mongoDB

Kristina Chodorow

Please ask questions!

Have people used non-relational dbs before?
MongoDB?

$ whoami

Software Engineer at 10gen

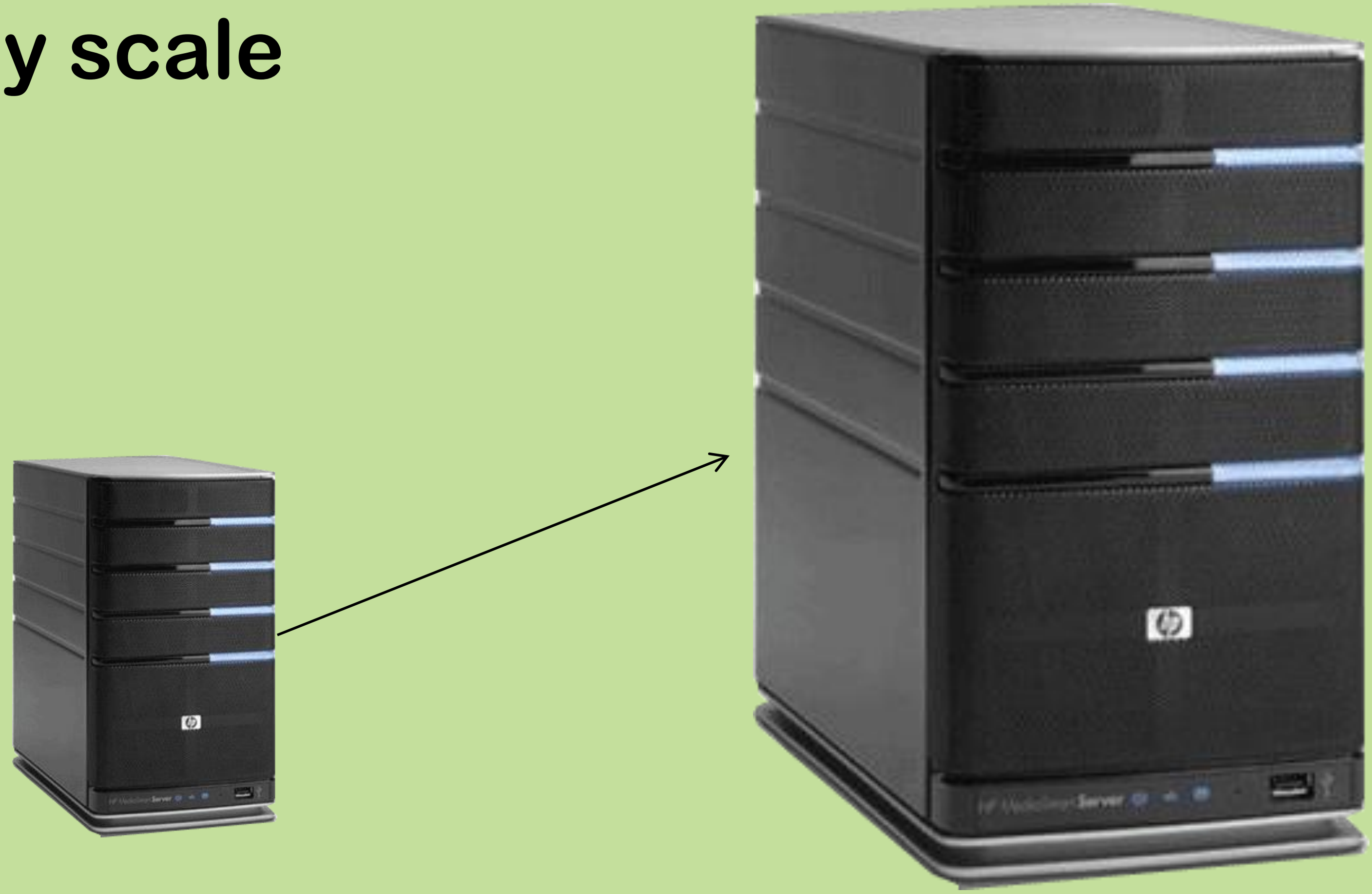
Java

PROGRAMMING REPUBLIC OF PERL

php

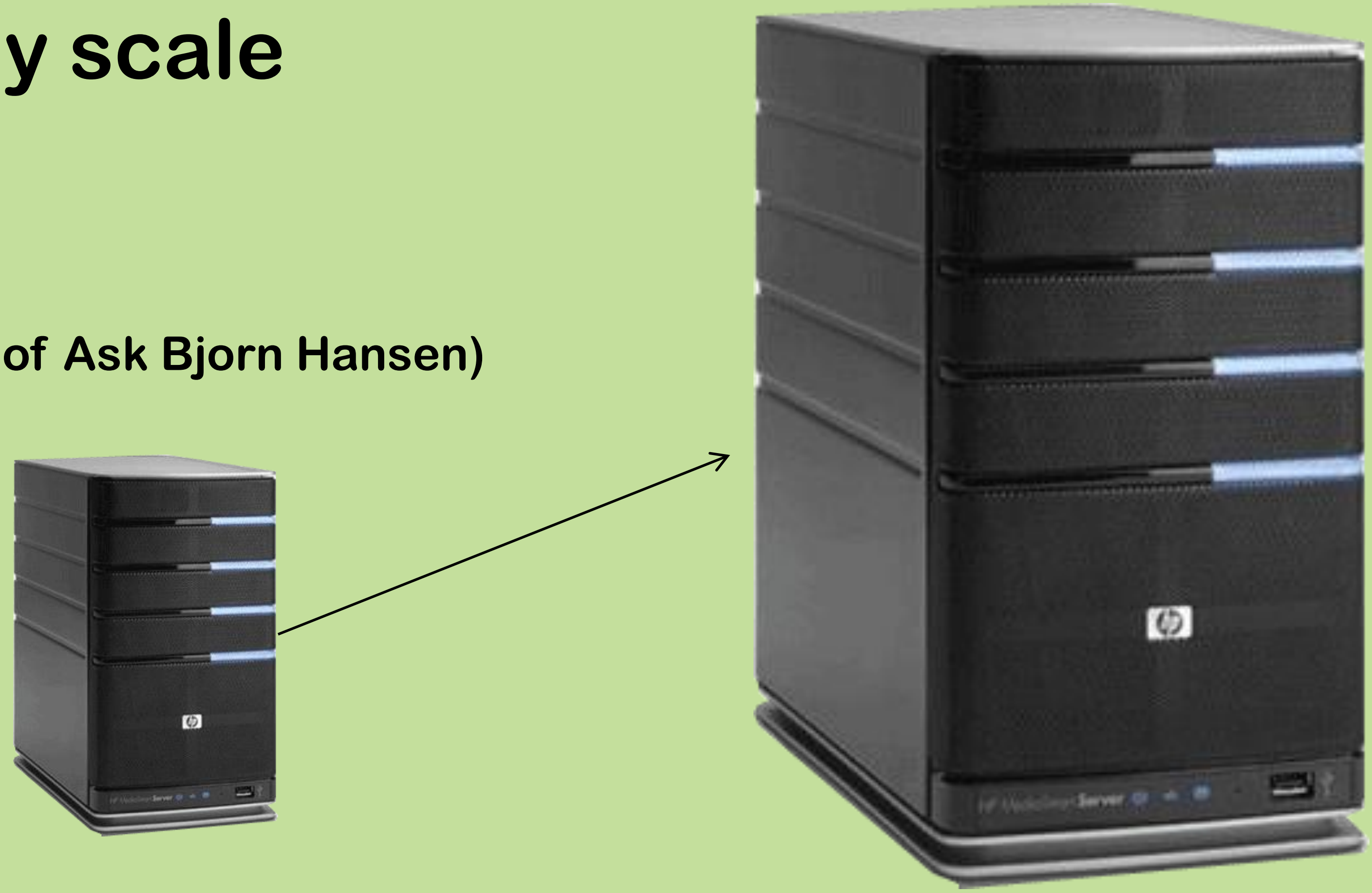
Scaling a Pre-WWW DB

literally scale



literally scale

(Courtesy of Ask Bjorn Hansen)



