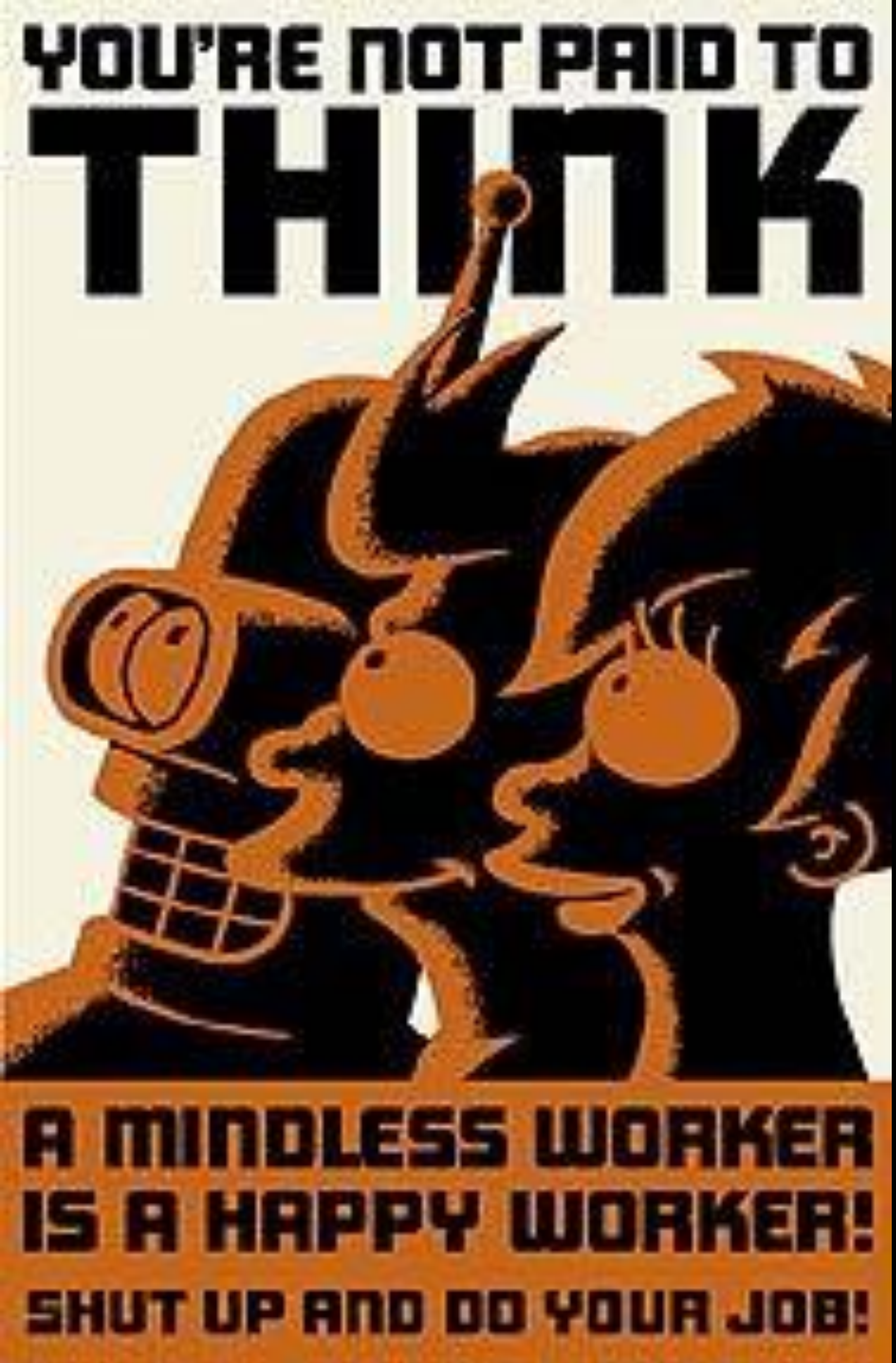


Following Google

Or

Don't Follow the Followers, Follow the Leaders

Or

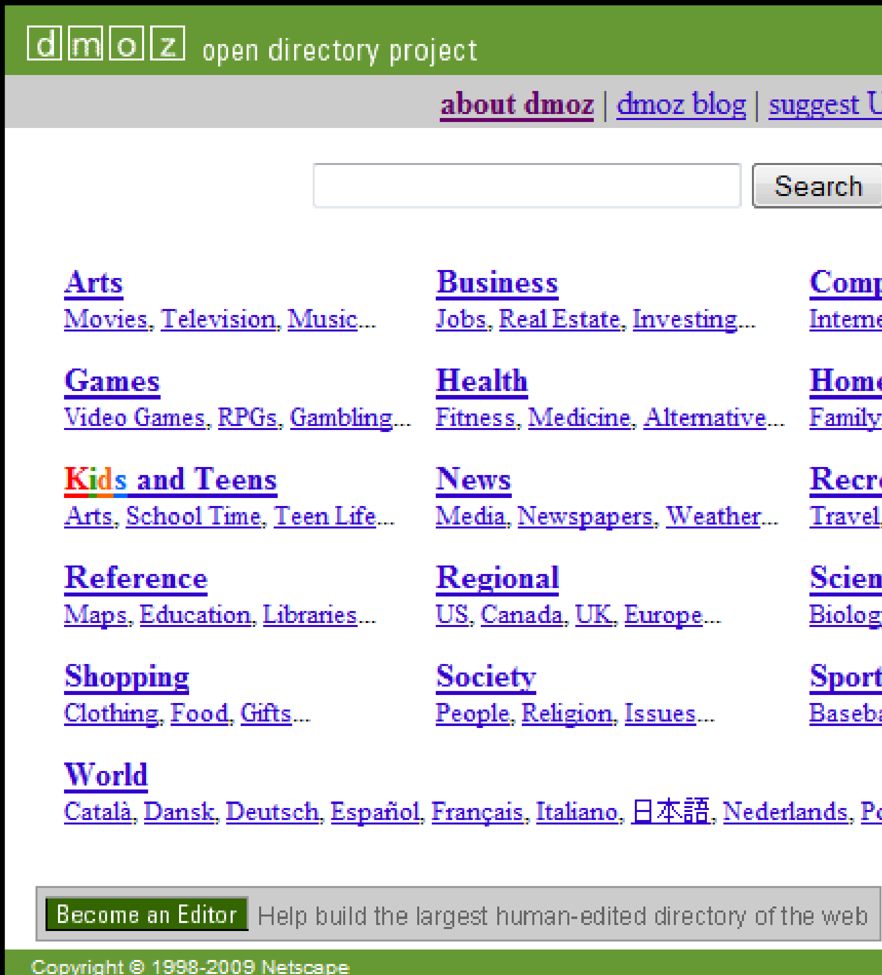
The problem probably isn't the database, the problem is probably you

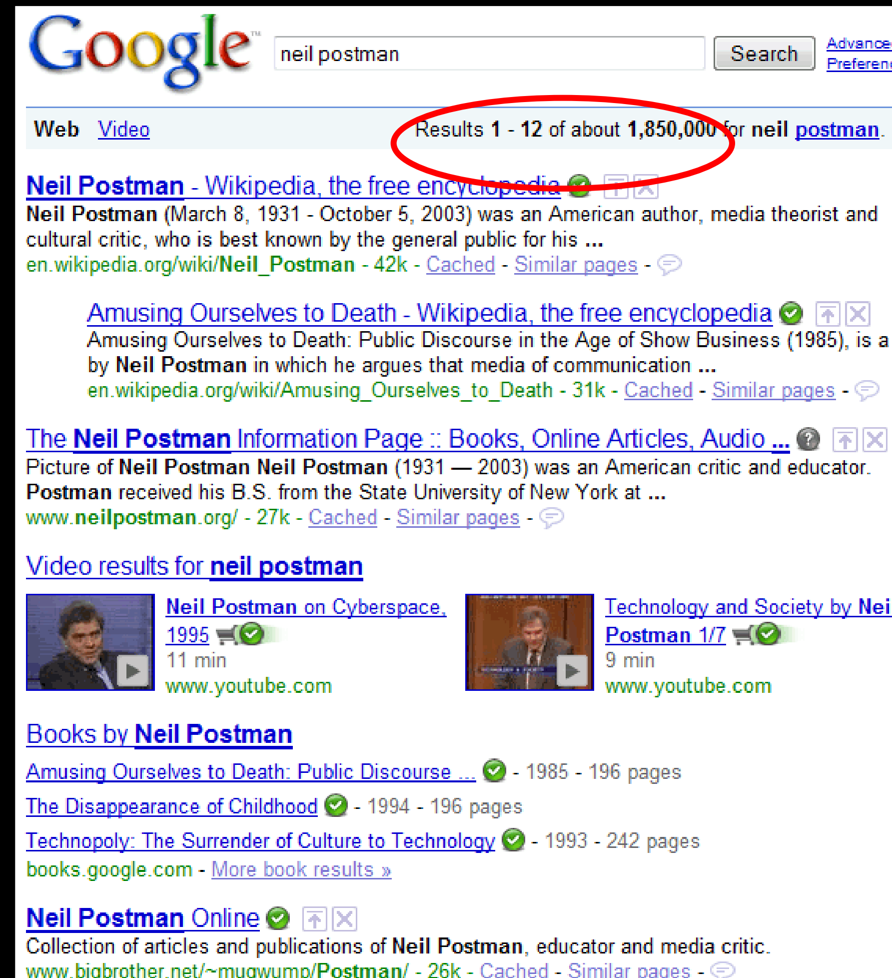
Qcon SF, November 2014

Mark Madsen
www.ThirdNature.net
@markmadsen

Third Nature
Third Nature

A Story of two companies

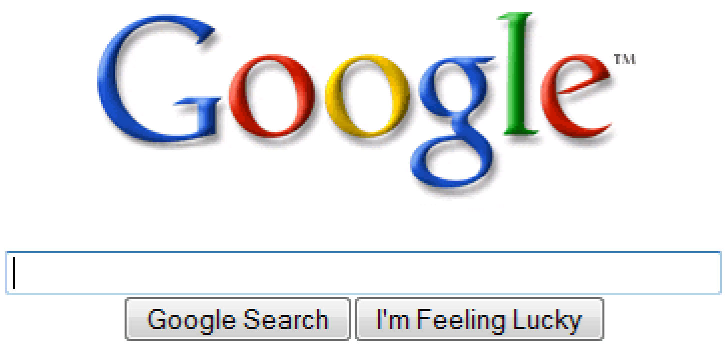




What happened to directories? Scalability: human & data volume.

Third Nature

Who are you following?



And why are you following them?

Third Nature

History isn’t taught in most university science curricula
(probably because it’s a rabbit hole)

A BRIEF HISTORY OF DATA STORAGE AND RETRIEVAL

Databases: the problem statements over time

“Information has become a form of garbage, not only incapable of answering the most fundamental human questions but barely useful in providing coherent direction to the solution of even mundane problems.” – ***Neil Postman, 1985***

"We have reason to fear that the multitude of books which grows every day in a prodigious fashion will make the following centuries fall into a state as barbarous as that of the centuries that followed the fall of the Roman Empire.” – ***Adrien Baillet, 1685***

”…so many books that we do not even have time to read the titles." – ***Anton Francesco Doni, 1550***

Third Nature

The origin of information management problems

MS 4631
Bulla-envelope with 11 plain and complex tokens inside.
Near East, ca. 3700-3200 BC

For ~5000 years we used counters of various types, eventually developing writing to cope with civilization’s needs.

Writing is more efficient than counters you can lose.

Sumerian bulla envelope with tokens.
The beta period.

Third Nature

Information Technology v1.0: Clay Tech, ~3000 bce

The first information explosion

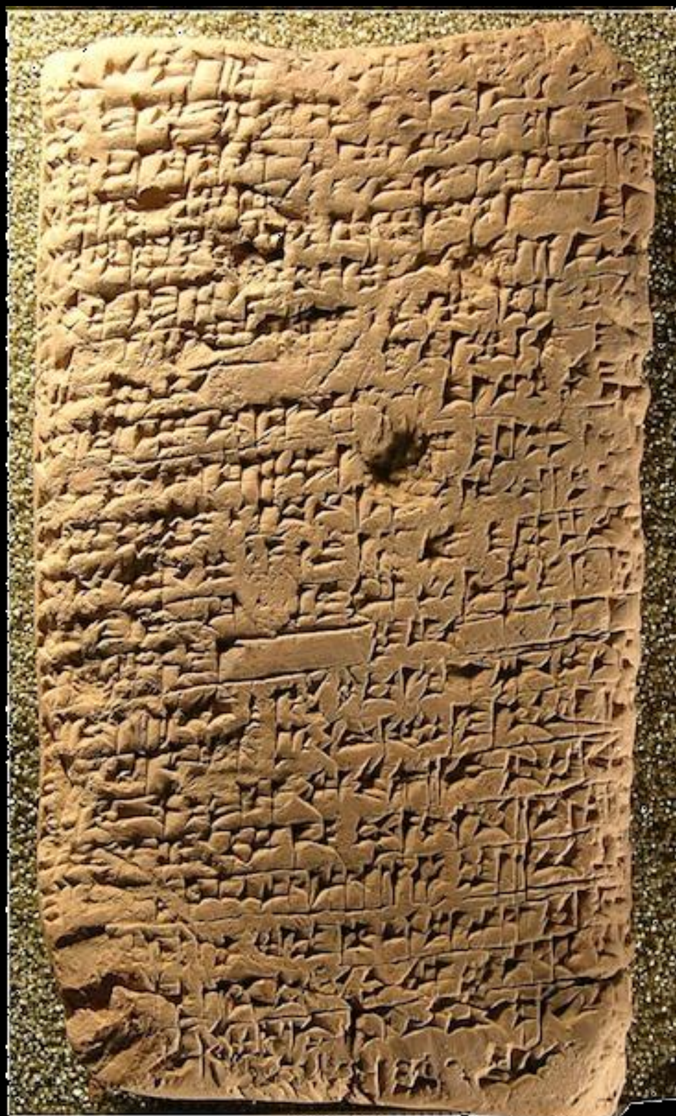
That explosion led to the first metadata

“tags”

Small piles in baskets are easy to tag and search

Third Nature

Metadata v1.1: tablets about tablets



MS 3391
Library catalogue. Babylonia, 2000-1600 BC

When there are enough of these lying around you need to work on organization of the collection by categorizations, aka “taxonomy”, “schema”

Like working out what tables are in a database, or what files are stored in HDFS.

Babylonian library catalog ~2000bce

Third Nature

Metadata v1.2: tablets about what's *in* tablets

When literacy rates are higher and people need to communicate more effectively, you need to invent mechanisms to cope, like dictionaries.

Now we're worried about what's inside the documents, not where they are placed.

Synonym list, Ashurbanipal, ~900 bce

Third Nature

Clay Tech has some familiar limitations

Information Management v2.0 Paper Tech*

Lighter, denser, faster storage media

Third Nature

More information = need for new metadata techniques: content tagging, author catalogs

The first real *library ~300BC*

Discovery of one tradeoff between clay and paper…

Recorded information creates permanence *and* instability

You can't have discontinuous reading* until you have a random access technology.

*Indexing and encyclopedias are hard in linear scrolls. Hello ISAM

Paper Tech v2.1: increased storage density, smaller form factor, durability, high res RGB graphics

Paper Tech v2.2

The change in printing over time accelerates.

