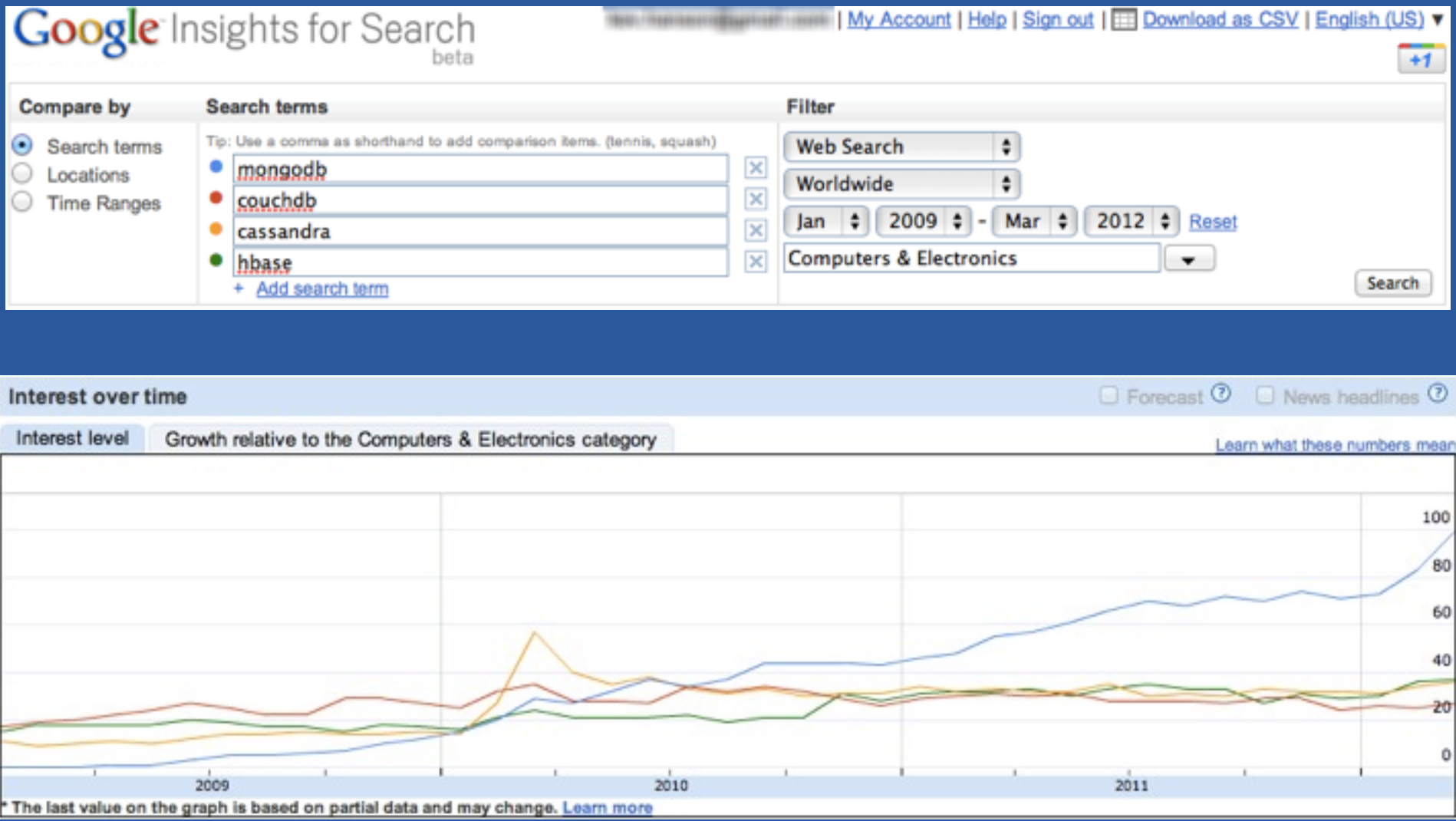
mongoDB

Technical Overview

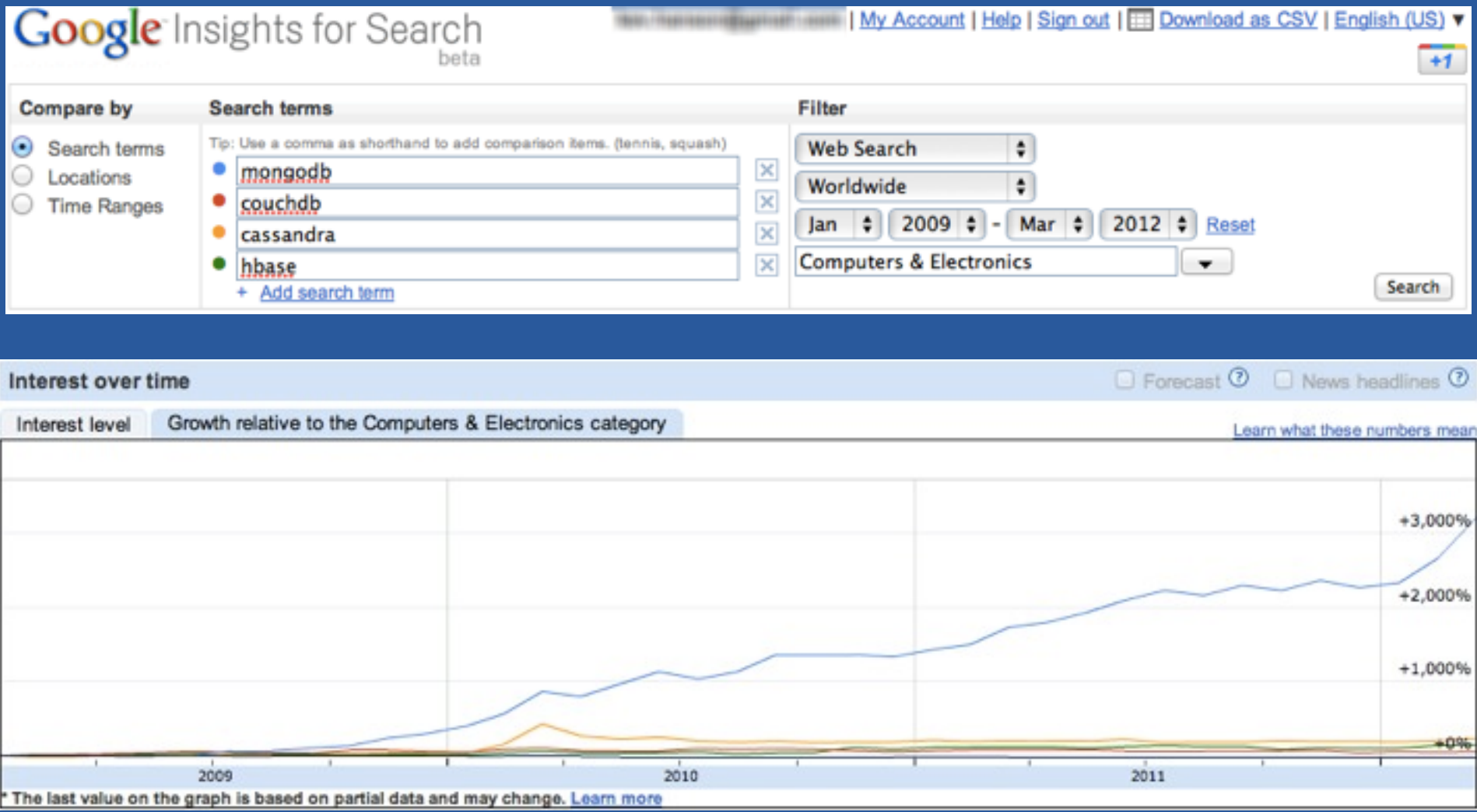
Kevin Hanson
Solutions Architect, 10gen
@hungarianhc
kevin@10gen.com

mongoDB Adoption

mongoDB Adoption



mongoDB Adoption



mongoDB Adoption

- Over 1 Million Downloads in 2011
- Currently over 30,000 Downloads / Week
- Enterprise Adoption Increasing
- En Route to Becoming the De Facto Non Relational Database

Why Build MongoDB

- Doubleclick - 400,000 ads/second
- People writing their own stores
- Caching is de rigueur
- Complex ORM frameworks
- Computer architecture trends
- Cloud computing