$3,200 x 1250 = $4,000,000

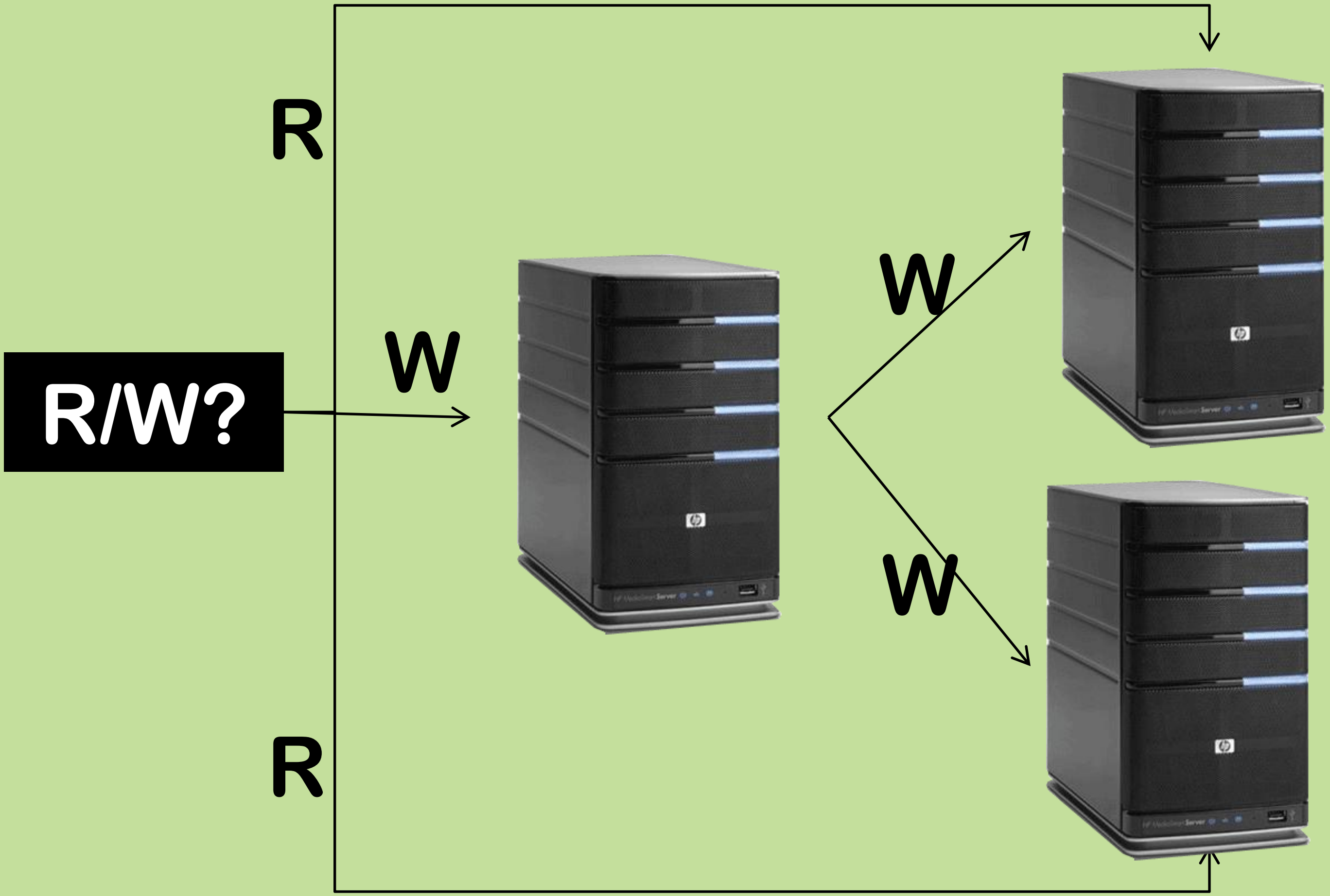
shard

now-yesterday

yesterday-
the day before

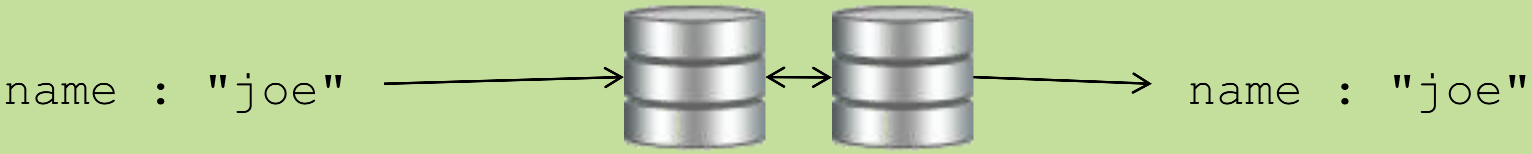
the day before-
the day before that

master-slave replication



CAP Theorem: Choose Two

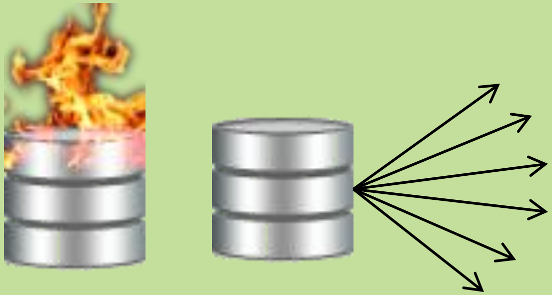
- Consistency



`name : "joe"`

`name : "joe"`

- Availability