Block printing replaced by movable metal type.

The job of production is faster and cheaper.

Commoditization changes the landscape over the next 200 years.

The *printed* becomes more important than the *printer*.

Third Nature

The Elizabethan Era

Production: printing presses

Data management tech:

- Perfect copies
- Topical catalogs
- Font standardization
- Taxonomy ascends

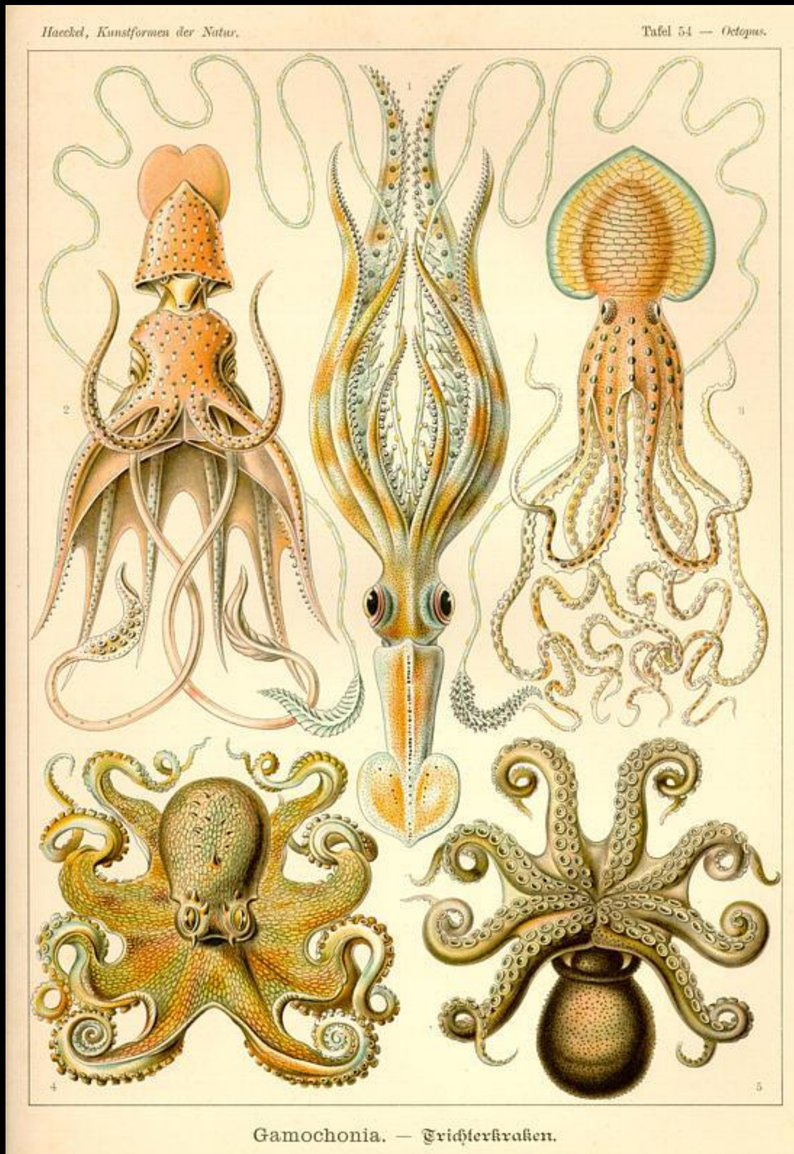
Information explosion:

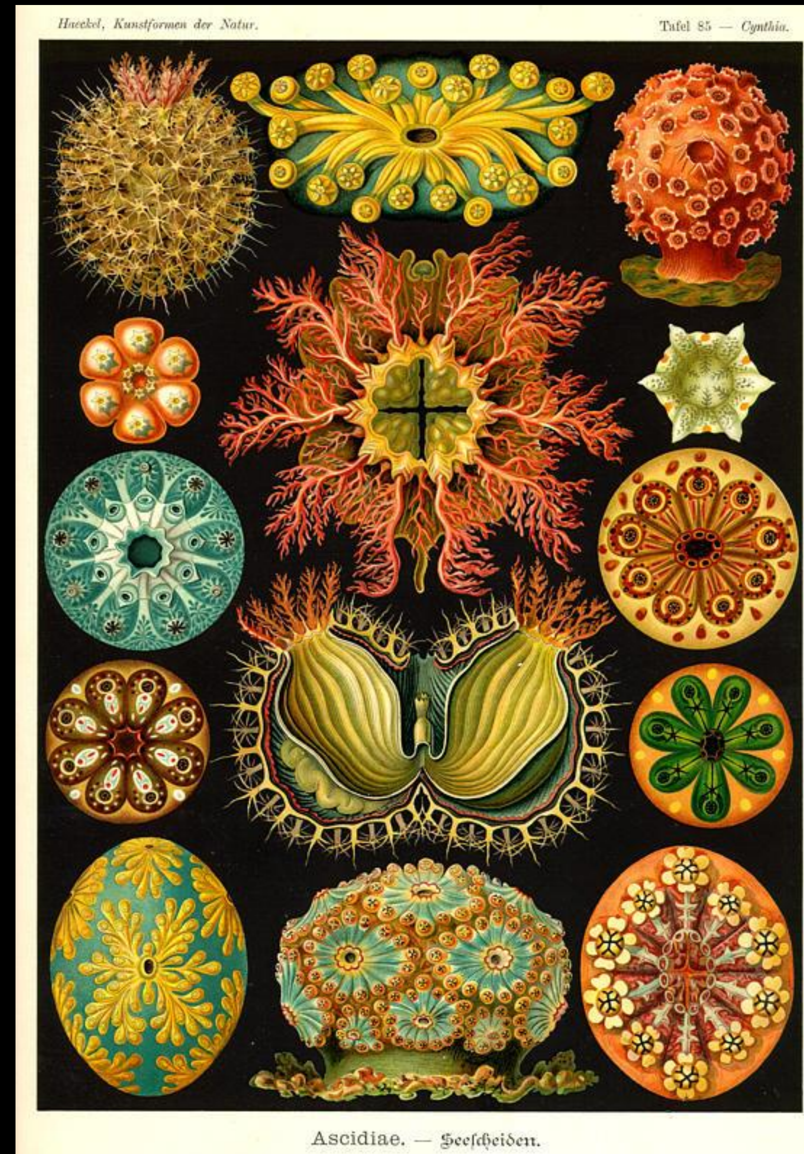
- 8M books in 1500
- 200M by 1600
- Commoditization
- Overload

Third Nature

Better *embedded* metadata: title page, colophon, ToC

The Georgian Era: The Explosion of Natural Philosophy





Sharing knowledge in a larger community required common language, structure

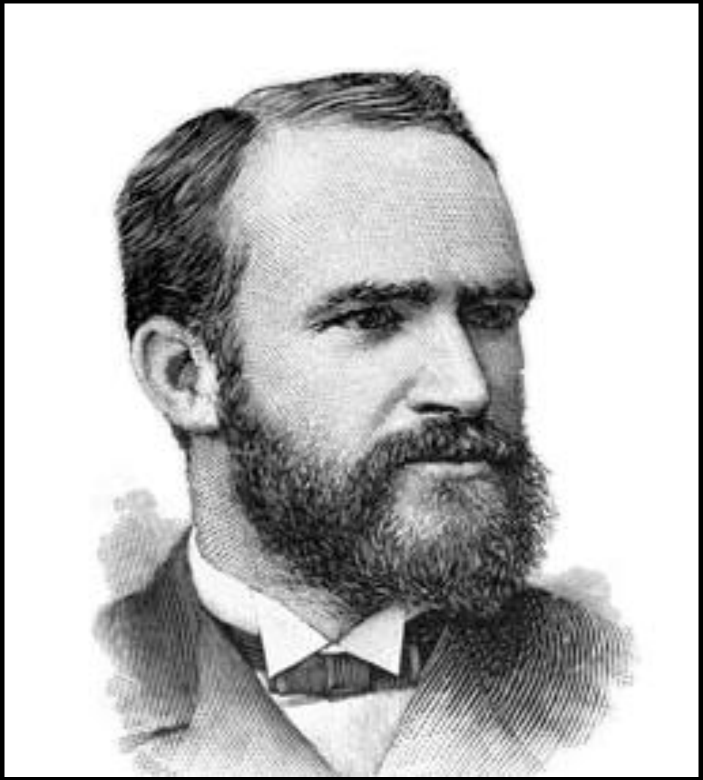
The Victorian Era

The powered printing information explosion:

- Card catalogs, cross-referencing, random access metadata
- Universal classification
- Extended information management debates
- Trading effort and flexibility for storage and retrieval
- <u>Stereotyping</u>

Third Nature

Melvil Dewey



Dewey Decimal System

Top down orientation

Static structure

Descriptive rather than explanatory

Taxonomic classification

Charles Ammi Cutter

Cutter Expansive Classification System (~1882)

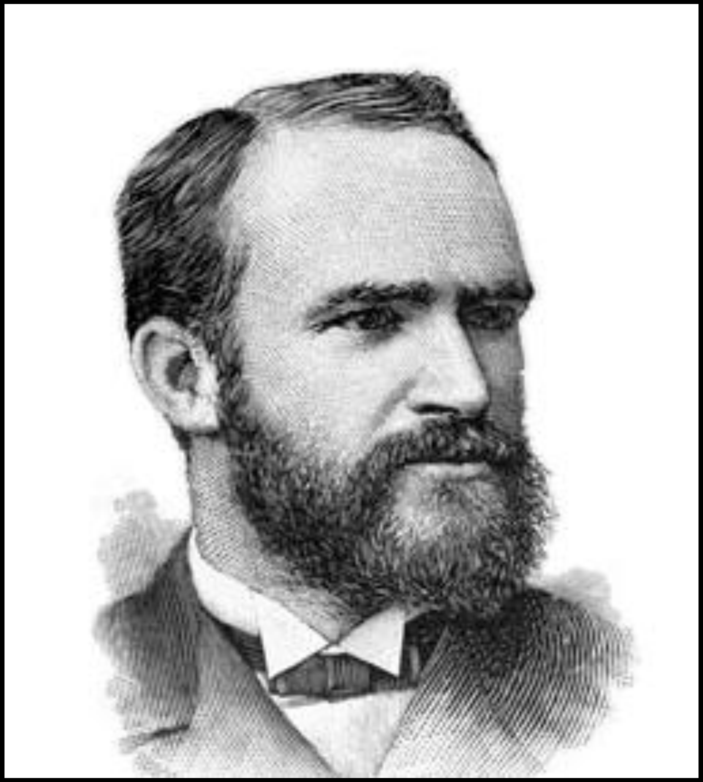
Bottom up orientation

More flexible structure

Explanatory, descriptive

Faceted classification

SQL



vs

NoSQL

Third Nature

History is always the same

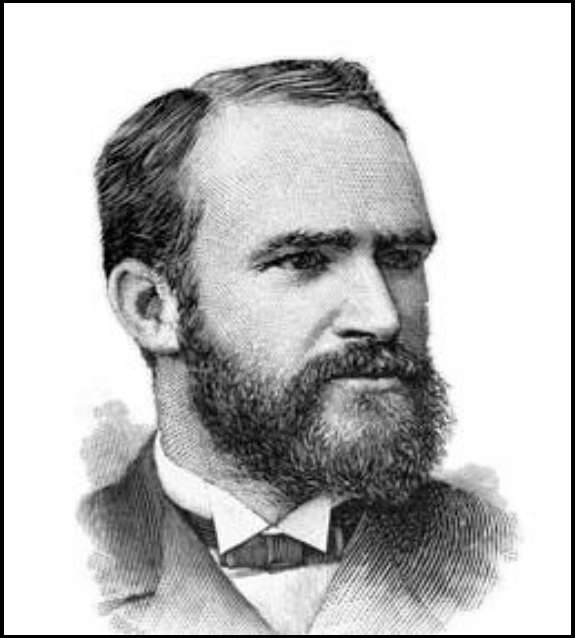
Every technology is a trade:

- Top down vs. bottom up
- Authority vs. anarchy
- Bureaucracy vs. autonomy
- Control vs. creativity
- Hierarchy vs. network
- Dynamic vs. static
- Power vs. ease
- Work up front vs. postponed

In every choice, something is lost and something is gained.

So why did Dewey beat Cutter?

Pragmatism



Good enough wins the day

It wasn't solving the problem you thought it was.

In every choice, something is lost when something is gained.

Third Nature

What has this to do with data and persistence?

"schema" is a broad term, a way of organizing and making something relatable and findable.

"Data" (or object) is to "Database"

as

"Books" are to "Library"

Summarizing

Thousands of years of thought have been put into principles of organization and use. The abstract patterns are the same, only the implementation changed.

- Clay: tablets about tablets, tablets about what's in tablets, 100X increase in data density over counting tech
- Scrolls: scrolls about scrolls, scrolls about what's in scrolls, prepended/appended navigation, >100X increase in density
- Books: books about books, books about what's in books, embedded internal navigation, >1000X increase in density
- Digitized data: similar, far denser, and *different because it isn't locked into physical forms*

Third Nature

Information management through human history always follows the same pattern

New technology development

creates

New methods to cope

creates

New information scale and availability

creates…

Big Data

Third Nature

"The most amazing achievement of the computer software industry is its continuing cancellation of the steady and staggering gains made by the computer hardware industry.“ - *Henry Peteroski*

DEALING WITH BIG: SOME SCALING HISTORY

Why doesn't your database scale?

