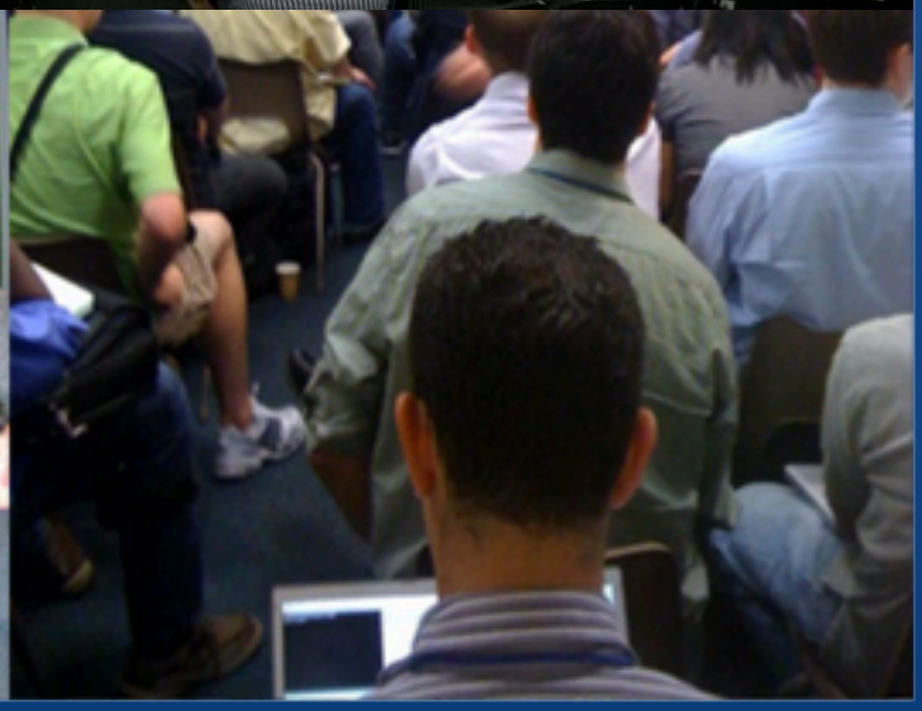
10gen mongoDB

Administration

Brendan McAdams
brendan@10gen.com



Some Duplication with Kyle's talk on Deployment (After this)

Hardware

Memory

- MongoDB revolves around memory mapped files

Operating System map files on the Filesystem to Virtual Memory

- (200 gigs of MongoDB files creates 200 gigs of virtual memory)
- OS controls what data in RAM
- When a piece of data isn't found, a page fault occurs (Expensive + Locking!)
- OS goes to disk to fetch the data
- Indexes are part of the Regular Database files
- Deployment Trick: Pre-Warm your Database (PreWarming your cache) to prevent cold start slowdown

MongoDB will take advantage of multiple cores

- For working set queries, CPU usage is typically low

10gen mongoDB

Full Tablescans

- Surprise: Queries which don't hit indexes make heavy use of CPU & Disk
- Deployment Trick: Avoid counting & computing on the fly by caching & precomputing data

10gen mongoDB

Map Reduce

- Currently Single Threaded; runs in parallel across shards
- Deployment Trick: Use the new aggregation output options

Working set is crucial!!!

- Working set should be, as much as possible, in memory
- your entire dataset need not be

10gen mongoDB

Disks & I/O

- Disk I/O becomes your definer of performance in non-working set queries

10gen mongoDB

Surprise: Faster Disks is better than slow disks. More is also better

- RAID is good for a variety of reasons
- Our Recommendations ...