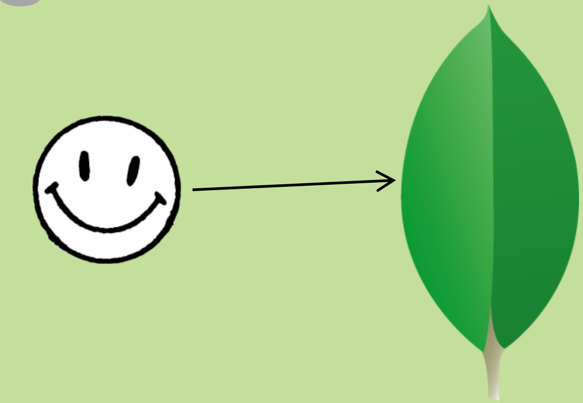


- Partitioning

Consistency

Availability

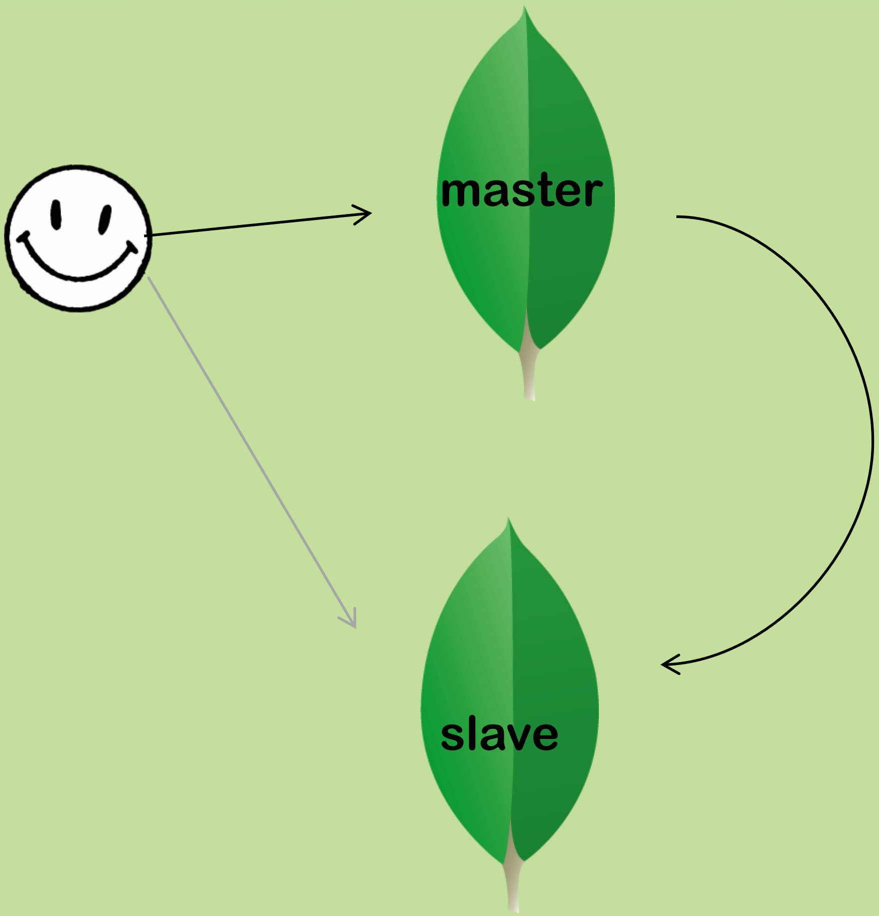
Partitioning



Consistency

Availability

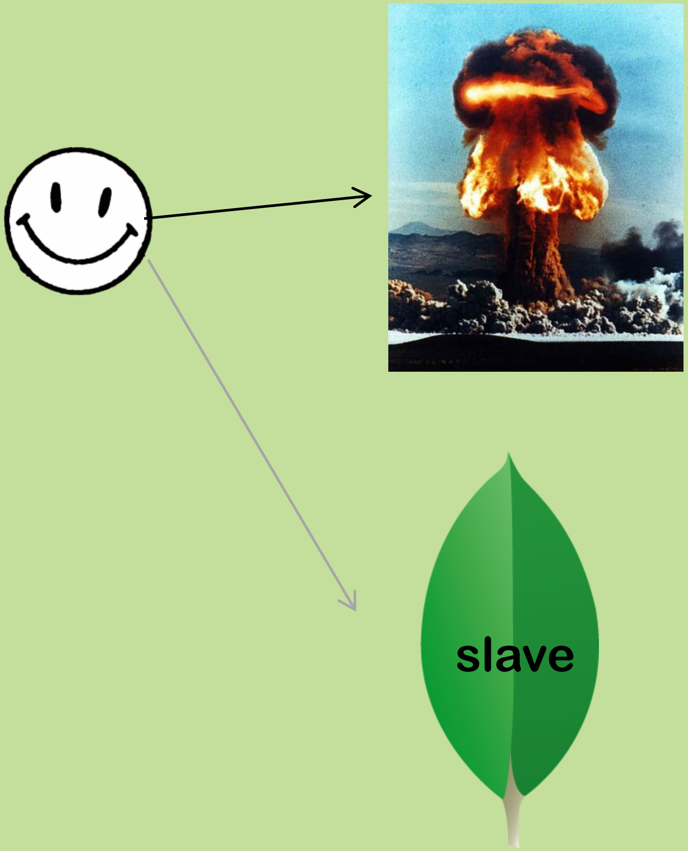
Partitioning



Consistency

Availability

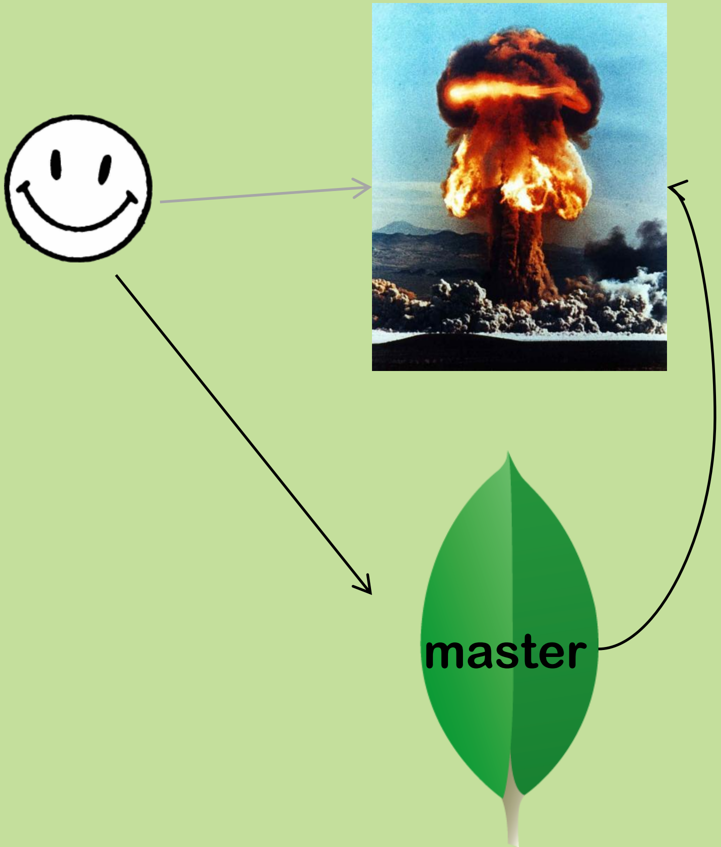
Partitioning



Consistency

Availability

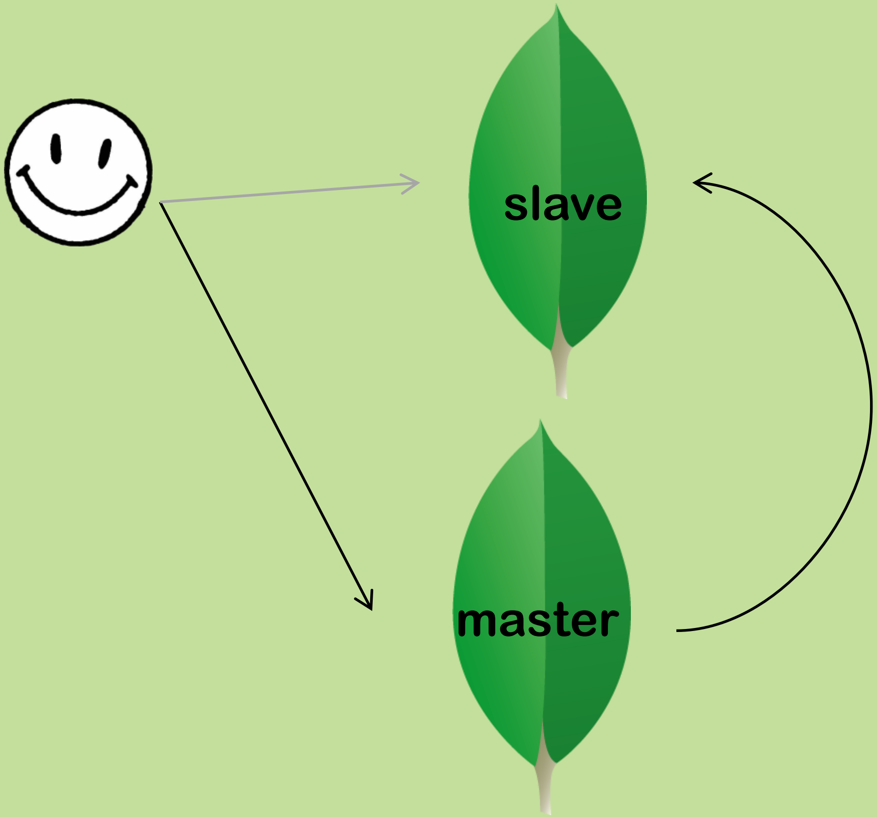