Hipster bullshit

I can't get MySQL to scale

therefore

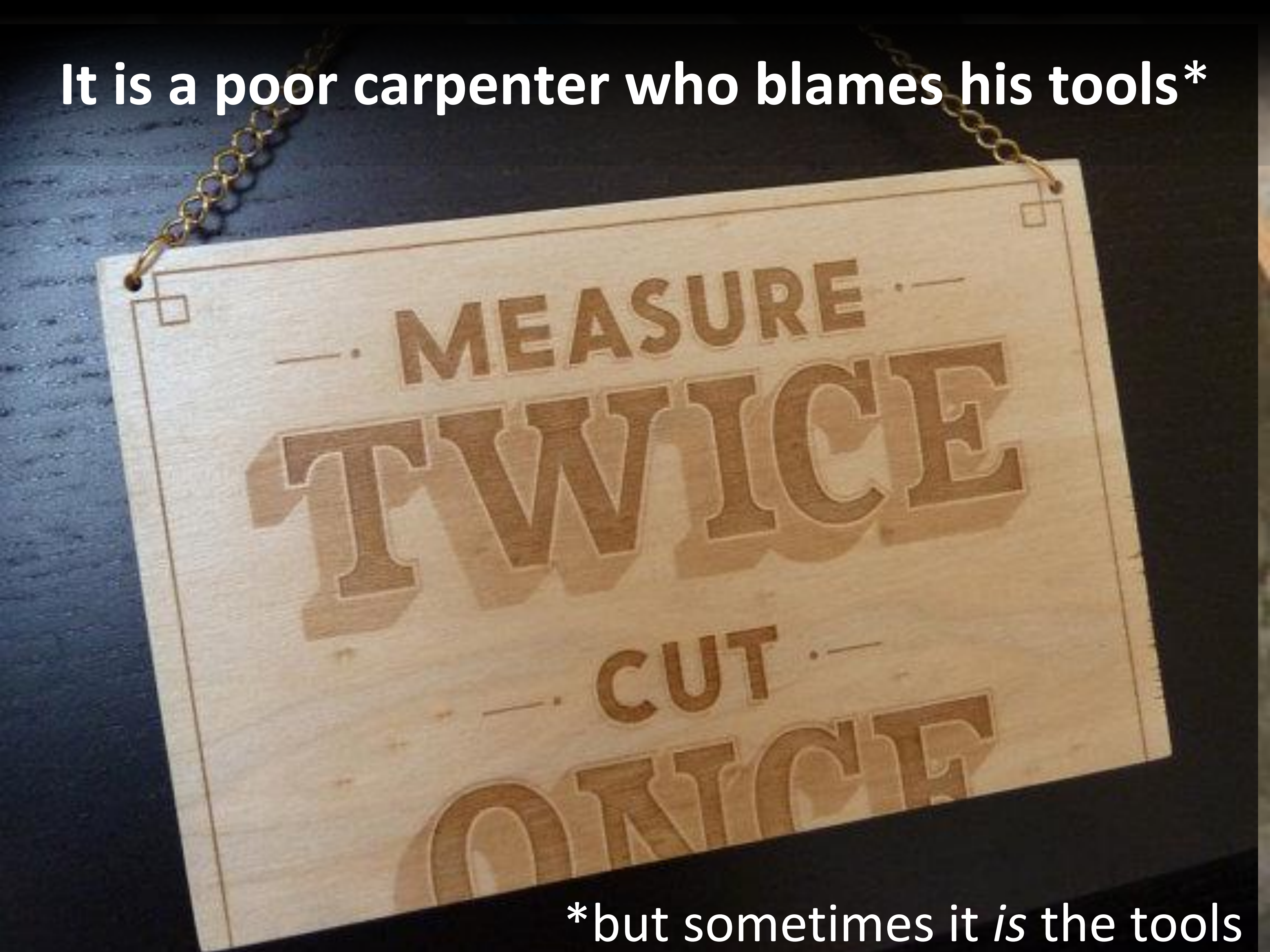
Relational databases don't scale

therefore

We must use NoSQL* for everything

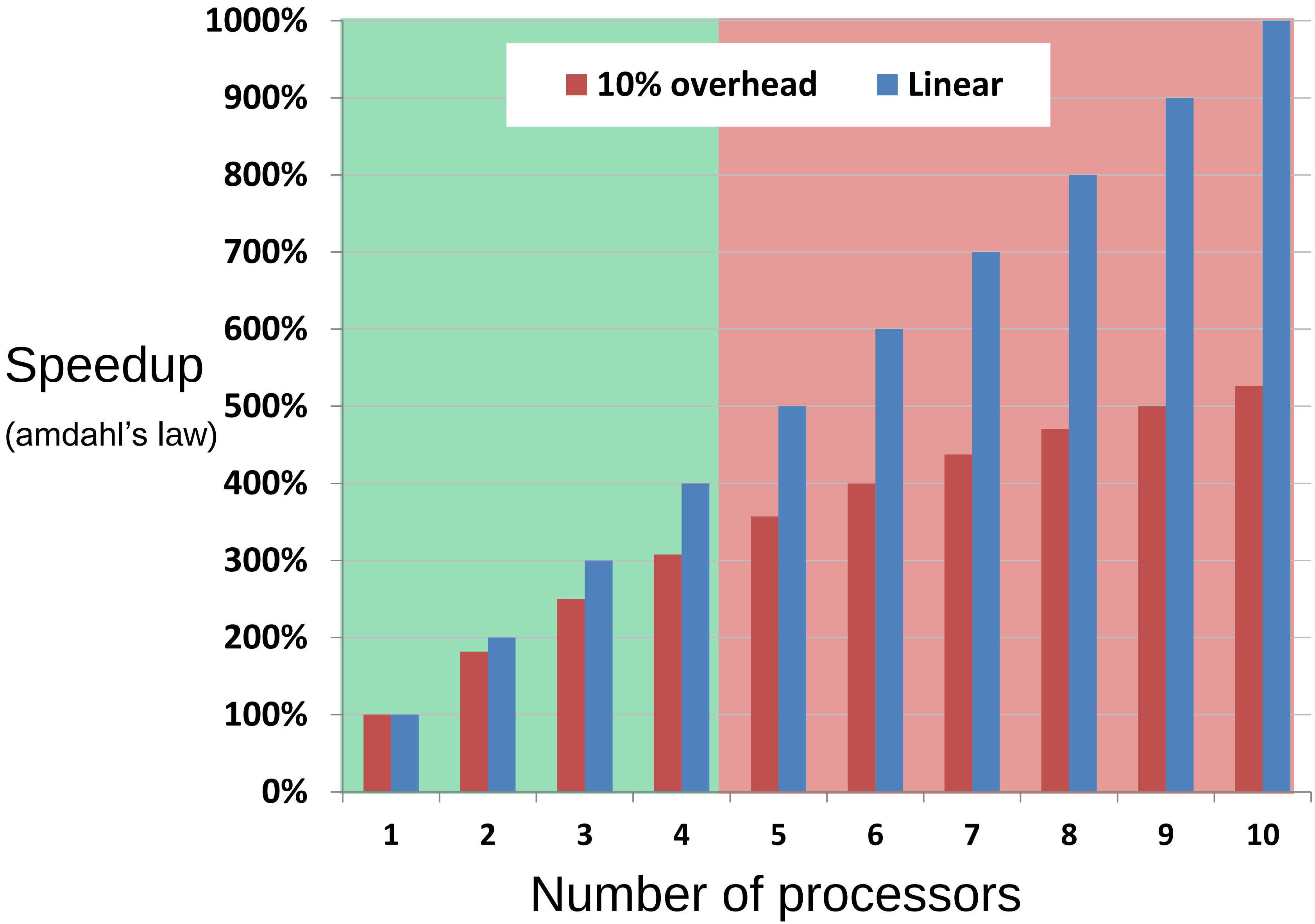
**including Hadoop and related*

It is a poor carpenter who blames his tools*



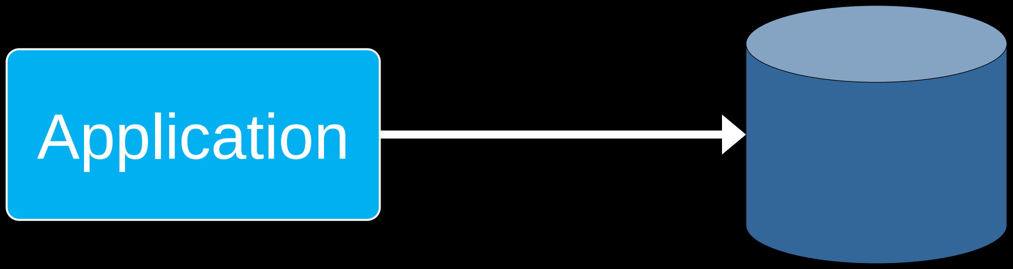
*but sometimes it *is* the tools

OMG just add nodez!



Third Nature

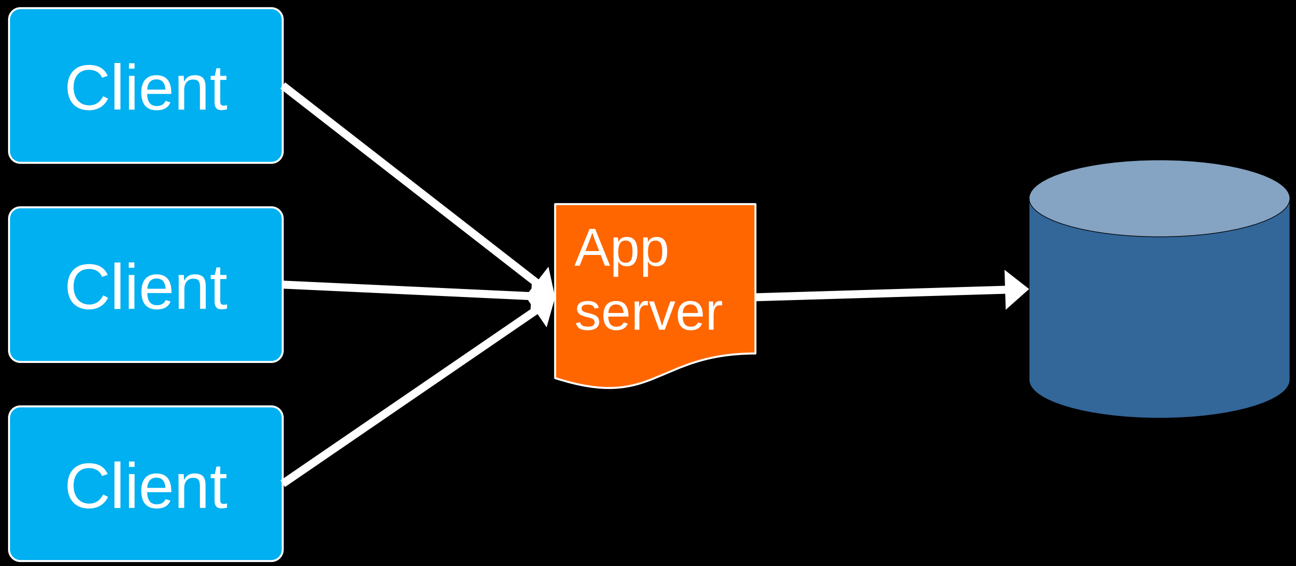
The early days: client/server as the starting point



We had transaction processing against the DB, all on the same machine. Then on two separate machines.

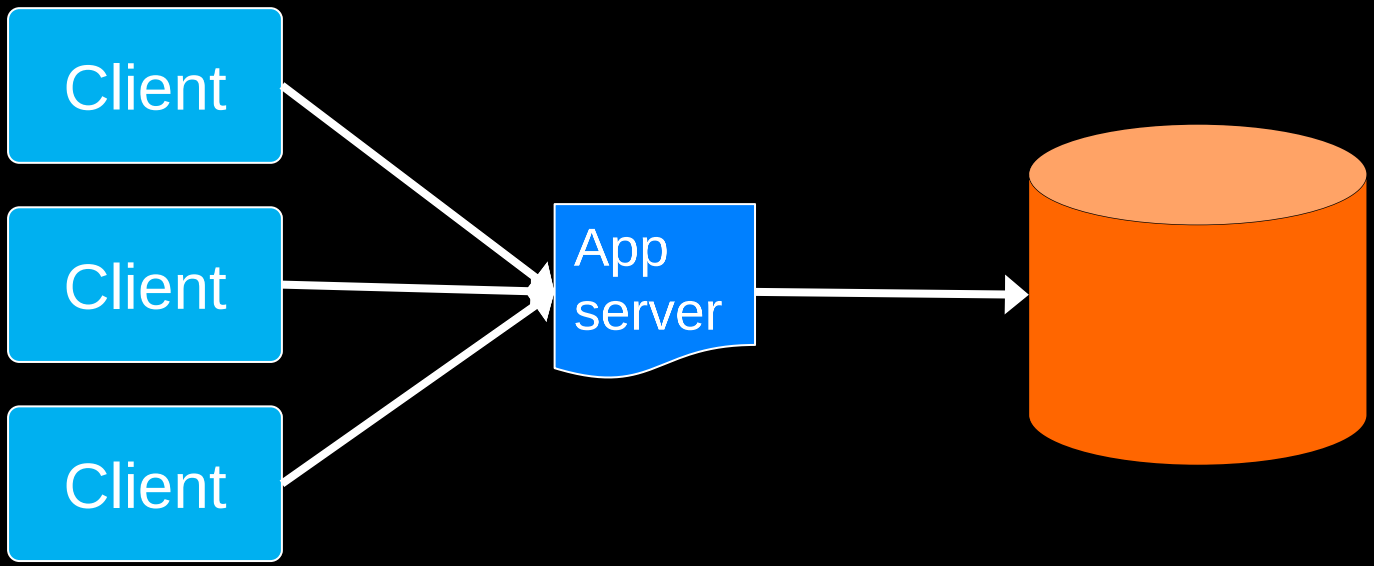
Third Nature

Scaling client/server



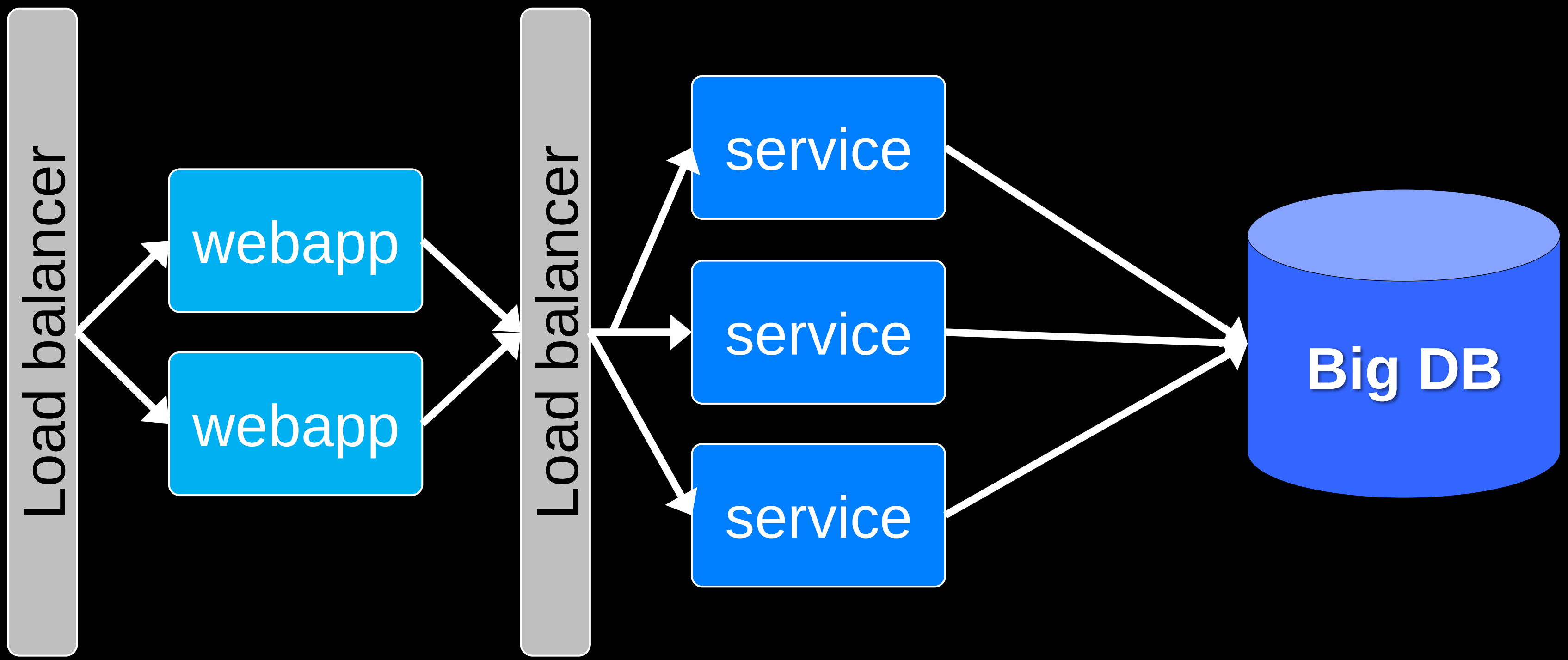
We added app servers and pooled connections.