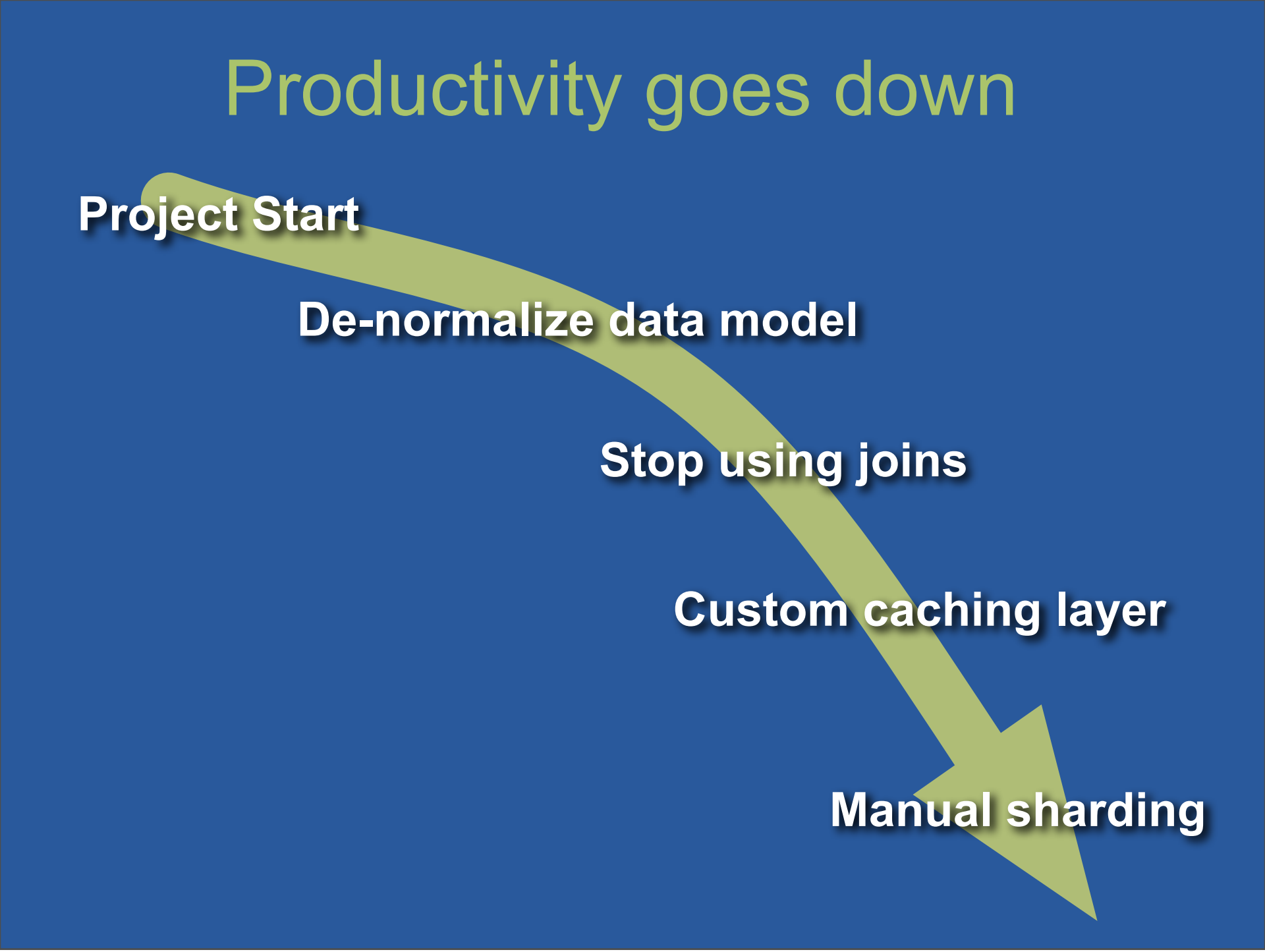
Scaling RDBMS...
Costs go up

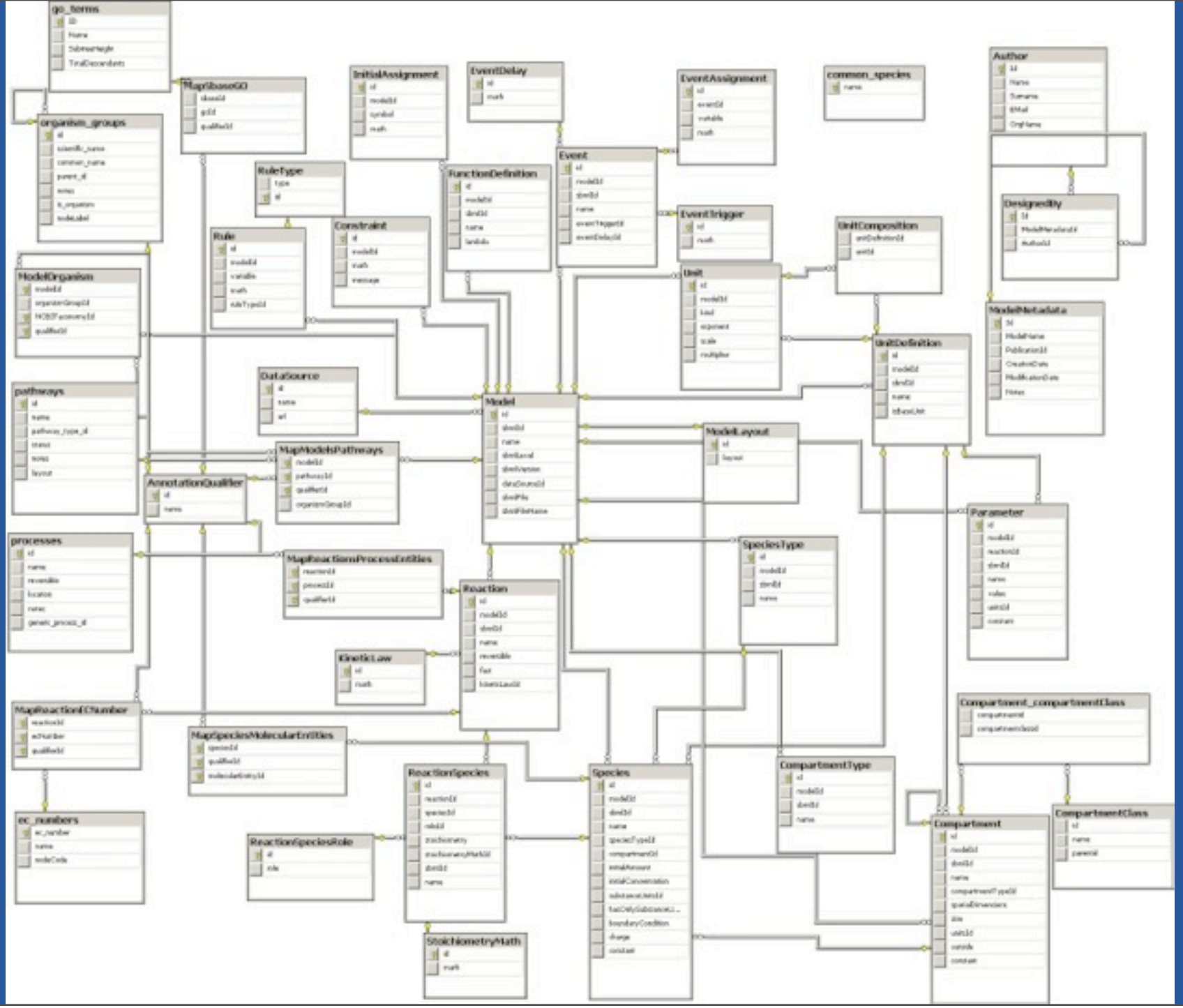
Scaling RDBMS...
Costs go up

VERTICAL SCALE

Productivity goes down

Productivity goes down

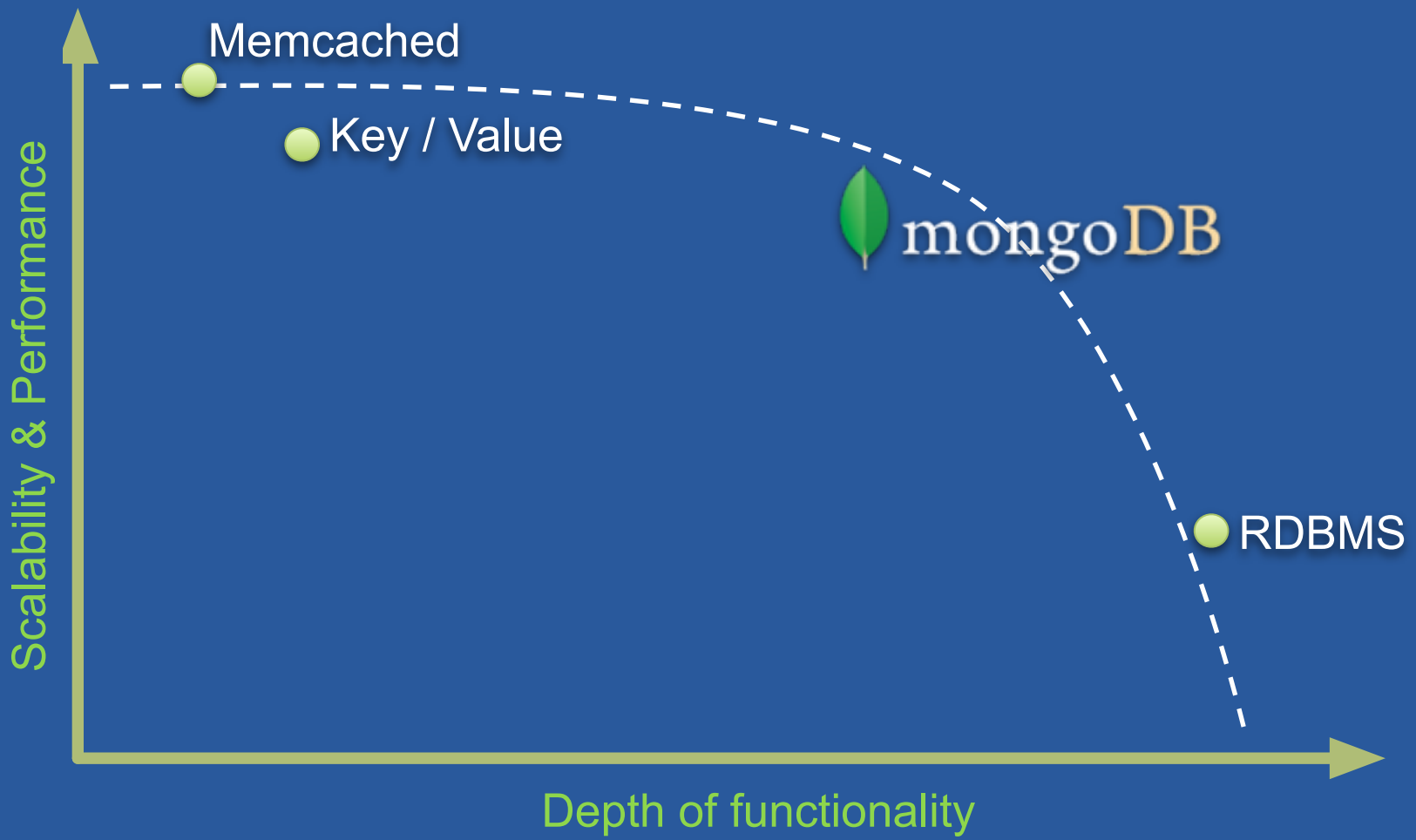



go_terms
MapGbaseGO
organism_groups
InitialAssignment
EventDelay
EventAssignment
common_species
Author
RuleType
FunctionDefinition
Event
EventTrigger
UnitComposition
Designedby
Rule
Constraint
Unit
ModelOrganism
ModelMetaData
UnitDefinition
DataSource
pathways
Model
ModelLayout
MapModelsPathways
AnnotationQualifier
Parameter
processes
MapReactionsProcessEntities
SpeciesType
Reaction
KineticLaw
MapReactionsChamber
Compartment_compartmentClass
MapSpeciesMolecularEntities
ReactionSpecies
Species
CompartmentType
Compartment
CompartmentClass
ec_numbers