Partitioning



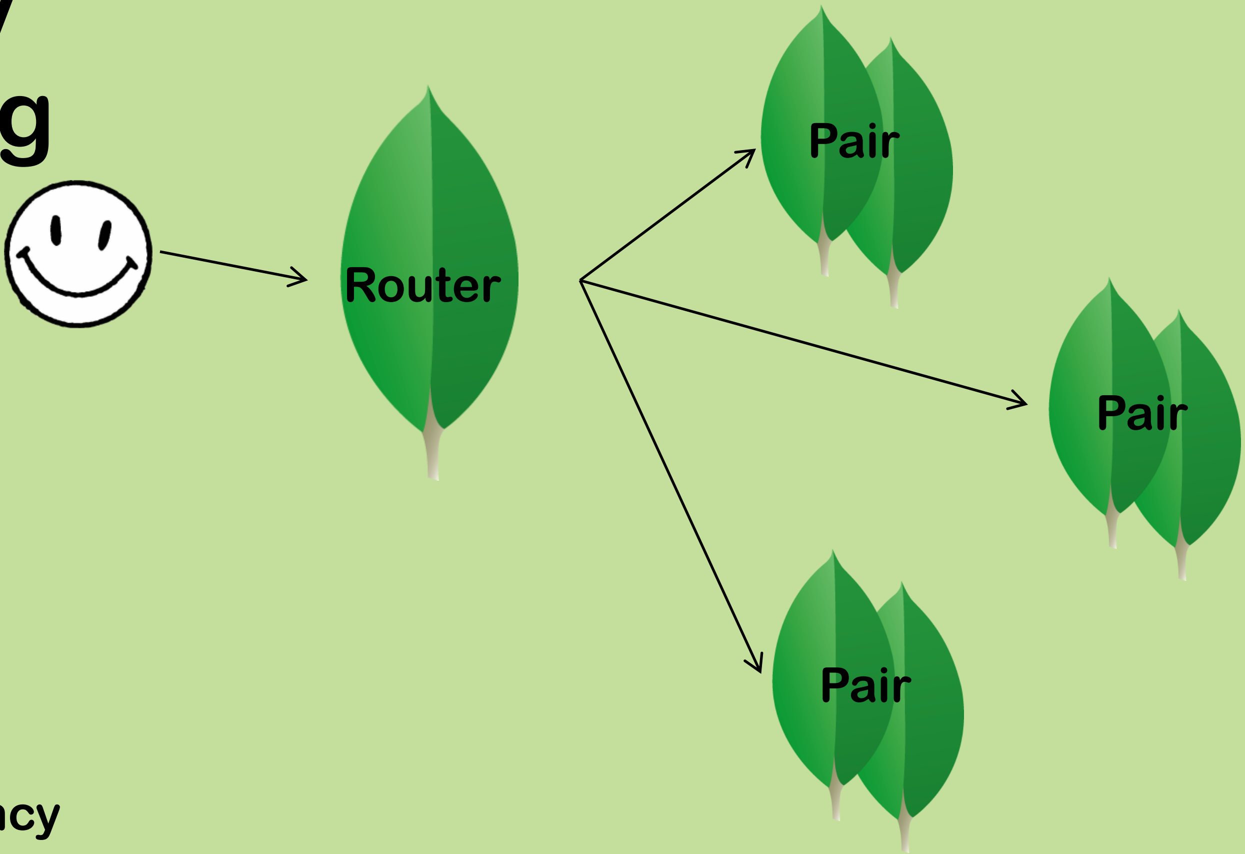
Consistency

Availability

Partitioning

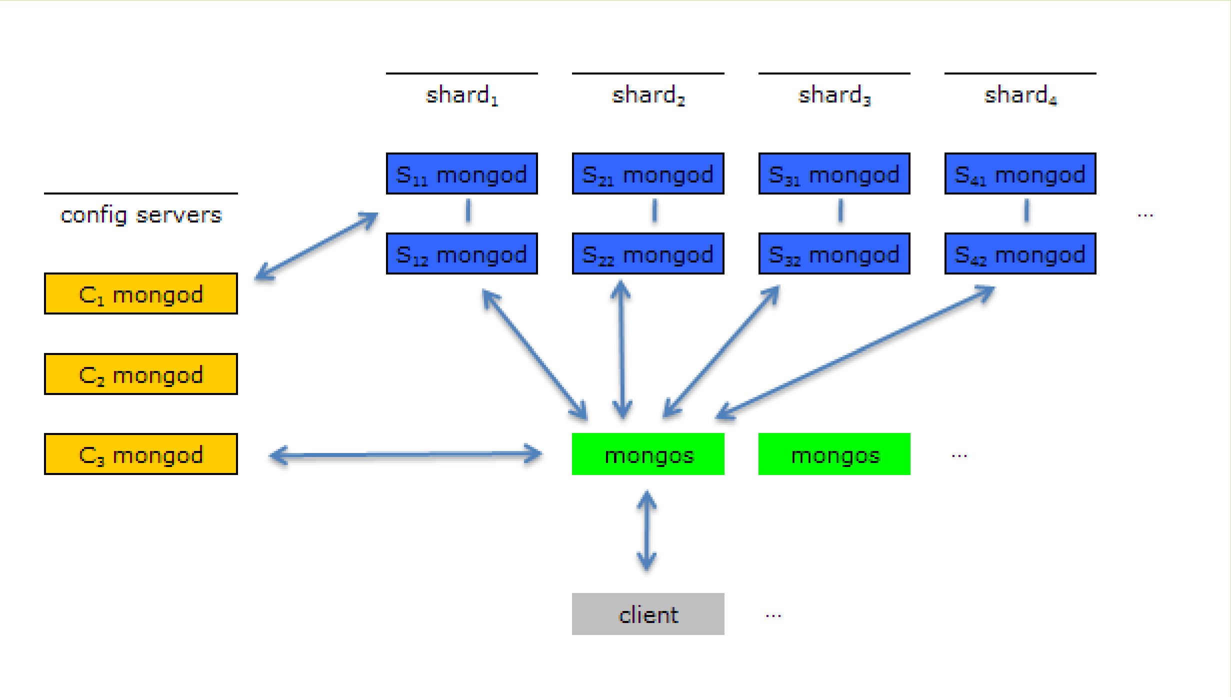


Consistency
Availability
Partitioning



...eventual consistency

mongoDB


shard$_1$
shard$_2$
shard$_3$
shard$_4$
config servers
S$_{11}$ mongod
S$_{21}$ mongod
S$_{31}$ mongod
S$_{41}$ mongod
...