Scaling client/server



Then threw money at the problem in the form of hardware (made the database bigger).

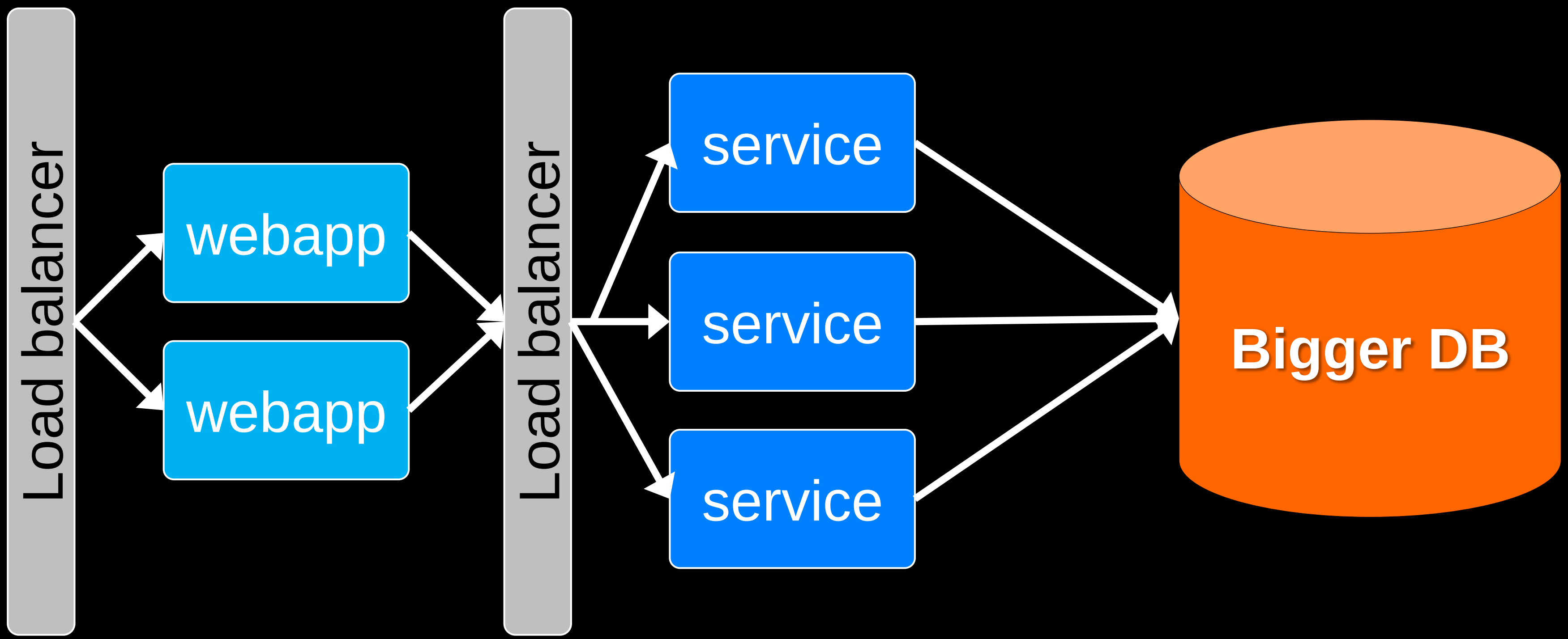
Web apps were a huge increase in concurrency



Architecture changed to reflect new stateless model.

We had scalability and availability problems.

Increasing traffic?

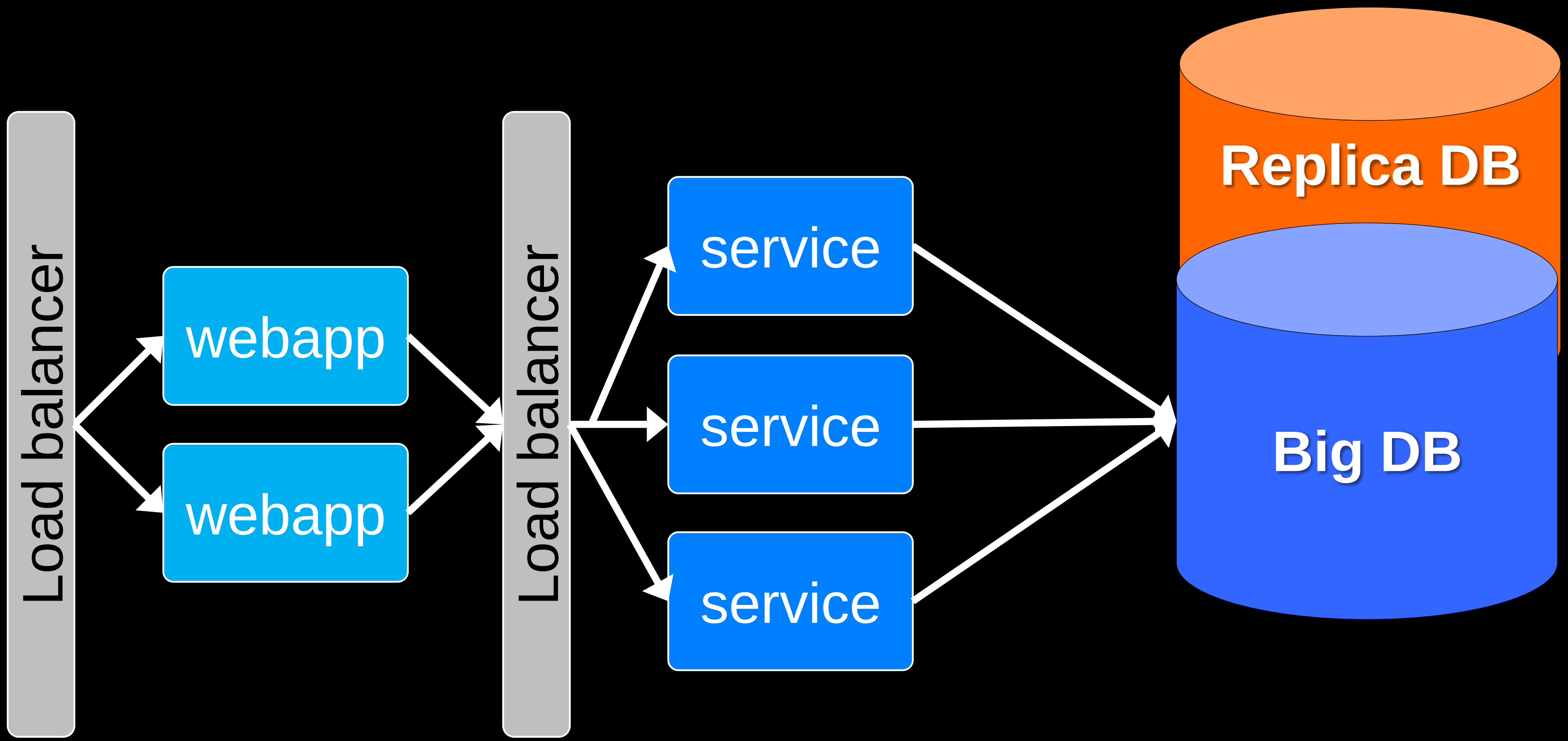


Keep adding hardware, make the DB bigger.

Limits reached, performance, scalability and availability problems.

Third Nature

Increasing traffic?

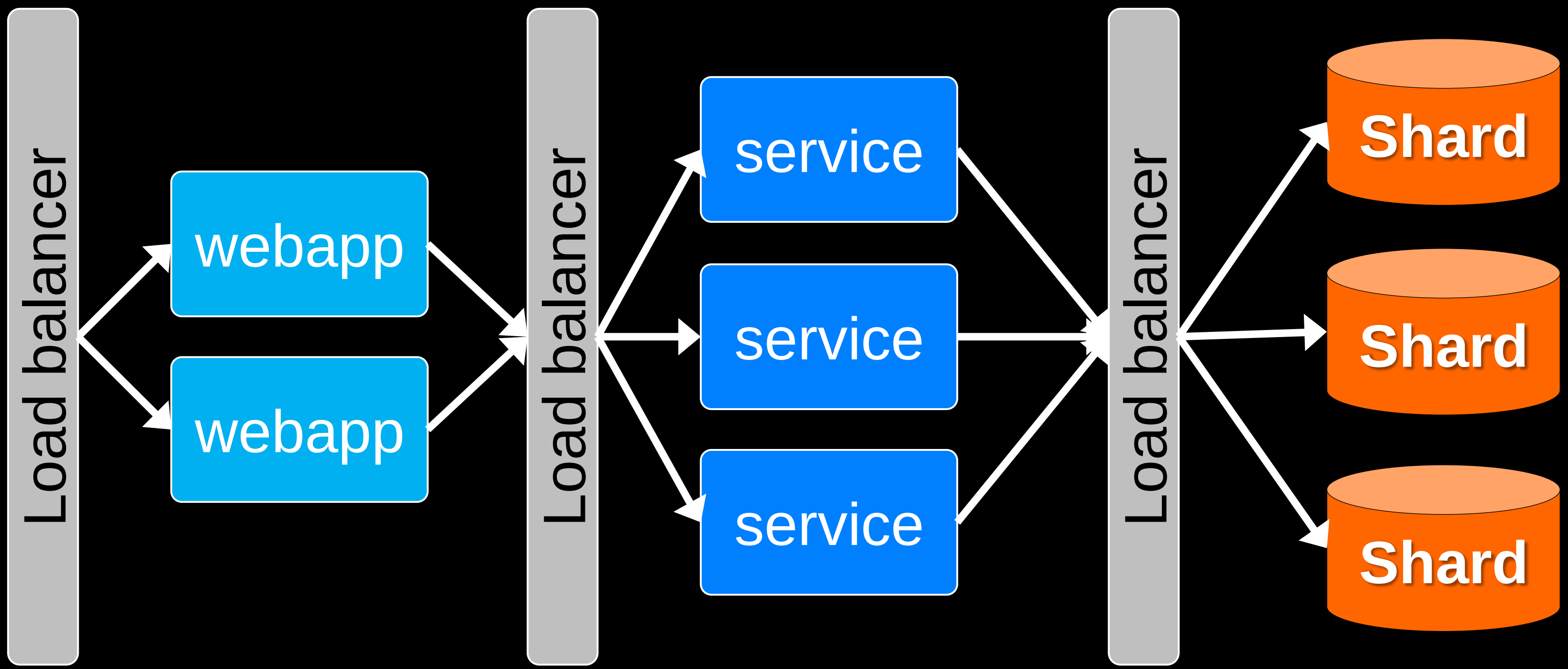


Read-only replicas will save the day!

Still have scalability and availability problems.

And now operational overhead and problems.

Increasing traffic?

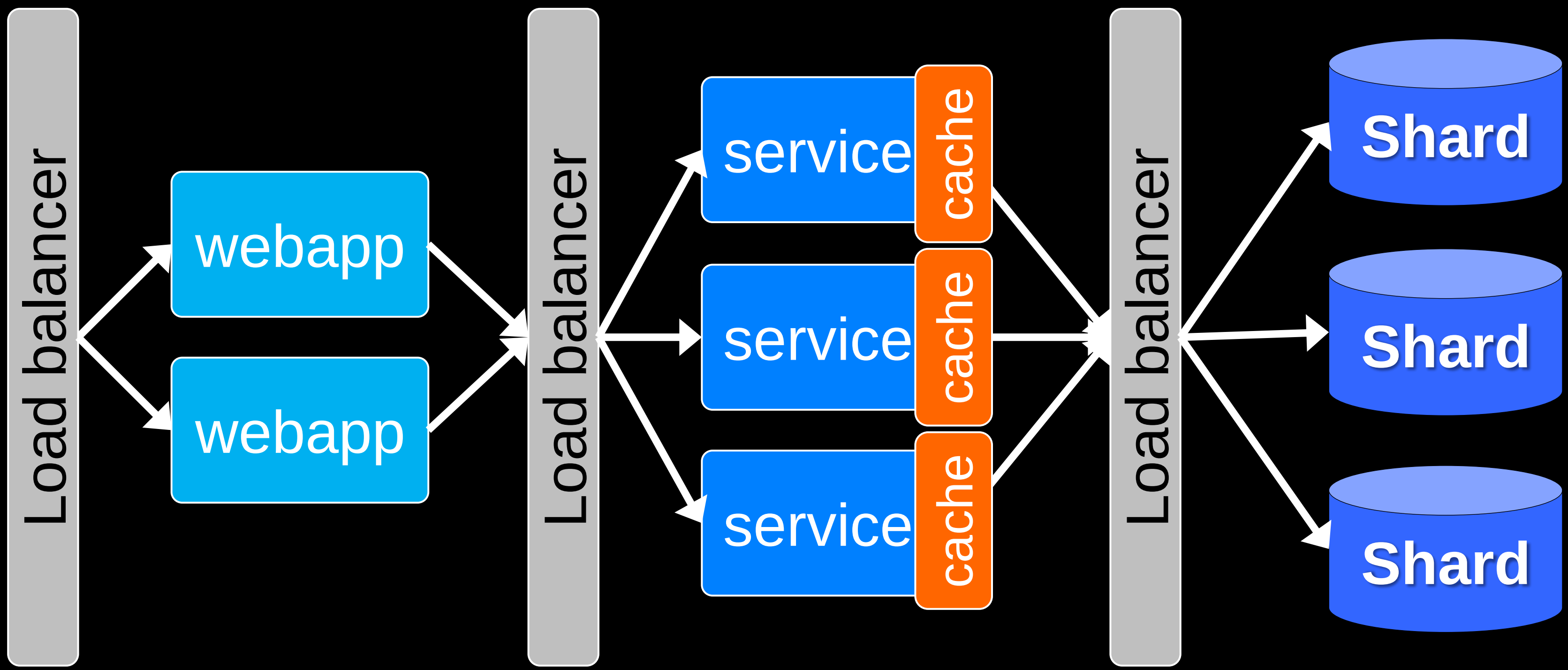


Scaling and perf better, overhead and operational complexity high and worsening.

Sharding seems a fine thing. But it's one letter from…

Third Nature

Increasing traffic?



Let's cache data at the service tier!

Performance better, overhead and operational complexity higher still.