ReactionSpeciesRole
StoichiometryMath

Tables to Documents

```
{
  title: 'MongoDB',
  contributors: [
    { name: 'Eliot Horowitz',
      email: 'eliot@10gen.com' },
    { name: 'Dwight Merriman',
      email: 'dwight@10gen.com' }
  ],
  model: {
    relational: false,
    awesome: true
  }
}
```

As simple as possible, but no simpler



Documents

```
var p = { author: "roger",
          date: new Date(),
          text: "Spirited Away",
          tags: ["Tezuka", "Manga"]}

> db.posts.save(p)
```

Nested Documents

```
{  _id : ObjectId("4c4ba5c0672c685e5e8aabf3"),
   author : "roger",
   date : "Sat Jul 24 2010 19:47:11 GMT-0700 (PDT)",
   text : "Spirited Away",
   tags : [ "Tezuka", "Manga" ],
   comments : [
         {
                  author : "Fred",
                  date : "Sat Jul 24 2010 20:51:03 GMT-0700 (PDT)",
                  text : "Best Movie Ever"
         }
   ]}
```

Querying

```
>db.posts.find()

  { _id : ObjectId("4c4ba5c0672c685e5e8aabf3"),
    author : "roger",
    date : "Sat Jul 24 2010 19:47:11 GMT-0700 (PDT)",
    text : "Spirited Away",
    tags : [ "Tezuka", "Manga" ] }
```

Notes:

- _id is unique, but can be anything you'd like

Secondary Indexes

Create index on any Field in Document

```
>db.posts.ensureIndex({author: 1})
```

```
// Index nested documents
> db.posts.ensureIndex( "comments.author":1 )
> db.posts.find({'comments.author':'Fred'})

// Index on tags
> db.posts.ensureIndex( tags: 1)
> db.posts.find( { tags: 'Manga' } )

// geospatial index
> db.posts.ensureIndex({ "author.location":"2d" )
> db.posts.find( "author.location" : { $near : [22,42] } )
```