S$_{12}$ mongod
S$_{22}$ mongod
S$_{32}$ mongod
S$_{42}$ mongod
C$_1$ mongod
C$_2$ mongod
C$_3$ mongod
mongos
mongos
...
client
...

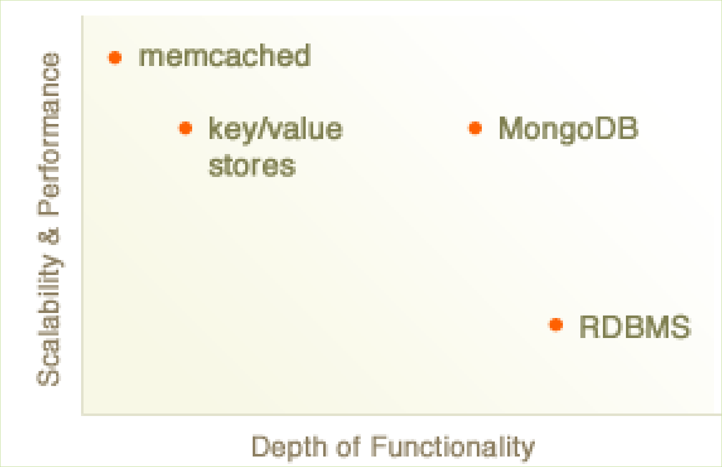
Mongo Philosophy

- Go horizontal
- You can always lose a machine
- Safety in numbers

Using MongoDB & PHP

More Mongo Philosophy

- Everyone knows some JavaScript
- Performance!