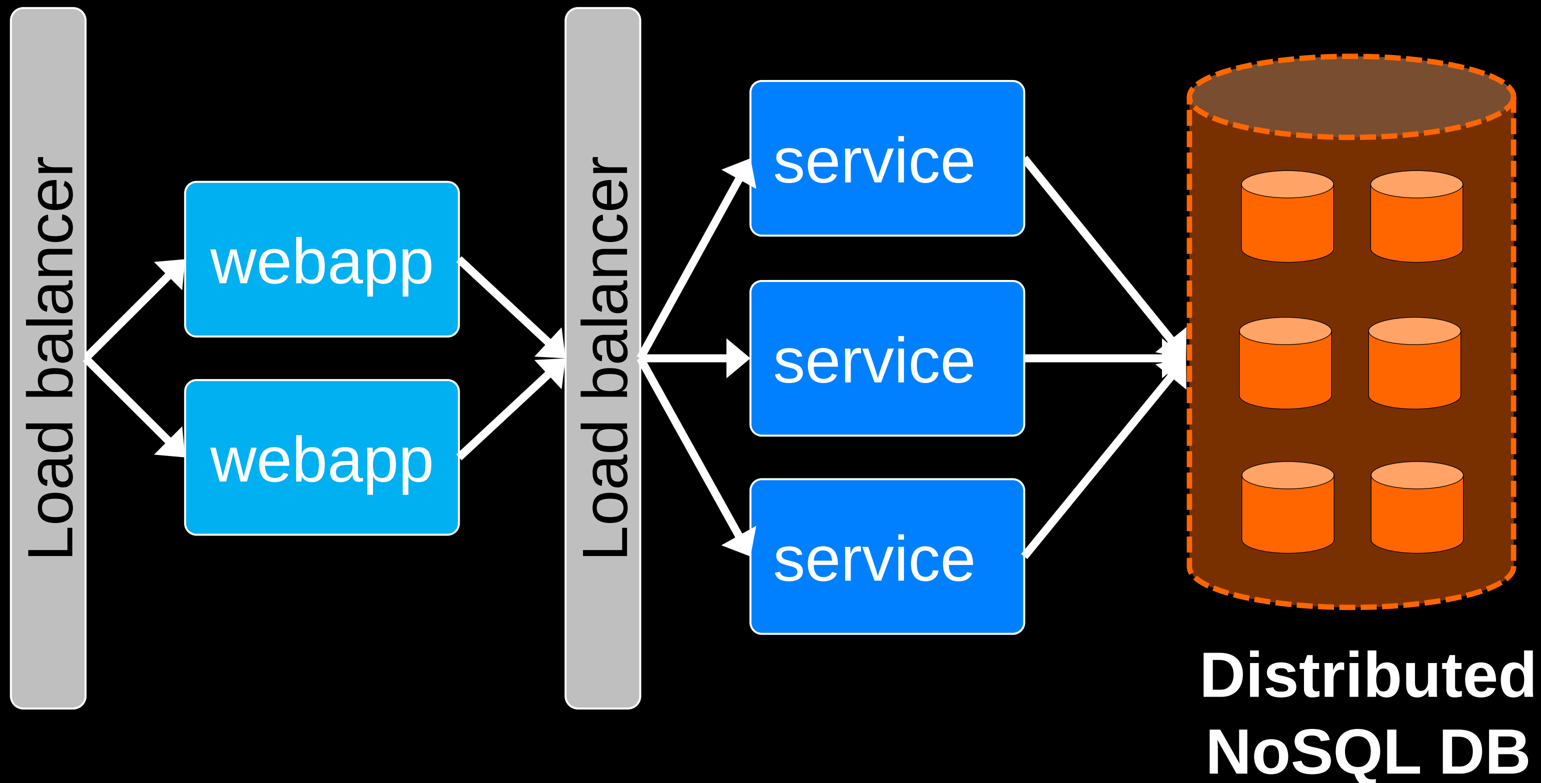
Third Nature

What are the problems now?

1. More hardware, more things to break
2. More management and administration
3. More software complexity
4. Increasing distance for data to travel = latency
5. Data administration difficult to impossible

Third Nature

Problem solved?



Distributed database (handles cache, load balance, data distribution). Similar performance, simpler scaling, reduced operational problems, simpler application architecture. Finished!

Third Nature

Not finished: remember the cycle of history…

The biggest hole in the prior section on scaling is that we scaled OLTP, what about OLAP (query)?

Queries <> transactions.

Solving query problems

Aggregate or low selectivity queries were a problem early on, when people wanted to *use* the data.

Every report or query is a program.

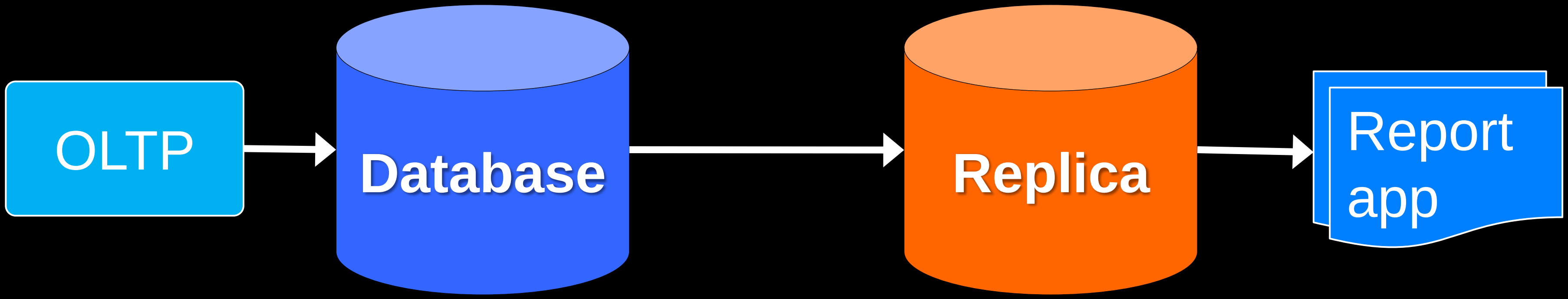
Third Nature

Increasing data volume

Make it faster by throwing money at hardware (sound familiar?)

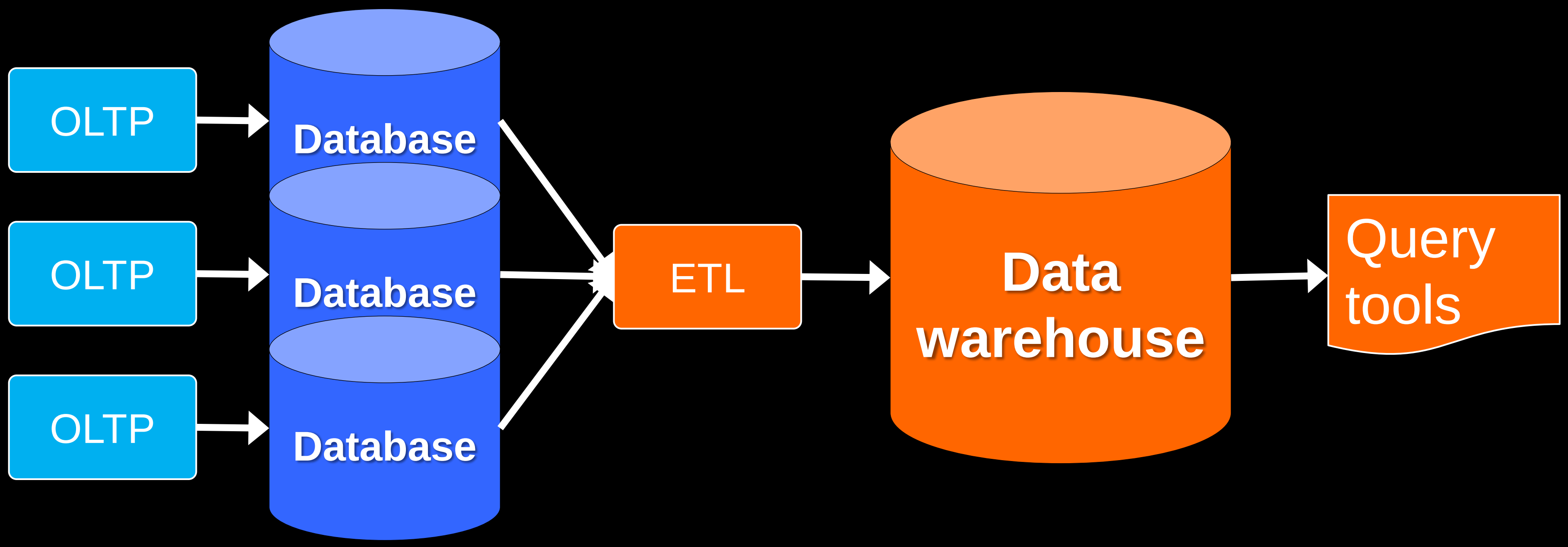
Third Nature

Increasing data volume



Replicas: split the workload and tune the systems based on their workload.

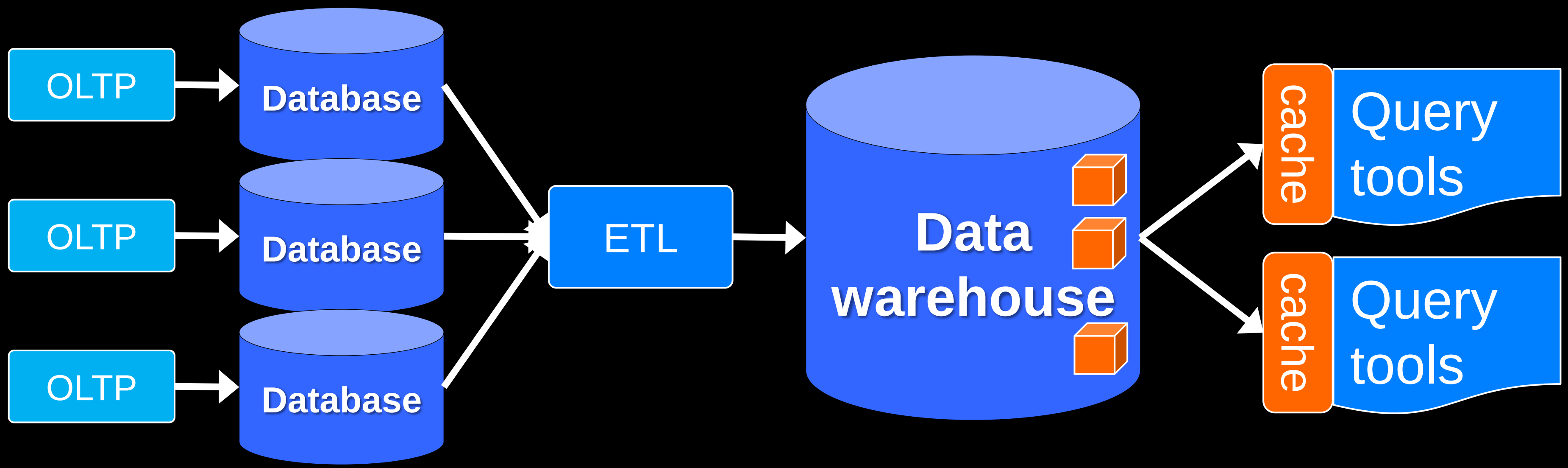
Increasing data volume breaks the old model



Devise a new architecture.

Reschematize the database, eliminate cyclic joins, selective denormalization, *query generators*. But it takes bulk processing to reschematize the data.

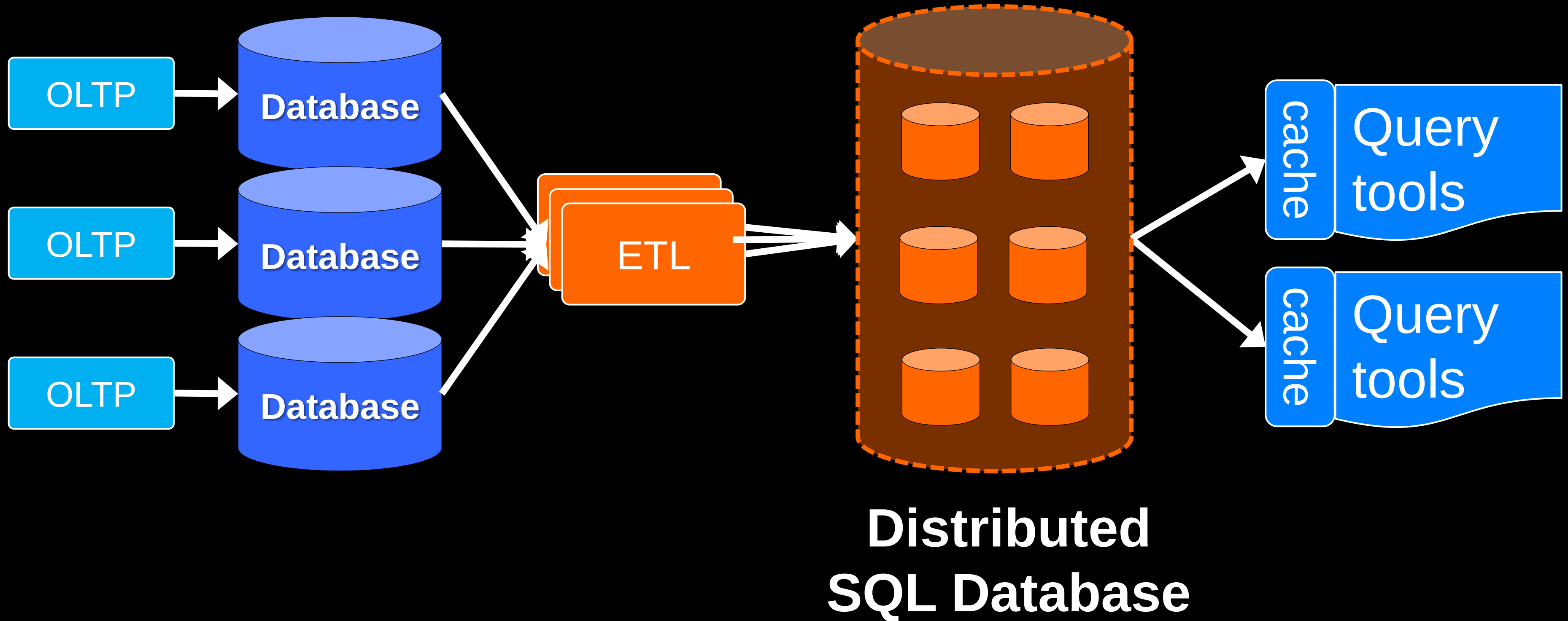
Increasing data volume



Improve response time with caching in the query tools, and by using MOLAP tools that map into cache or memory.

Third Nature

Increasing data volume



Parallel processing for ETL. Distributed query databases for fine grained high volume parallelism.