10gen mongoDB

RAID 10 (Mirrored sets inside a striped set; minimum 4 disks)

- Improved write performance
- Survives single disk failure
- Downside: Needs double storage needs
- e.g. 4 20 gig disks gives you 40 gigs of usable space
- LVM of RAID 10 on EBS seems to smooth out performance and reliability best for MongoDB

10gen mongoDB

RAID 5 or 6

- 1 or 2 additional disks required for parity
- Can survive 1 or 2 disk failures
- Implementations seem inconsistent, buyer beware

Flash (SSD)

- Expensive, but getting cheaper
- Significantly reduced seek time and increased I/O Throughput
- Random Writes and Sequential Reads are still a weak point

10gen mongoDB

OS

- For production: Use a 64 bit OS and a 64 bit MongoDB Build
- 32 Bit has a 2 gig limit; imposed by the operating systems for memory mapped files
- Clients can be 32 bit
- MongoDB Supports (little endian only)
- Linux, FreeBSD, OS X (on Intel, not PowerPC)
- Windows
- Solaris (Intel only, Joyent offers a cloud service which works for Mongo)

10gen mongoDB

Put your journal on a separate spindle if possible

10gen mongoDB

Server Status

Some tools for examining server status

10gen mongoDB

MongoStat - free tool which comes with MongoDB

- Shows I/O counters, time spent in locks, etc

```
insert/s query/s update/s delete/s getmore/s command/s flushes/s mapped  vsize    res faults/s locked %
    0       0        0        0         0         1         0   56694   57360    688        0        0
    0       9        0        0         0         4         0   56694   57360    688        0      0.3
    0       8        0        0         0         1         0   56694   57360    688        0        0
    0      14        0        0         0         4         1   56694   57360    688        0      0.4
    0      14        1        0         0         4         0   56694   57360    688        0        0
    0       0        0        0         0         7         0   56694   57360    688        0        0
    0       0        0        0         0         1         0   56694   57360    688        0        0
    0       0        0        0         0         7         0   56694   57360    688        0        0
    0       0        0        0         0         4         0   56694   57360    688        0        0
    0       0        0        0         0         7         0   56694   57360    688        0        0
    0       0        0        0         0         2         0   56694   57360    688        0        0
    0       0        0        0         0        33         0   56694   57360    688        0        0
    0       0        0        0         0         2         0   56694   57360    688        0        0
```

```
faults/s locked % idx miss %     q t|r|w  conn       time
       0        0          0     0|0|0    48   11:37:13
       0      0.3          0     0|0|0    48   11:37:14
       0        0          0     0|0|0    48   11:37:15
       0      0.4          0     0|0|0    48   11:37:16
       0        0          0     0|0|0    48   11:37:17
       0        0          0     0|0|0    48   11:37:18
       0        0          0     0|0|0    48   11:37:19
       0        0          0     0|0|0    48   11:37:20
       0        0          0     0|0|0    48   11:37:21
       0        0          0     0|0|0    48   11:37:22
       0        0          0     0|0|0    48   11:37:23
       0        0          0     0|0|0    48   11:37:24
       0        0          0     0|0|0    48   11:37:25
```

10gen mongoDB

Similarly, iostat ships on most Linux machines (or can be installed)

- iostat [args] <seconds per poll>
- -x for extended report
- Disk can be a bottleneck in large datasets where working set > ram
- ~200-300Mb/s on XL EC2 instances, but YMMV (EBS is slower)
- On Amazon Latency spikes are common, 400-600ms (No, this is not a good thing)

IOStat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.32    0.00    0.32    0.35    0.15   97.90

Device:            rrqm/s   wrqm/s     r/s     w/s    rsec/s    wsec/s avgrq-sz avgqu-sz   await  svctm  %util
sda1               1.22     3.84       0.70    2.87   34.46     53.67    24.68     0.05    15.04   4.80   1.71
sda2               0.01     0.01       0.00    0.00    0.08      0.09    31.59     0.00    27.94  12.18   0.01
```

10gen mongoDB

db.serverStatus()

```
> db.serverStatus()
{
        "host" : "b-mac",
        "version" : "1.8.0",
        "process" : "mongod",
        "uptime" : 845952,
        "uptimeEstimate" : 355978,
        "localTime" : ISODate("2011-04-26T17:04:40.938Z"),
        "globalLock" : {
                "totalTime" : 845952742184,
                "lockTime" : 5154824,
                "ratio" : 0.000006093512962310125 4,
                "currentQueue" : {
                        "total" : 0,
                        "readers" : 0,
                        "writers" : 0
                },
                "activeClients" : {
                        "total" : 0,
                        "readers" : 0,
                        "writers" : 0
                }
        },
        "mem" : {
```

10gen mongoDB

db.serverStatus()