www.mongodb.org

Comes with binaries for MongoDB server, shell, and import/export tools

The PHP Driver

```
$ pecl install mongo
```

Add to php.ini:

```
extension=mongo.so
```

Other Drivers

C#, Erlang, Factor

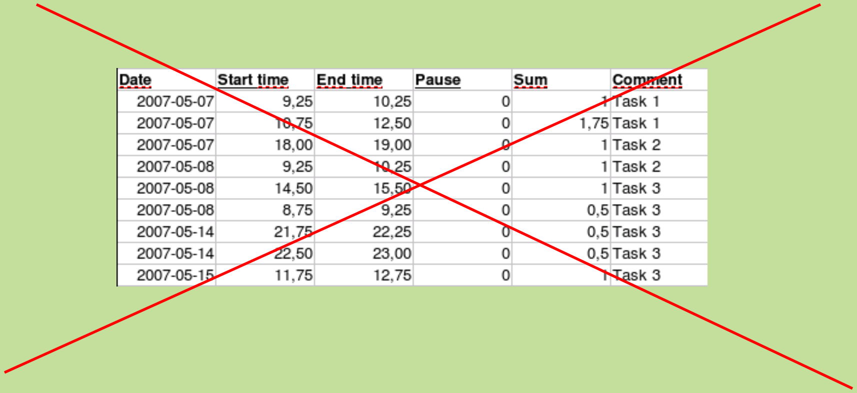
Connecting to the Database

```
$connection = new Mongo();

$db = $connection->selectDB('test');

$collection = $db->selectCollection('foo');
```

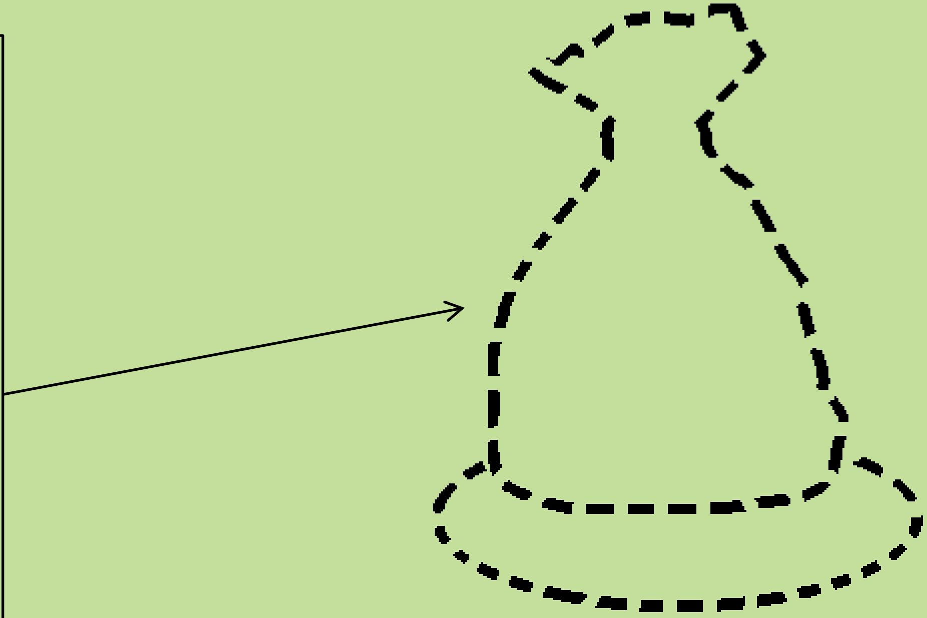
Collections, not Tables



Date	Start time	End time	Pause	Sum	Comment
2007-05-07	9,25	10,25	0	1	Task 1
2007-05-07	10,75	12,50	0	1,75	Task 1
2007-05-07	18,00	19,00	0	1	Task 2
2007-05-08	9,25	10,25	0	1	Task 2
2007-05-08	14,50	15,50	0	1	Task 3
2007-05-08	8,75	9,25	0	0,5	Task 3
2007-05-14	21,75	22,25	0	0,5	Task 3
2007-05-14	22,50	23,00	0	0,5	Task 3
2007-05-15	11,75	12,75	0	1	Task 3

Collections

```
array(
    'date' => new MongoDate,
    'time' => array(
        'start' => 9.25,
        'end' => 10.25
    ),
    'sum' => 0,
    'comments' => array(
         "task 1",
         "run 4"
    )
)
```



Inserting

```
$c->insert(array('name' => "Joe", 'age' : 34))

var_dump($c->findOne(array('name' => "Joe")));
array(3) {
    ["_id"]=>
    object(MongoId)#6 (0) {
    }
    ["name"]=>
    string(3) "Joe",
    ["age"]=>
    int(34)
}
```

Inserting Objects

```
class Post {
   public $comments = array();
   public function __construct($author, $title, $content) {
       $this->author = $author;
       $this->title = $title;
       $this->content = $content;
   }
}

$c->insert(new Post("Fred", "My first blog post", "Hello,
   World!"));
```

MongoId

an autogenerated primary key

```
$x = array("foo" => 1, "bar" => 2);
$collection->insert($x);
echo $x['_id'];
```

Result:

```
2fe3e4d892aa73234c910bed
```

Autogenerated Primary Keys

```
MongoId: "2fe3e4d892aa73234c910bed"
```

12 bytes:

```
2fe3e4d892aa73234c910bed
|------||----||--||----|
   ts      mac  pid  inc
```

Updating

```
$collection->update(array("_id" => $id),
array('$push' => array('comments' =>
  array(
     'author' => 'Fred',
     'comment' => 'Dumb post.'
  )
)));
```

...which gives us:

```
echo json_encode($collection->findOne());
--------------------------------------------
 {
   _id : {},
   "title" : "My first blog post",
   "author" : "Joe",
   "content" : "Hello, world!"
   "comments" : [{
      "author" : "Fred",
      "comment" : "Dumb post"
   }]
 }
```

$ instead of >, <, =, etc.