Third Nature

The architecture looks familiar

Two workloads, two not dissimilar architectures:

- Load-balanced front ends
- Distributed caching layers
- Scalable distributed parallel databases

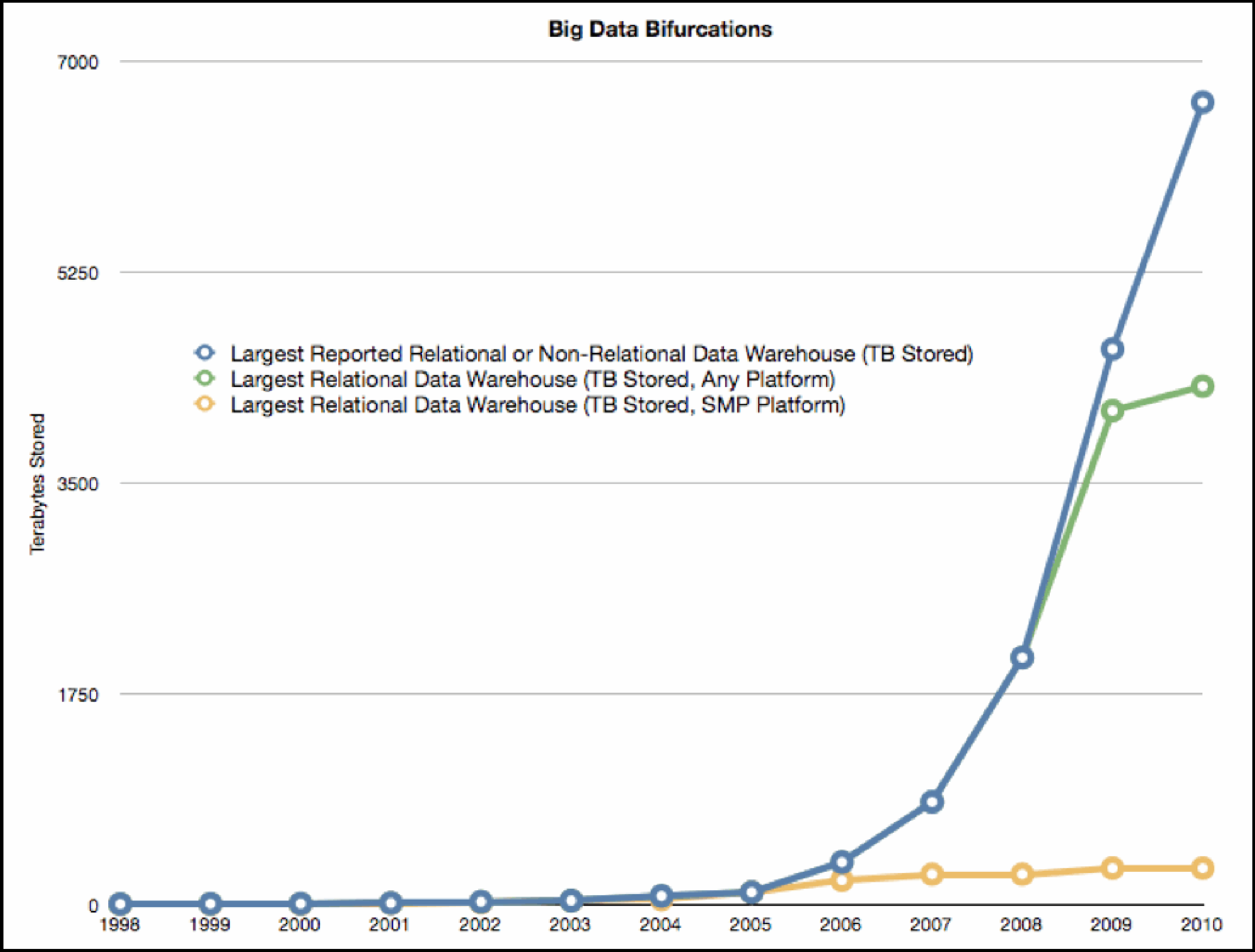
But the nature of the OLTP and OLAP workloads is very different. Forcing them into one platform is almost impossible for data architecture reasons and particularly at scale*

Third Nature

(It’s all being done in the name of)

BIGNESS AND SCALABILITY

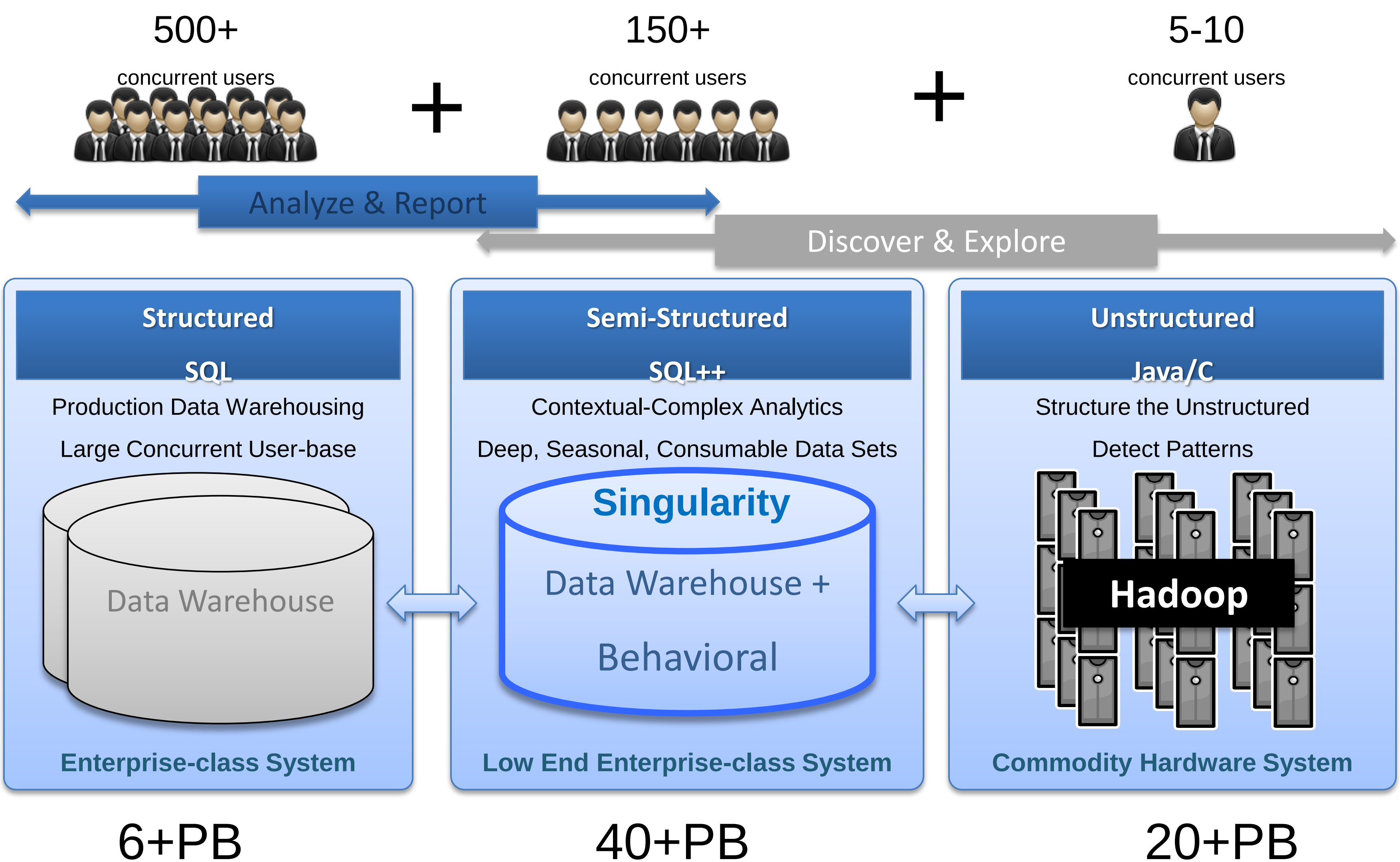
Technology Capability and Data Volume: Solved?



Source: Noumenal, Inc.

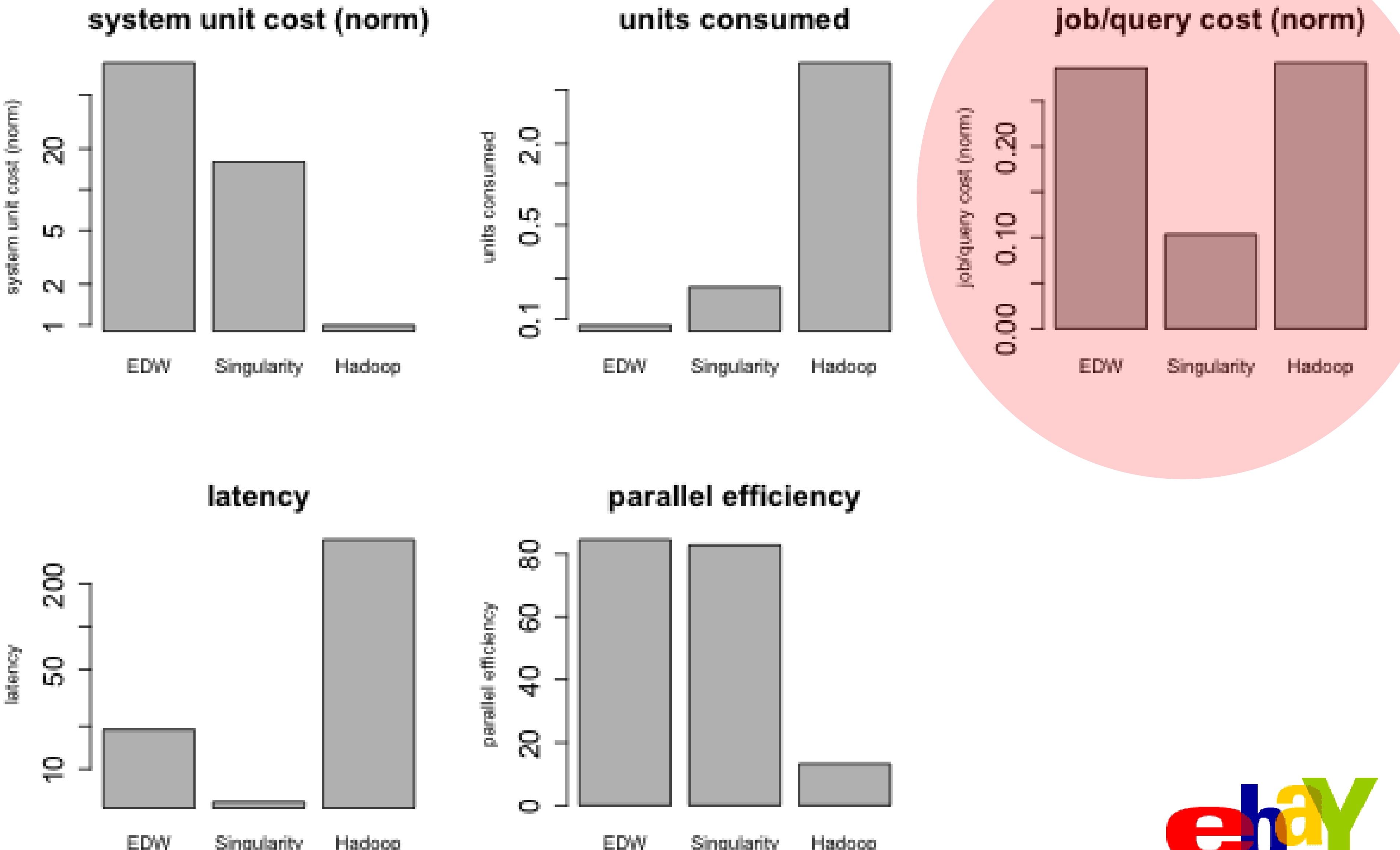
Third Nature

Data Platforms

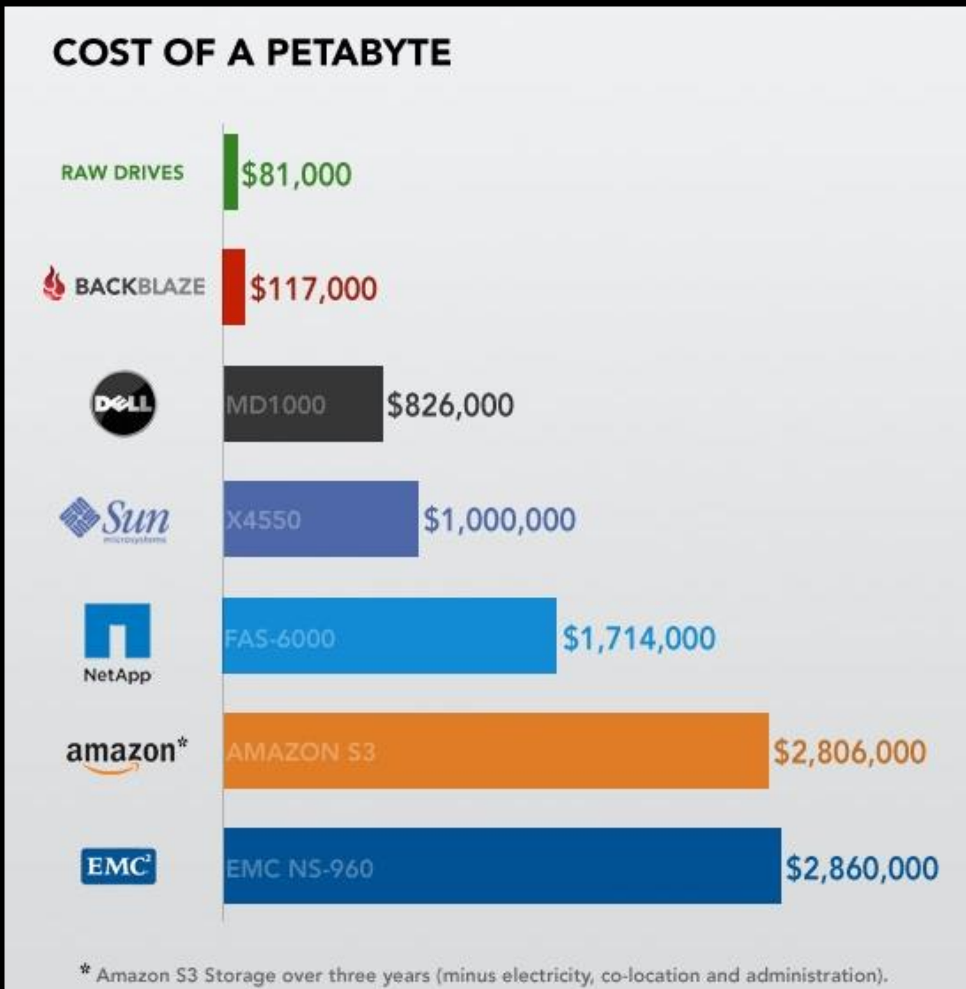


Thanks to eBay for these case slides.

Platform Metrics for Table Scan and Sum, Hadoop vs Teradata



Pricing and performance: Hadoop is a storage and processing play, *not* a database play*



Source: Venturebeat

With big data systems, the cost of storing data is an order of magnitude lower than with databases today (but not the cost or ability to query it back out).

Processing data at scale is at least an order of magnitude cheaper too.