```
                "writers" : 0
        }
},
"mem" : {
        "bits" : 64,
        "resident" : 7,
        "virtual" : 28917,
        "supported" : true,
        "mapped" : 13245
},
"connections" : {
        "current" : 1,
        "available" : 203
},
"extra_info" : {
        "note" : "fields vary by platform"
},
"indexCounters" : {
        "btree" : {
                "accesses" : 37,
                "hits" : 37,
                "misses" : 0,
                "resets" : 0,
                "missRatio" : 0
},
"backgroundFlushing" : {
```

db.serverStatus()


```
                "misses" : 0,
                "resets" : 0,
                "missRatio" : 0
        }
},
"backgroundFlushing" : {
        "flushes" : 5999,
        "total_ms" : 83634,
        "average_ms" : 13.941323553925654,
        "last_ms" : 2,
        "last_finished" : ISODate("2011-04-26T17:04:02.582Z")
},
"cursors" : {
        "totalOpen" : 0,
        "clientCursors_size" : 0,
        "timedOut" : 21
},
"network" : {
        "bytesIn" : 1614262,
        "bytesOut" : 19368250,
        "numRequests" : 6395
},
"opcounters" : {
        "insert" : 2359,
        "query" : 435,
        "update" : 94,
```

db.serverStatus()


```
                "getmore" : 2776,
                "command" : 725
        },
        "asserts" : {
                "regular" : 0,
                "warning" : 0,
                "msg" : 0,
                "user" : 8,
                "rollovers" : 0
        },
        "writeBacksQueued" : false,
        "dur" : {
                "commits" : 0,
                "journaledMB" : 0,
                "writeToDataFilesMB" : 0,
                "commitsInWriteLock" : 0,
                "earlyCommits" : 0,
                "timeMs" : {
                        "dt" : 3008,
                        "prepLogBuffer" : 0,
                        "writeToJournal" : 0,
                        "writeToDataFiles" : 0,
                        "remapPrivateView" : 0
                }
        },
        "ok" : 1
}
```


10gen mongoDB

Filesystems

All data & namespace files are stored in the 'data' directory (--dbpath)

- You can create symbolic links to keep different databases on different disks
- Best to aggregate your IO across multiple disks
- File Allocation