$gt, $gte, $lt, $lte, $eq, $neq, $exists,
$set, $mod, $where, $in, $nin, $inc
$push, $pull, $pop, $pushAll, $popAll

```
$c->find(array("x" => array('$gt' => 4)))
```

$where

```
$blog->findOne(array('$where' =>
    'this.y == (this.x + this.z)'));
```

Will work:

```
array("x" => 1, "y" => 4, "z" => 3)
array("x" => "hi", "y" => "hibye",
    "z" => "bye")
```

Won’t work:

```
array("x" => 1, "y" => 1)
```

Optimizing $where

```
db.blogs.findOne({
    name : "Sally",
    age : {'$gt' : 18},
    $where : 'Array.sort(this.interests)[2]
  == "volleyball"'});
```

php.ini

```
mongo.cmd = ":"
$c->find(array("x" => array(':gt' => 4)));

mongo.cmd = "="
$c->find(array("x" => array('=gt' => 4)));

mongo.cmd = "#"
$c->find(array("x" => array('#gt' => 4)));
```

ini_set

```
ini_set("mongo.cmd", ".");
$c->find(array("x" => array('.gt' => 4)));

ini_set("mongo.cmd", "x");
$c->find(array("x" => array('xgt' => 4)));

ini_set("mongo.cmd", "\n");
$c->find(array("x" => array("\ngt" => 4)));
```

Querying

```
$posts = $blog->find(array(
     "author" => "Joe"));

$commentsByFred = $blog->find(array(
     "comments.author" => "Fred"));

$commentsByFred = $blog->find(array(
     "comments.author" =>
     new MongoRegex("/fred/i")));
```

Speaking of indexing...

```
$people->ensureIndex(array("age" => 1));

$people->ensureIndex(array(
    "name" => -1,
    "ts" => -1,
    "comments.author" => 1
));
```

Cursors

```
$cursor = $c->find(array("foo" => "bar"))

foreach($cursor as $id => $value) {
  ...
}

$a = iterator_to_array($cursor);
```

Paging

```
$page_num = 3;
$results_per_page = 10;

$cursor = $results->find()
    ->sort(array("ts" => -1))
    ->skip($page_num * $results_per_page)
    ->limit($results_per_page);
```

BSON

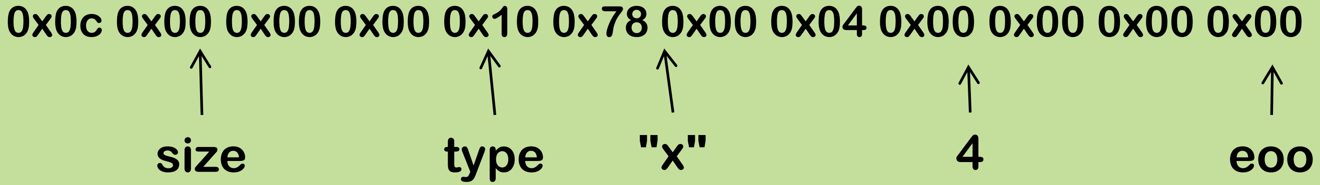
Binary Serializable Object Notation

(Binary JSON)

JS: { x : 4 }

PHP: array("x" => 4)

BSON: 0x0c 0x00 0x00 0x00 0x10 0x78 0x00 0x04 0x00 0x00 0x00 0x00



size

type

"x"