Query Operators

- Conditional Operators
  - $all, $exists, $mod, $ne, $in, $nin, $nor, $or, $size, $type
  - $lt, $lte, $gt, $gte

```
// find posts with any tags
> db.posts.find( {tags: {$exists: true }} )

// find posts matching a regular expression
> db.posts.find( {author: /^rog*/i } )

// count posts by author
> db.posts.find( {author: 'roger'} ).count()
```

Aggregation and Processing

- Map/Reduce can be used for aggregation…
  - Currently being used for totaling, averaging, etc
  - Map/Reduce is a big hammer
- Aggregation Framework: Simple, Fast
  - No Javascript Needed
  - Filter or Select Only Matching Sub-documents or Arrays
- MongoDB Hadoop Connector
  - Useful for Hadoop Integration
  - Large Batch Processing Jobs

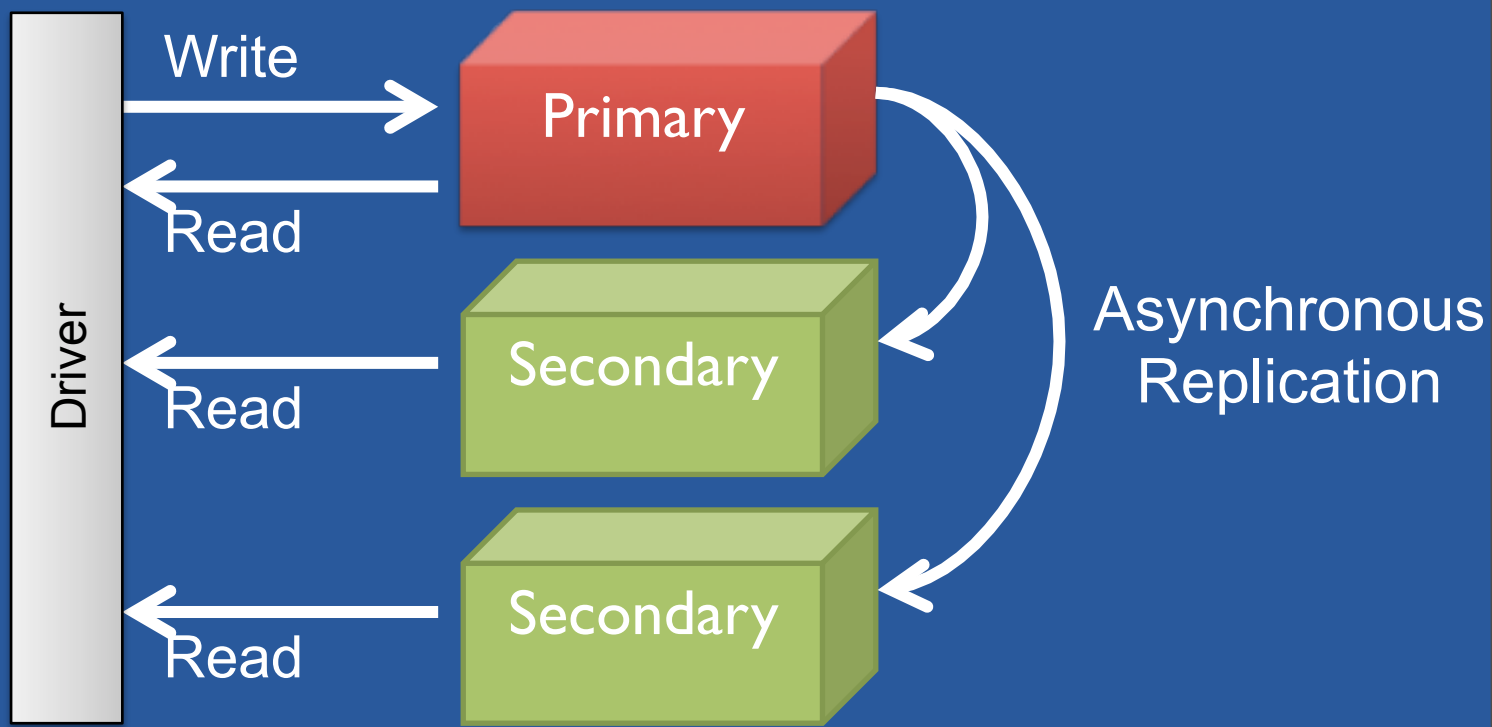
Atomic Operations

- $set, $unset, $inc, $push, $pushAll, $pull, $pullAll, $bit

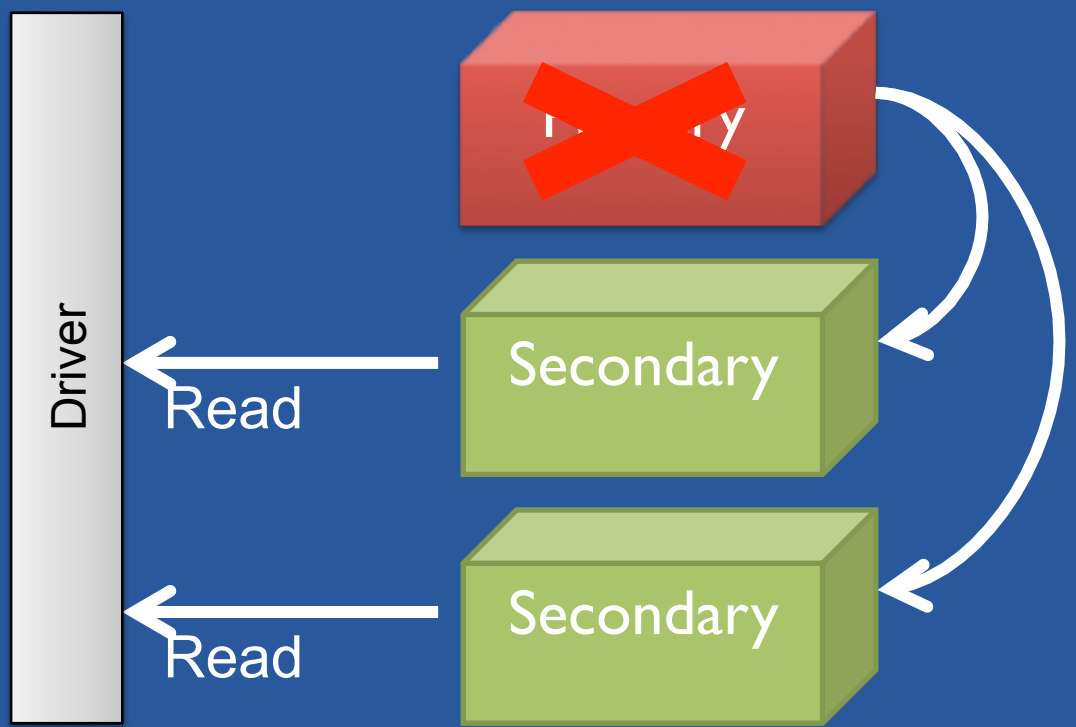
```
> comment = { author:"fred",
              date: new Date(),
              text:"Best Movie Ever"}

> db.posts.update( { _id:"..." },
                   $push: {comments: comment} );
```