Third Nature

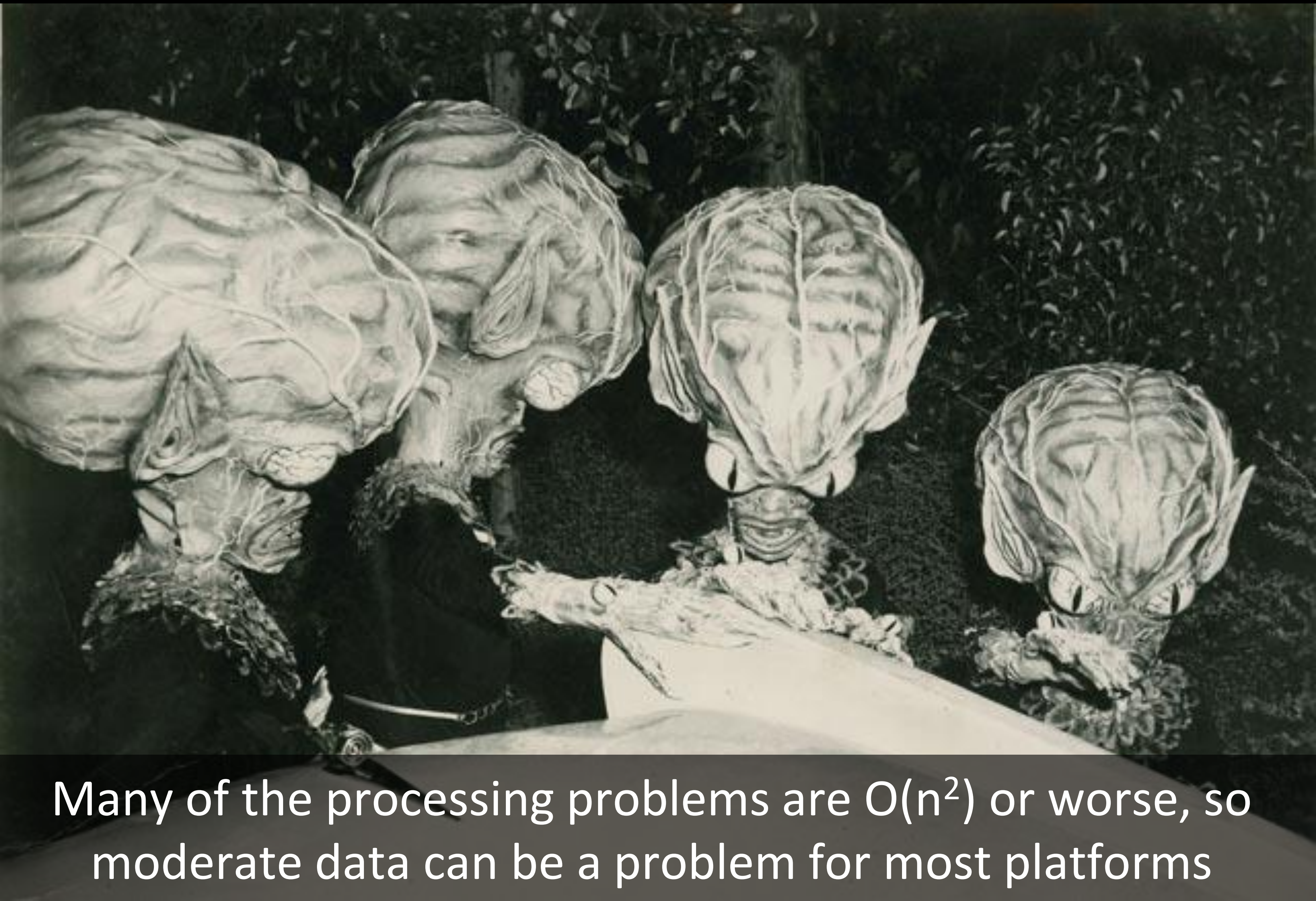
BIGNESS AND DATA COMPUTATIONAL WORKLOADS

Not finished: remember the cycle of history…

The biggest hole in the prior sections is that we scaled OLTP and OLAP but what about analytics?

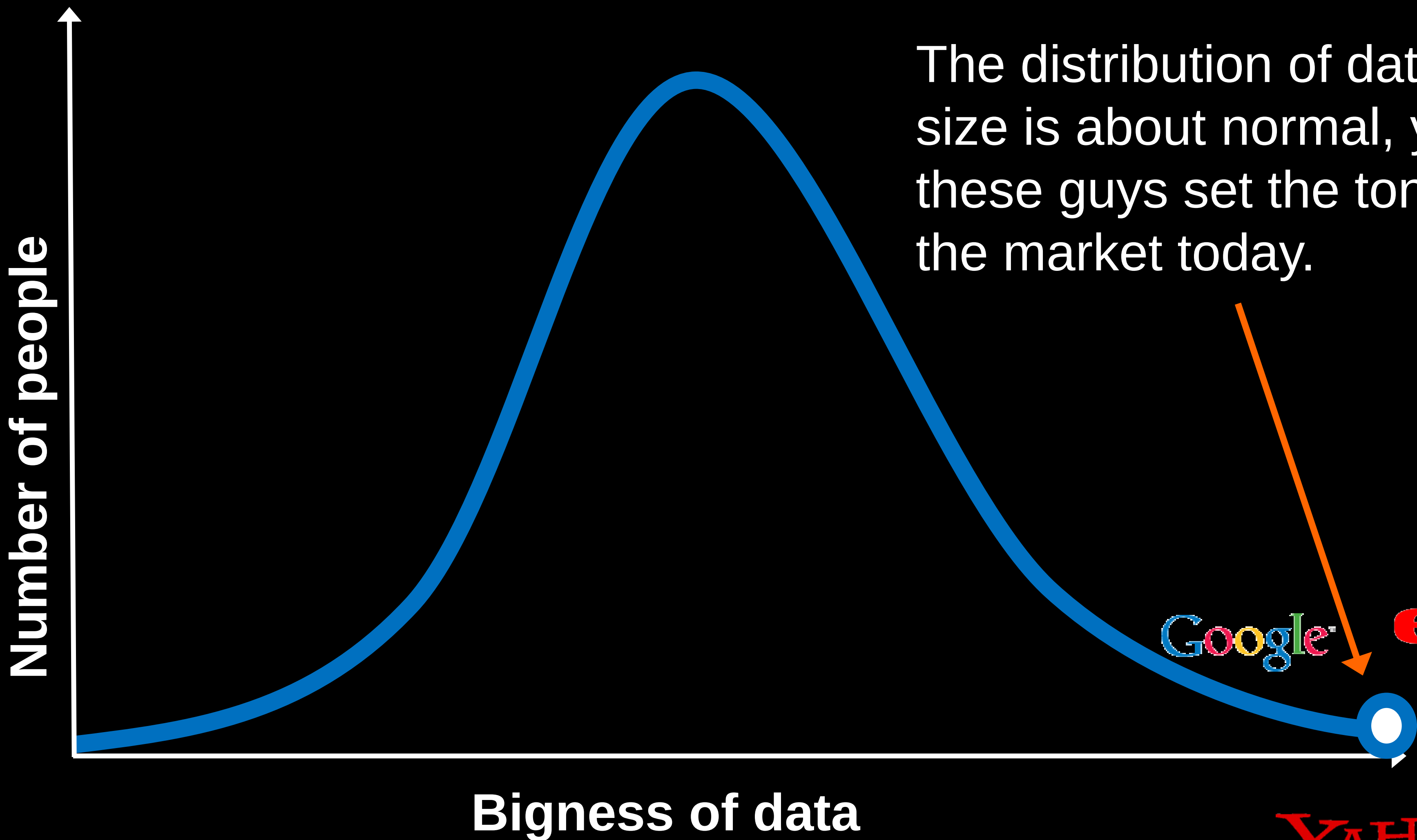
Queries <> transactions <> computations

Analytics embiggens the data volume problem



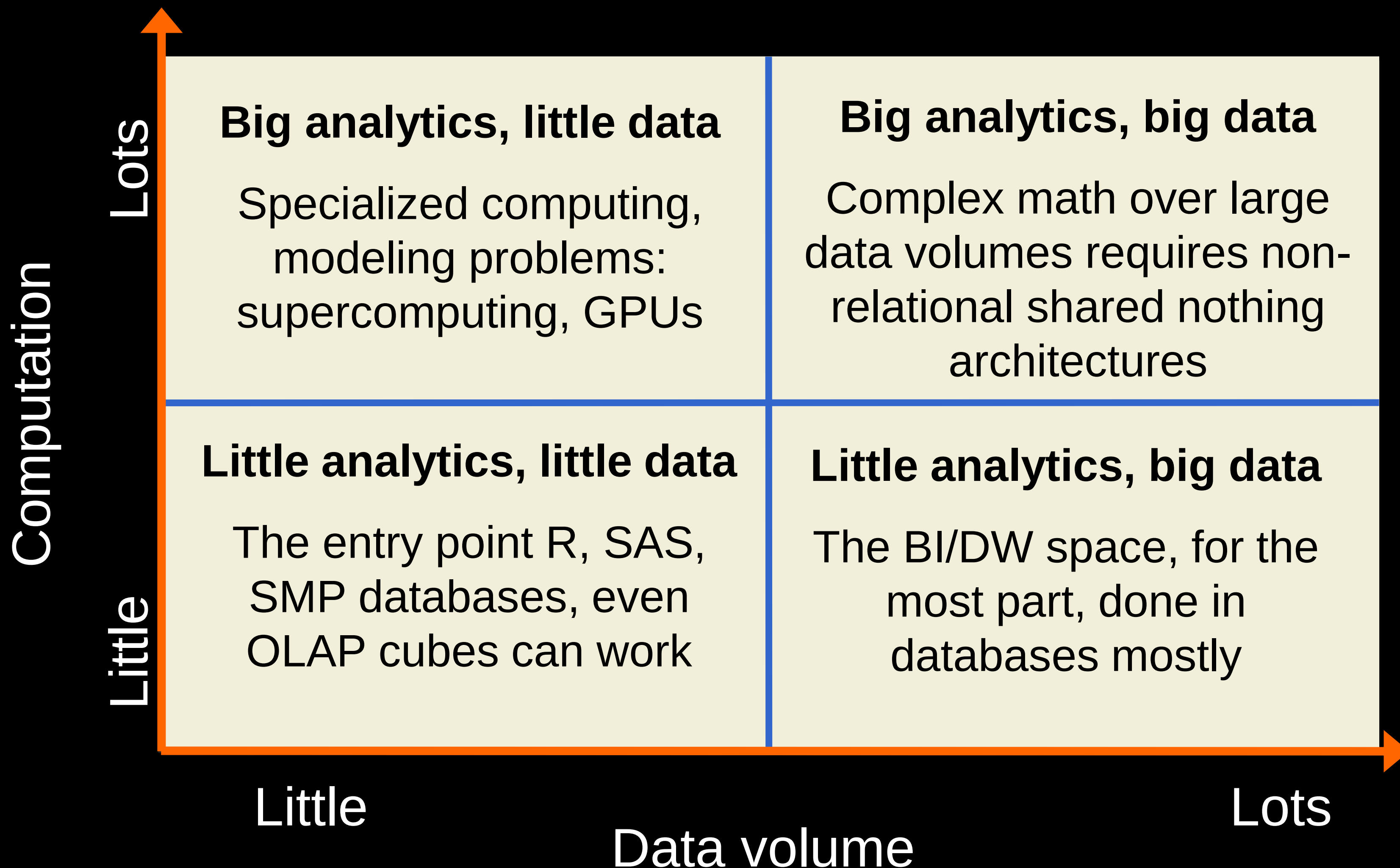
Many of the processing problems are $O(n^2)$ or worse, so moderate data can be a problem for most platforms

Bigness: most people do not need special technology



The distribution of data size is about normal, yet these guys set the tone of the market today.

A Simple Division of the Analytic Problem Space



The three way workload break

1. Operational: OLTP systems
2. Analytic: OLAP systems
3. Scientific: Computational systems

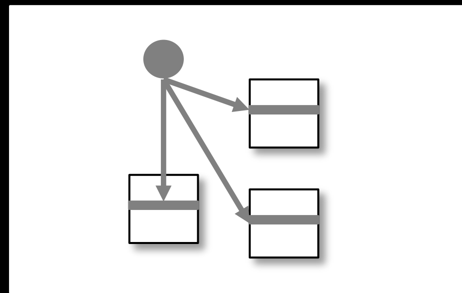
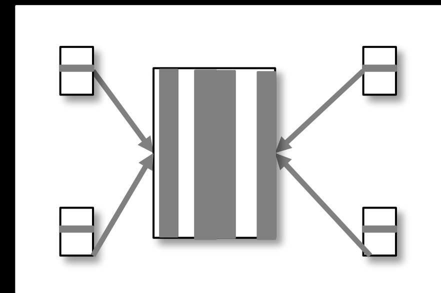
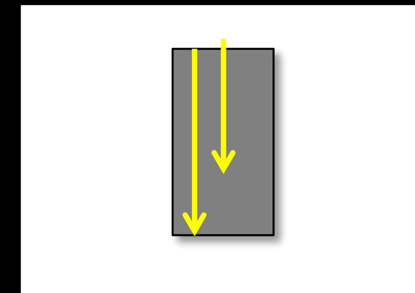
Unit of focus:

1. Transaction
2. Query
3. Computation

Different problems require different platforms

Third Nature

Workloads

	OLTP 	BI 	Analytics 
Access	Read-Write	Read-only	Read-mostly
Predictability	Fixed path	Unpredictable	All data
Selectivity	High	Low	Low
Retrieval	Low	Low	High
Latency	Milliseconds	<seconds	msecs to days
Concurrency	Huge	Moderate	1 to huge
Model	3NF, nested object	Dim, denorm	BWT
Task size	Small	Large	Small to huge

Third Nature

Why would digital data be any different than clay or scrolls or books?

DATA PERSISTENCE AND STORES

Third Nature

“Big data is unprecedented.”

- Anyone involved with big data in even the most barely perceptible way

1960s — 1970s — 1980s — 1990s — 2000s — 2010s

<u>MultiValue, Hierarchical</u>
PICK
IMS
IDS
ADABAS

<u>Relational</u>
CODASYL
System R (SEQUEL)
SQL/DS
INGRES (QUEL)
Mimer
Oracle

<u>RDBMS, SQL standard</u>
DB2
Teradata
Informix
Sybase
Postgres

<u>OODBMS, ORDBMS</u>
Versant
Objectivity
Gemstone
Informix*
Oracle*

<u>MPP Query & NoSQL</u>
Netezza
Paraccel
Vertica
MongoDB
CouchBase
Riak
Cassandra

<u>NewSQL</u>
SciDB
NuoDB
JethroDB
Metanautix

Spanner
F1

Third Nature

NoSQL?

There’s a difference between having no past and actively rejecting it.

Third Nature

A history of databases in No-tation

1970s: NoSQL = We have no SQL

1980s: NoSQL = Know SQL

2000s: NoSQL = No SQL!

2005s: NoSQL = Not only SQL

2013: NoSQL = No, SQL!

(R)DB(MS)

The secret of pre-relational DB: schema

Loose coupling – the physical model of data structures and physical placement are no longer a program's responsibility; data portability ensues.

Reusability – More than one program can access the same data, and no more custom coding for each application or OS

Scalability – Constraints of schema and typing reduce resource usage, have finer granularity for concurrent access, multiple online users.