```
$ ls -sk /data/db/
 16384 foo.ns
 65536 foo.0
131072 foo.1
 16384 bar.ns
   ...
```

double in size (up to 2 gigs)

allocated per database

10gen mongoDB

Logfiles

- --logpath <file>
- Rotation can be requested of MongoDB...
- db.runCommand("logRotate")
- kill -SIGUSR1 <mongod pid>
- killall -SIGUSR1 mongod
- Won't work for ./mongod > [file] syntax

10gen mongoDB

Filesystems

- MongoDB is filesystem neutral
- ext3, ext4 and XFS are most used
- BUT....
- ext4, XFS or any other filesystem with posix_fallocate() are preferred and best

EC2

- Many distros default to ext3 (but Amazon AMI now uses ext4 by default)
- For best performance reformat to EXT4 / XFS
- Make sure you use a recent version of EXT4
- Striping (MDADM / LVM) aggregates I/O
- See previous recommendations about RAID 10

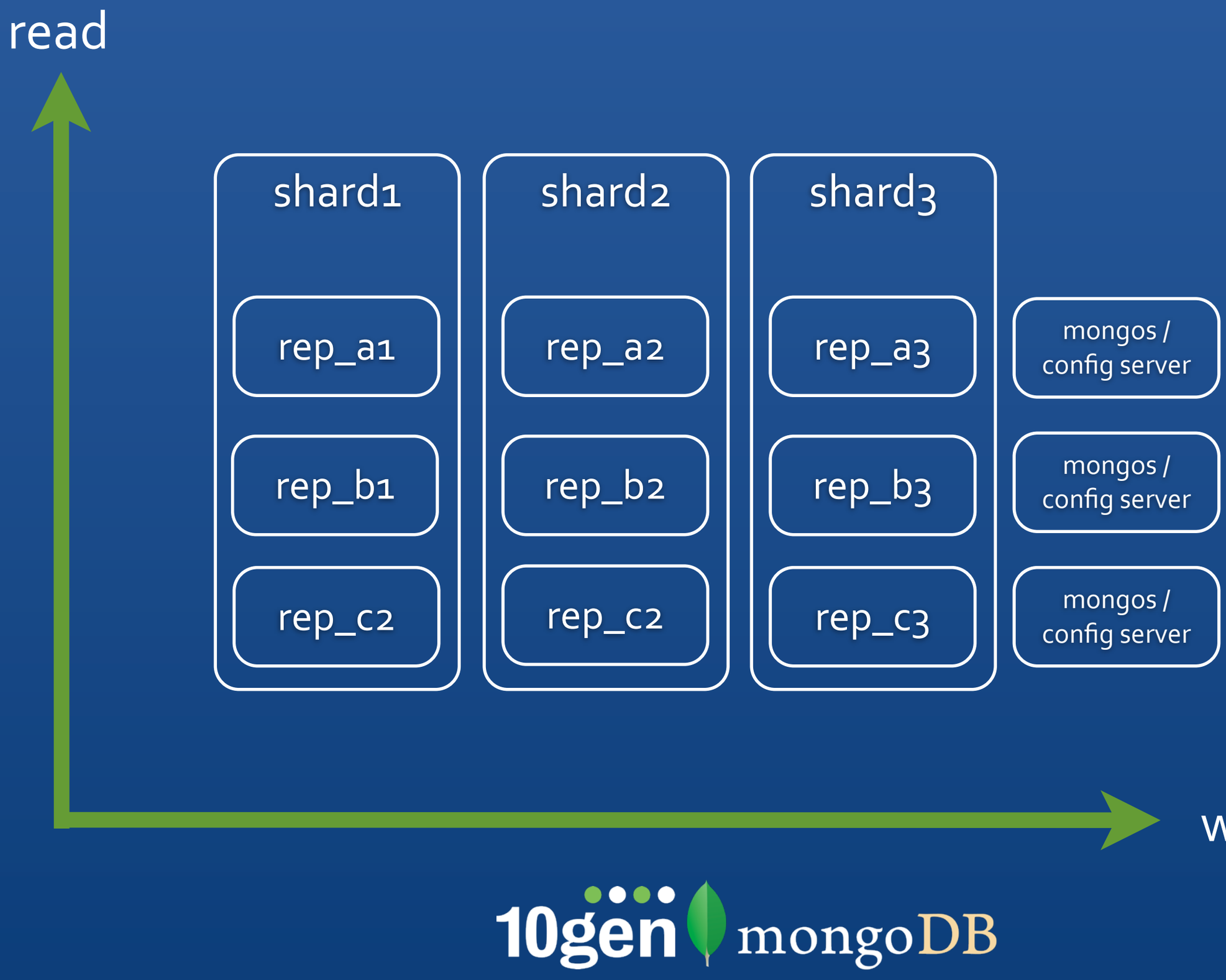
Monitoring

Maintenance

- When doing a lot of updates or deletes....
- Compaction may be needed occasionally on indices and datafiles
- db.repair()
- Replica Sets:
- Rolling repairs, start nodes up with --repair param

10gen mongoDB

Scale out