DEPLOYMENT & SCALING

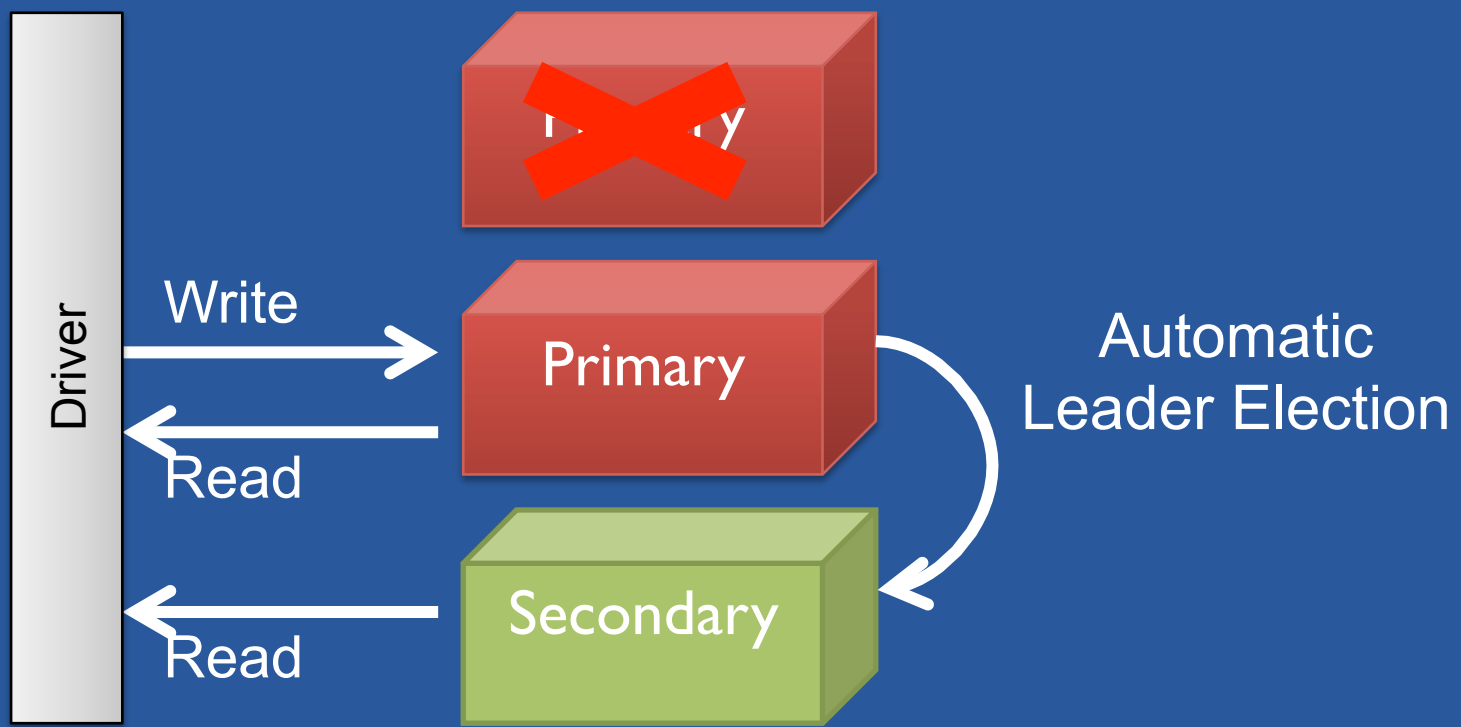
Replica Sets



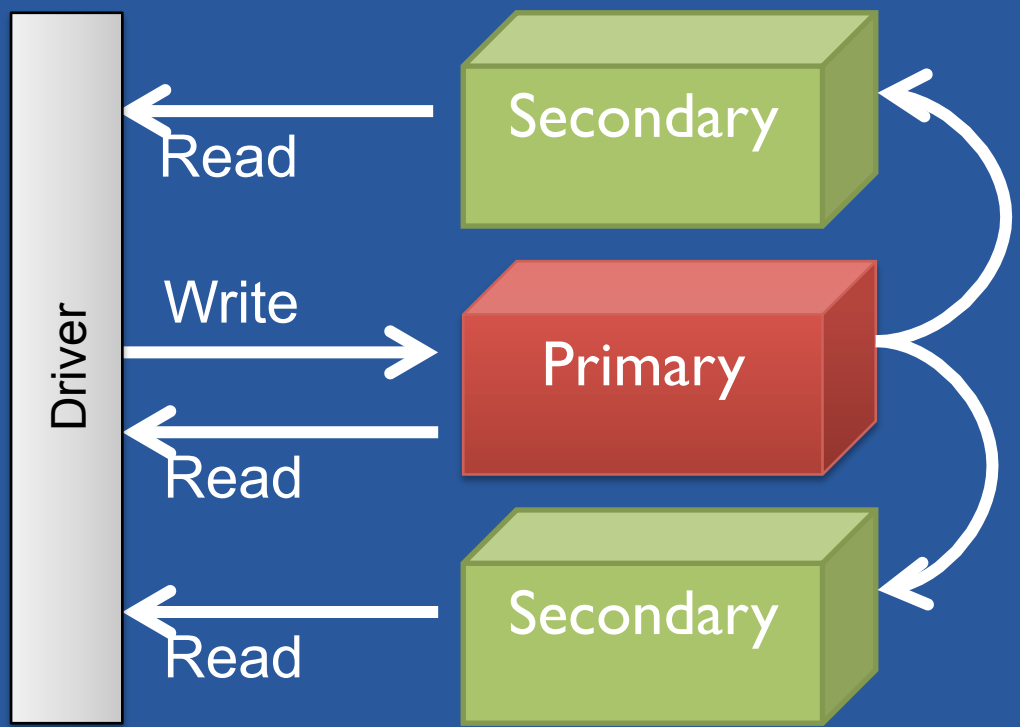
Replica Sets



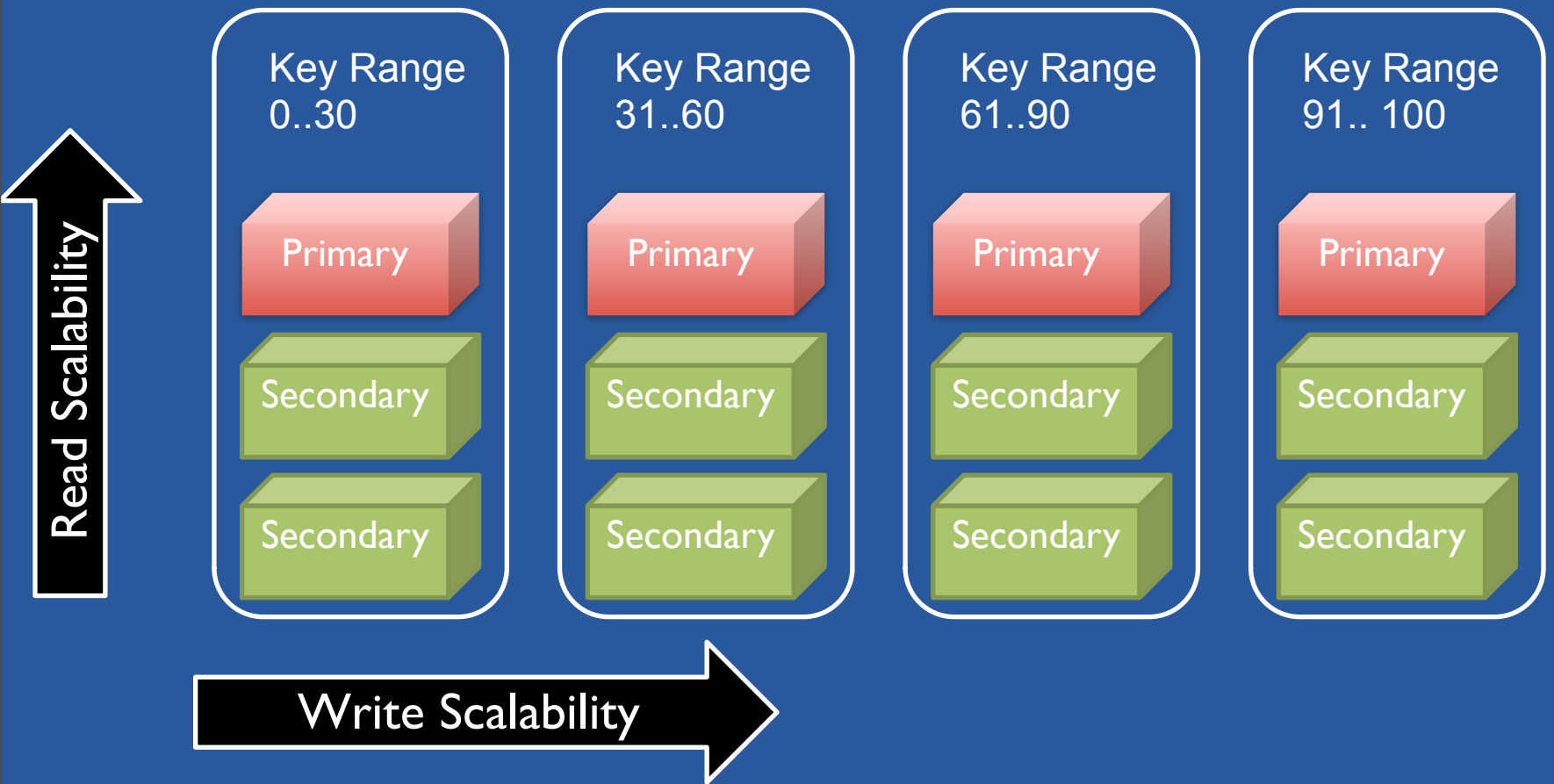
Replica Sets



Replica Sets



Sharding


Key Range
0..30
Primary
Secondary
Secondary
Key Range
31..60
Primary
Secondary
Secondary
Key Range
61..90
Primary
Secondary
Secondary
Key Range
91.. 100
Primary
Secondary
Secondary
Read Scalability
Write Scalability