4

eoo

BSON

```
$bson = bson_encode($mixed)
$array = bson_decode($bson)
```

Applications

silentale

floxee

sourceforge

MUSIC NATION
MUSICNATION.COM

soliMAP

TweetSaver

@trackmeet

SHOPwiki

Sluggy Freelance

Tweet Congress

style signal

Boxed Ice

eflyover

FetLife

Sifino

Dextify²

DEFENSIO
A WEBSENSE SERVICE

Sailthru
You Communicate, We Deliver

DISQUS

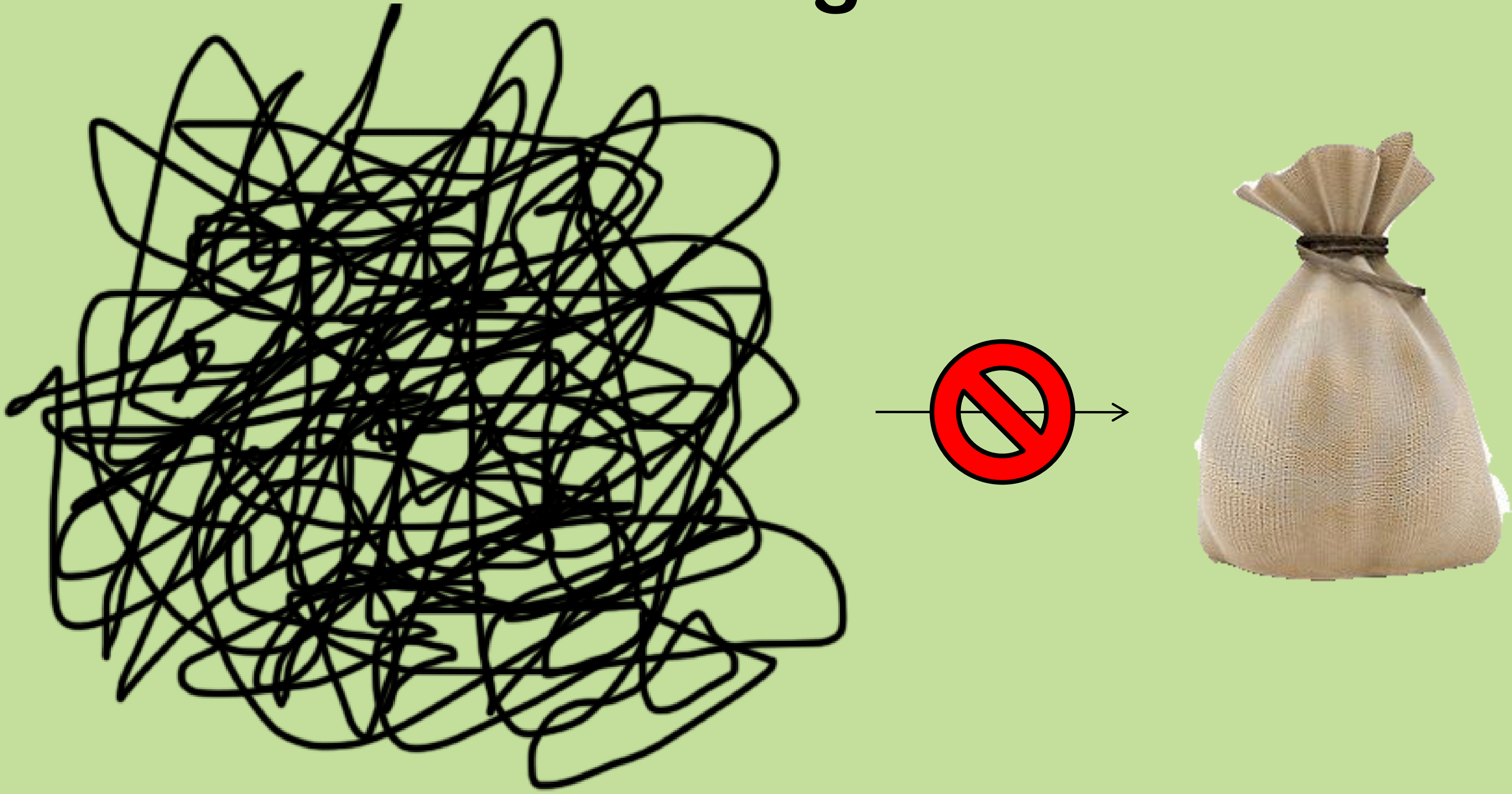
Starter App: Logging

- insert/update/remove is fast
- Capped collections
- Schemaless
- $inc for counts

Storing Files

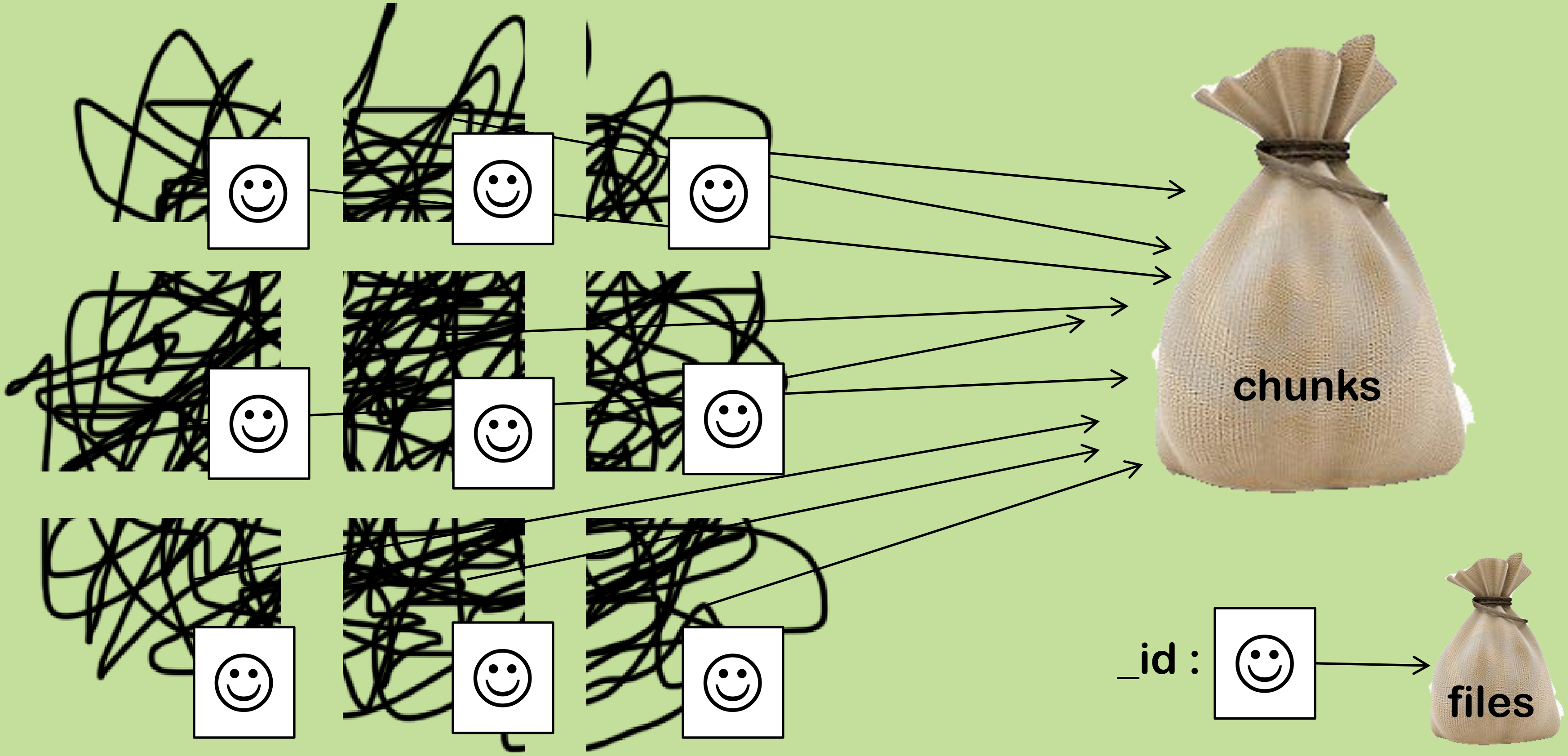
Max: 4Mb

Storing Files



(More than 4 Mb)

Storing Files



Storing Files

```
$fileId = new MongoId();

$fileObj  = array(
    '_id' => $fileId,
    'filename' => "ggbridge.png",
    'user' => "joe",
    'takenIn' => "San Francisco"
)

$chunkObj = array(
    'fileId' : $fileId,
    'chunkNum' : N
    'data' : <binary data>
)
```

Aggregation

```
count, distinct, group =~ GROUP BY
```

Map/Reduce

```
$db->command(array(
    'mapreduce' => <collection>,
    'map' => <mapfunction>,
    'reduce' => <reducefunction>
    [, 'query' => <query filter object>]
    [, 'out' => <outputcollectionname>]
    [, 'keeptemp' => <true|false>]
    [, 'finalize' => <finalizefunction>]
))
```

JavaScript Functions in PHP

```
$code = new MongoCode(
    "function() { return 'hello'; }");

$code = new MongoCode(
    "function() { return 'hello, '+name; }",
    array("name" => "Fred"));

$db->execute($code);
```

JavaScript Functions in PHP

```
$code = new MongoCode(
    "function(myname) { ".
         "return 'hello, '+name+', my name
  is '+myname;".
    "}",
    array("name" => "Fred")
);

$db->execute($code, array("Joe"));
```

One Last Bit of Mongo Philosophy

- User Support

irc.freenode.net#mongodb

http://groups.google.com/group/mongodb-user/

@mongodb

Thank you!

kristina@php.net

@kchodorow