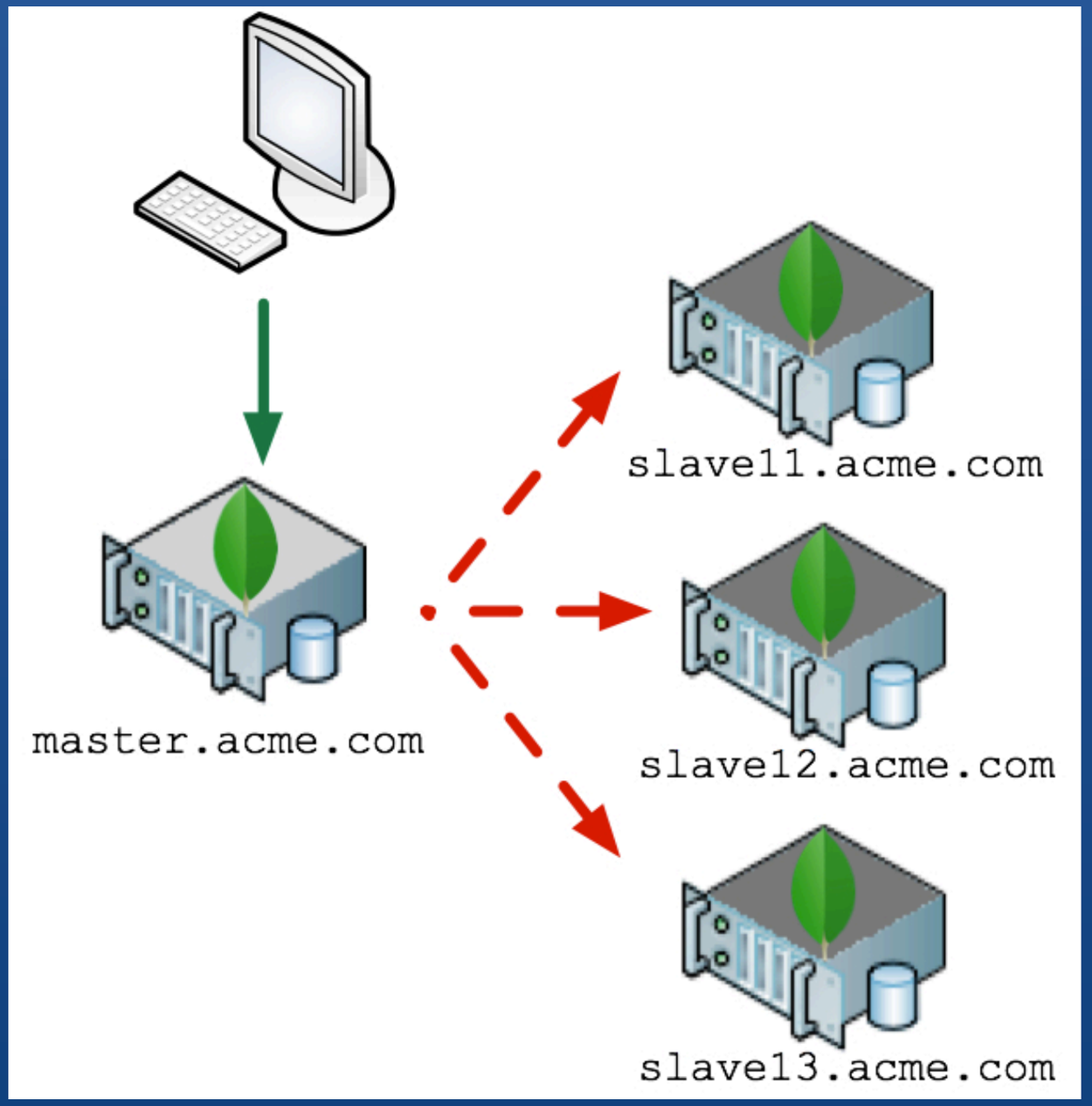


10gen mongoDB

Backups

Best driven from a slave

- Eliminates impact on master during backup
- Hidden Nodes in 1.8



10gen mongoDB

mongodump / mongorestore

- binary, compact object dump
- each consistent object is written
- NOT necessarily consistent from start to finish (Unless you lock the database)
- mongorestore to restore binary dump
- database doesn't have to be up to restore, can use dbpath

10gen mongoDB

filelock / fsync

- lock: blocks writes
- db.runCommand({fsync: 1, lock: 1})
- fsync to flush buffers to disk
- backup
- then, unlock
- db.$cmd.sys.unlock.findOne();

10gen mongoDB

With Journaling, you can run an LVM or EBS snapshot and recover later without locking

- EBS Can disappear (See: last week)
- S3 for longer term backups
- USE AMAZON AVAILABILITY ZONES
- DR / HA

10gen mongoDB

download at mongodb.org

We're Hiring !

brendan@10gen.com

(twitter: @rit)

conferences, appearances, and meetups

http://www.10gen.com/events

Facebook

http://bit.ly/mongofb

Twitter

@mongodb

LinkedIn

http://linkd.in/joinmongo

10gen mongoDB