Splitting

T1

Min Key	Max Key	Shard
- ∞	+ ∞	Shard 1

Splitting

T1

Min Key	Max Key	Shard
- ∞	+ ∞	Shard 1

T2

Min Key	Max Key	Shard
- ∞	Jared	Shard 1
Jared	+ ∞	Shard 2

Splitting

T1

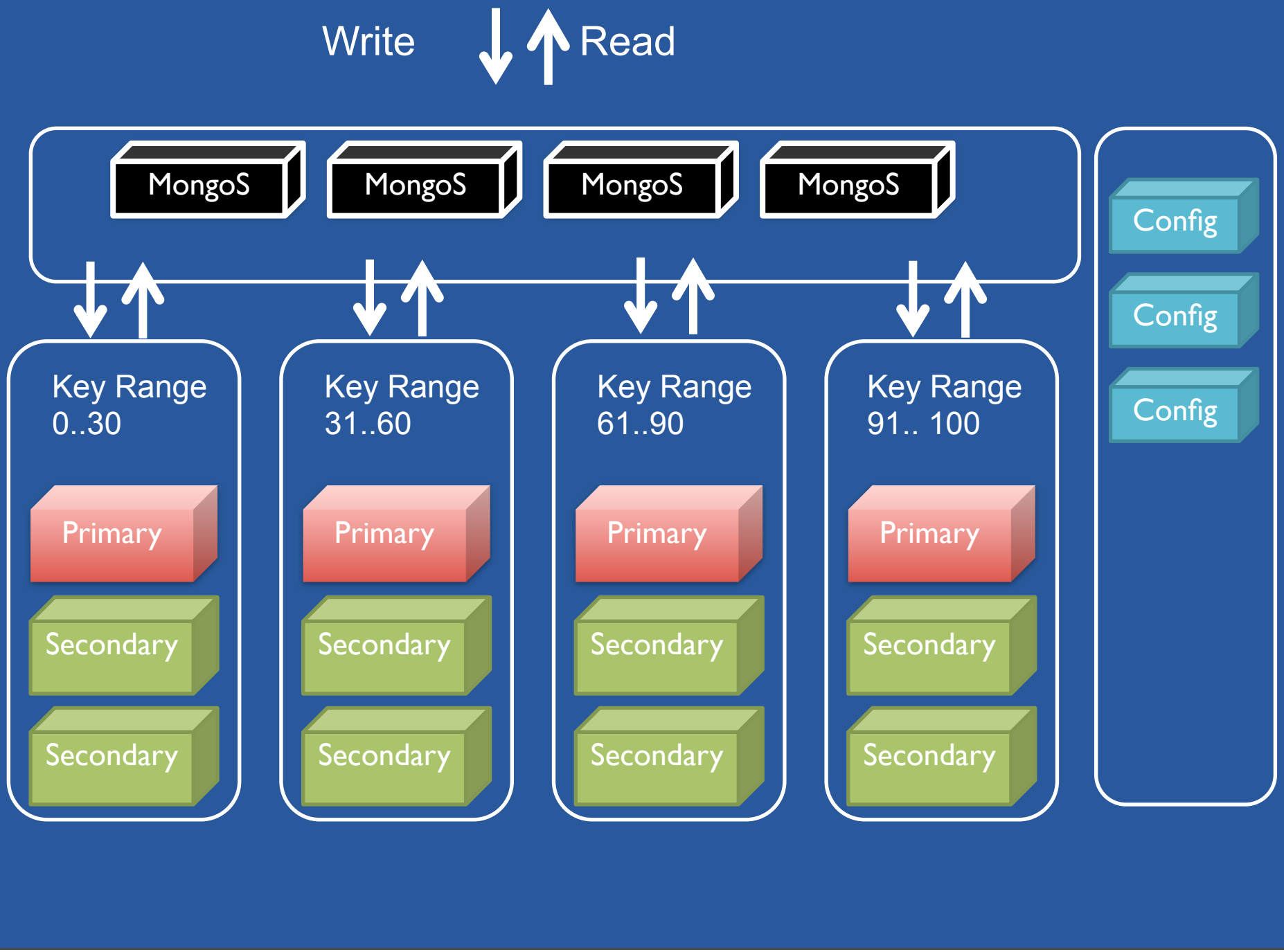
Min Key	Max Key	Shard
- ∞	+ ∞	Shard 1

T2

Min Key	Max Key	Shard
- ∞	Jared	Shard 1
Jared	+ ∞	Shard 2

T3

Min Key	Max Key	Shard
- ∞	Jared	Shard 1
Jared	Sam	Shard 2
Sam	+ ∞	Shard 3


Write
Read
MongoS
MongoS
MongoS
MongoS
Config
Config
Config
Key Range
0..30
Key Range
31..60
Key Range
61..90
Key Range
91.. 100
Primary
Primary
Primary
Primary
Secondary
Secondary
Secondary
Secondary
Secondary
Secondary
Secondary
Secondary

OTHER NON-RELATIONAL TECHNOLOGIES

WHAT'S THE DIFFERENCE??

General Comparison

	Cassandra	HBase	MongoDB
Data Model	Wide Column	Wide Column	Document
Query Model	Key + Column Range	Key + Column Range	Full Query Language
Consistency	Eventual	Strong	Tunable
Storage Engine	Append Only	Append Only	In Place
Distribution	Consistent Hash	Range	Range w/ Auto Balancing

10gen mongoDB

Most other products fill a specific niche and can be good at that niche.

mongoDB is a general purpose, high performance database.

More info at http://www.mongodb.org/

conferences, appearances, and meetups
http://www.10gen.com/events


Facebook
http://bit.ly/mongofb


Twitter
@mongodb


LinkedIn
http://linkd.in/joinmongo

10gen mongoDB