Third Nature

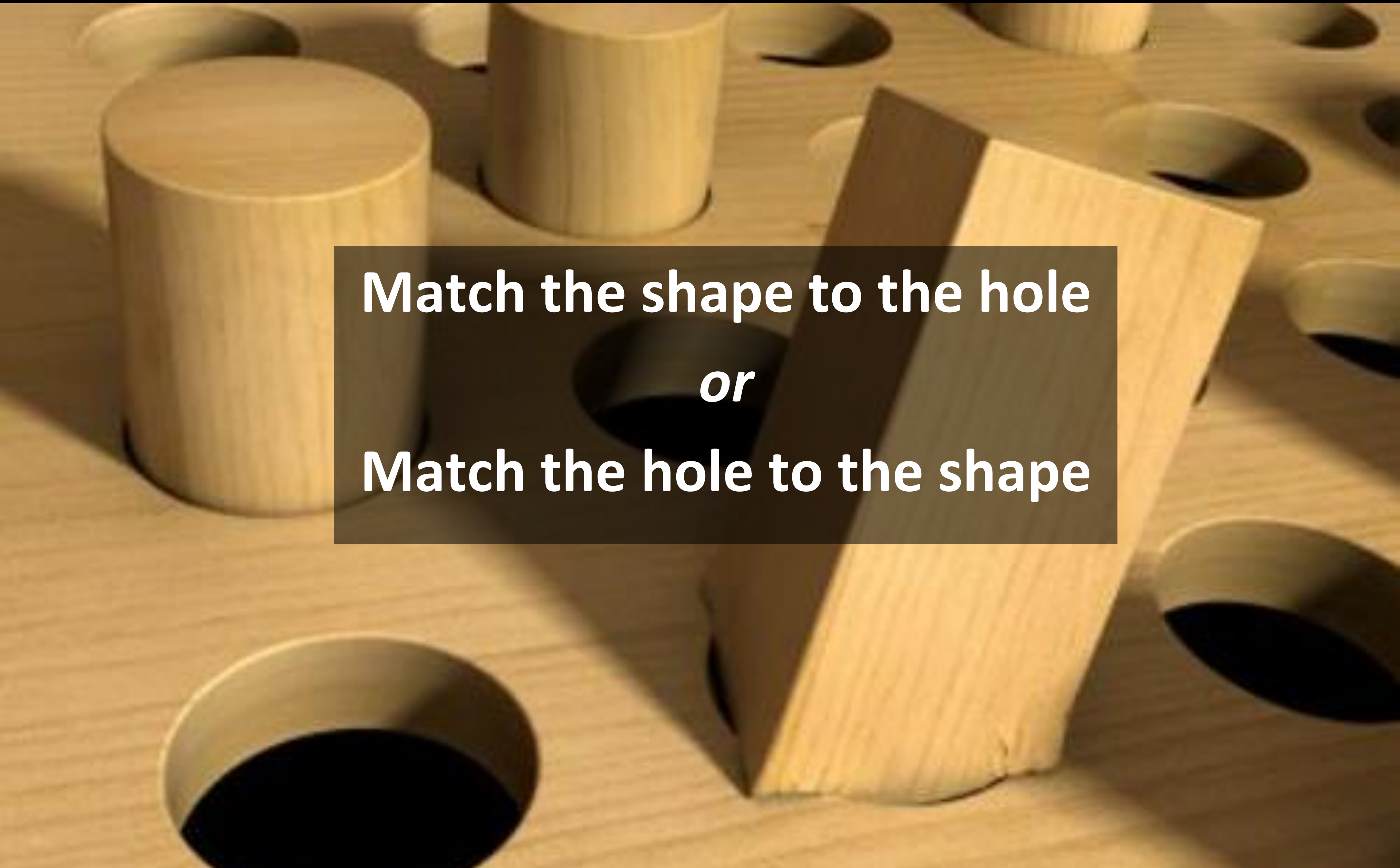
TANSTAAFL

When replacing the old with the new (or ignoring the new over the old) you always make tradeoffs, and usually you won't see them for a long time.

Technologies are not perfect replacements for one another. Often not better, only different.

Third Nature

Schema on write vs schema on read



Match the shape to the hole

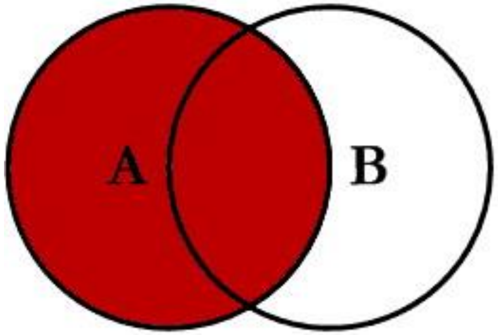
or

Match the hole to the shape

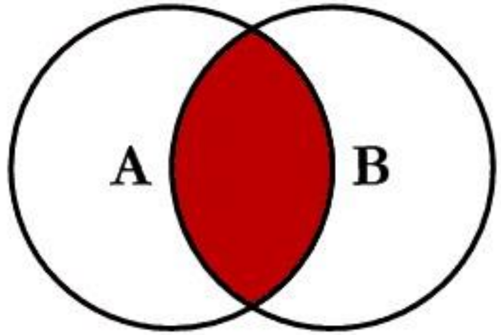
Predicate schemas for write flexibility (agility) and speed

SQL JOINS

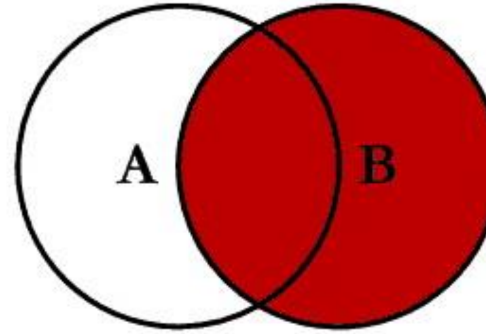
1986: Wait, there's more than one?



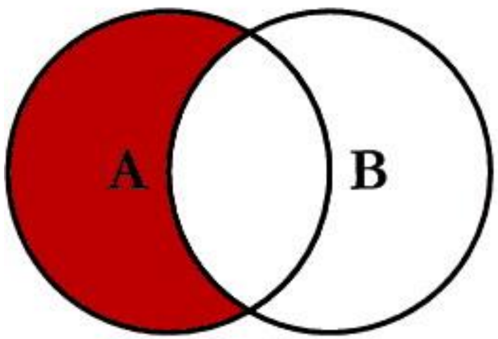
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key



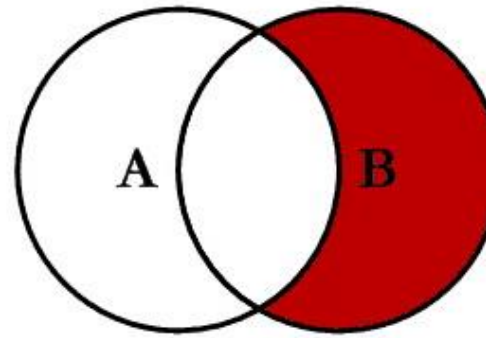
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key



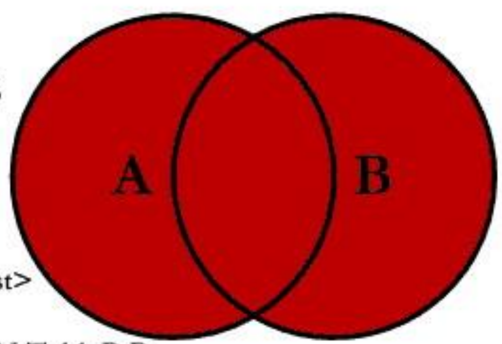
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key



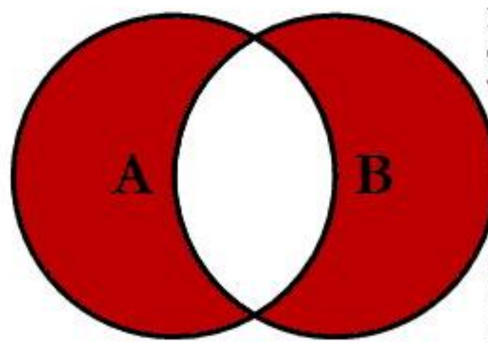
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL



SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL



SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key



SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL

What the optimizer does

It turns a SQL query into an optimal* execution plan for a parallel pipelined dataflow engine

1  Enumerate logically equivalent plans by applying equivalence rules

2  For each logically equivalent plan, enumerate all alternative physical query plans

3  Estimate the cost of each of the alternative physical query plans

4  Run the plan with lowest estimated overall cost

Diagram: David J. DeWitt

Third Nature

A simple 3 table join

```
SELECT C.name, O.num
FROM Orders O, Lines L, Customers C
WHERE C.City = "Copenhagen" AND L.status = "X"
      AND O.num = L.num AND C.cid = O.cid
```

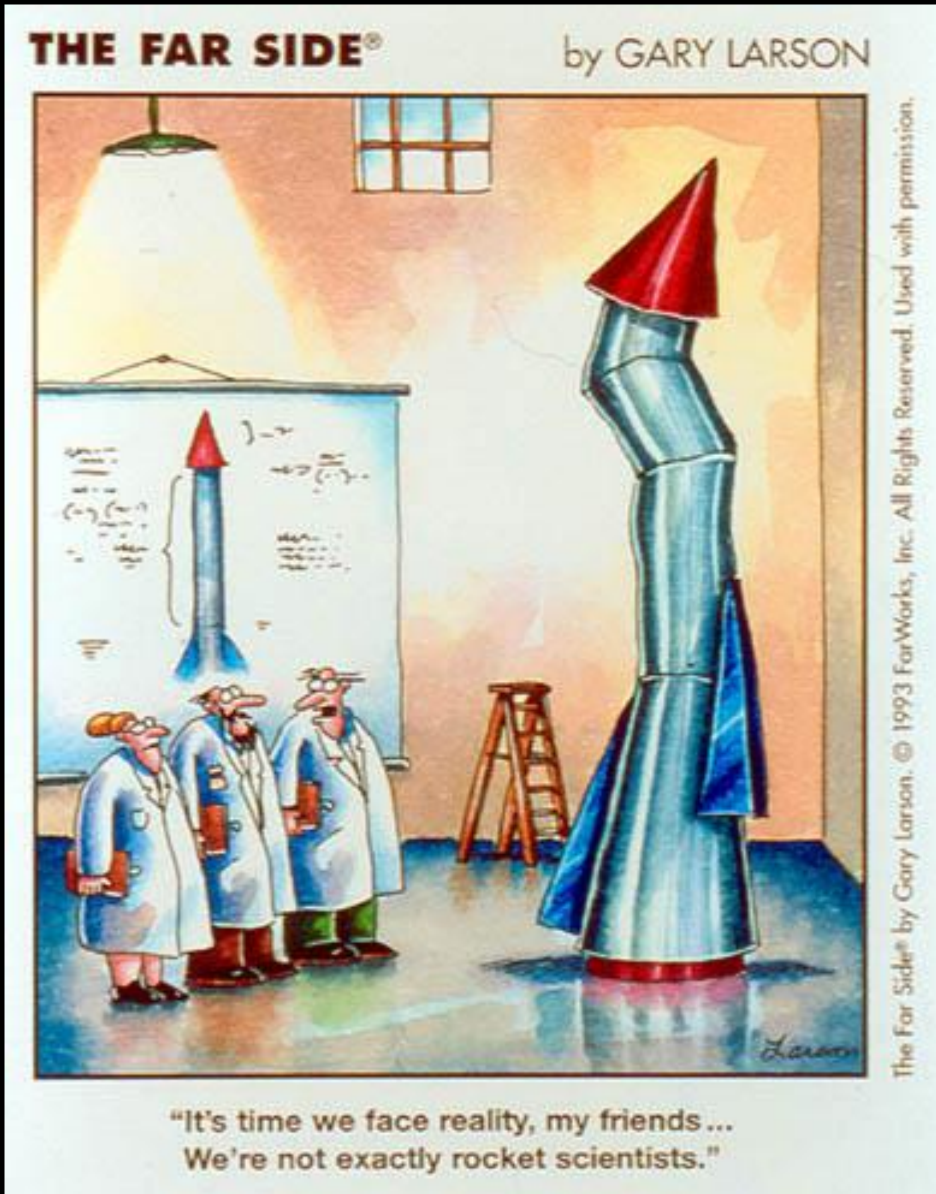
Number of logical plans based on equivalence rules: 9

Ways to join (hash, merge, nested): 3

For each plan, there are multiple physical plans: 36

That makes a total of 324 physical plans, the efficiency of which changes based on cardinality.

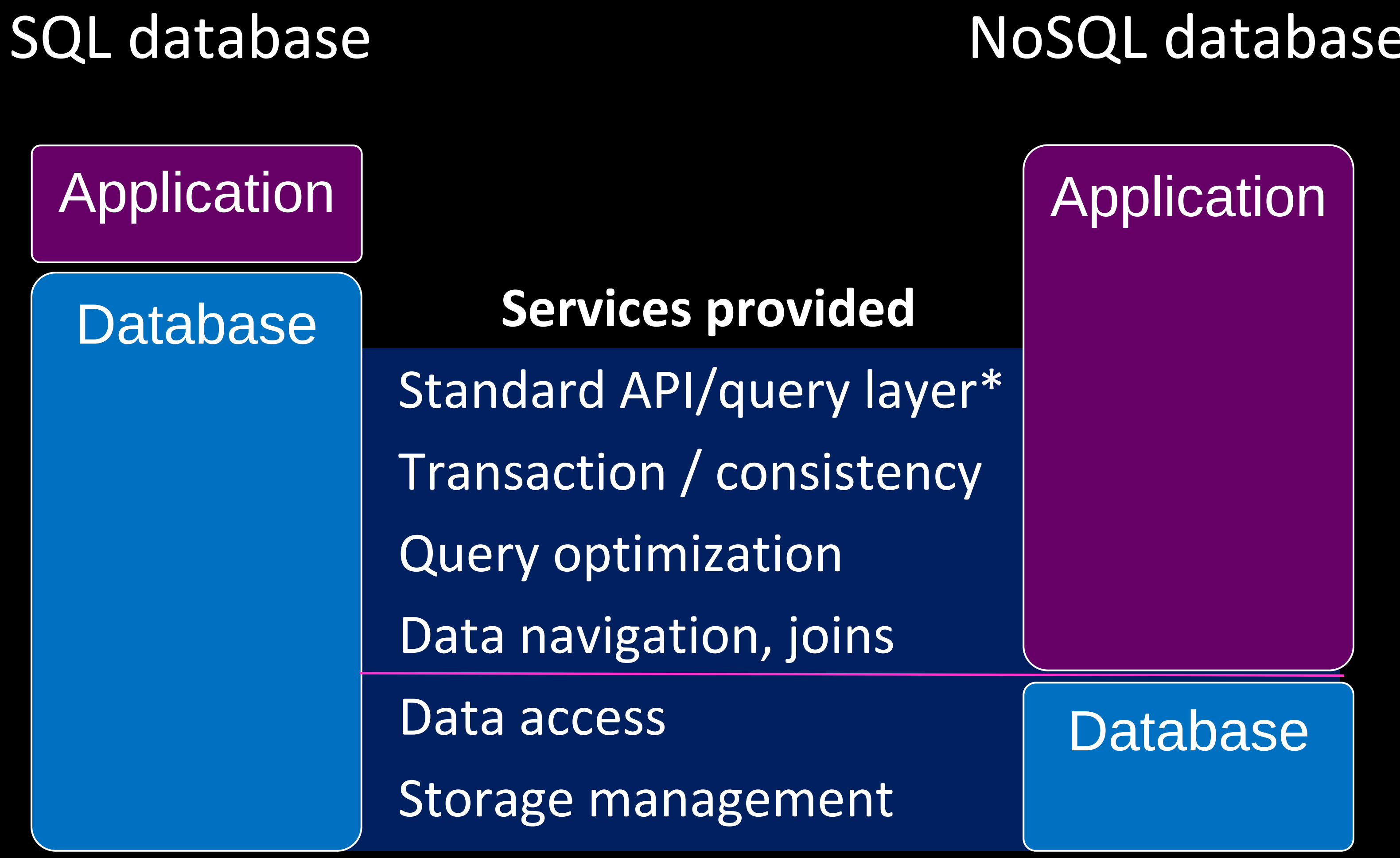
Tradeoffs? In NoSQL Land, Optimizer is You!



"Query optimization is not rocket science. When you flunk out of query optimization, we make you go build rockets."

Third Nature

Tradeoffs: In NoSQL the DBMS is in your code



Anything not done by the DB becomes a developer’s task.

Simplifying ACID vs BASE

Eventually consistent is a nice way of saying "not correct"

"transaction malleability" is a nice way of saying "broken"

Remember: it's a poor carpenter who blames his tools.

Maybe…


These aren’t the databases we’re looking for.
We need one that speaks pig latin.

Google on eventual consistency:

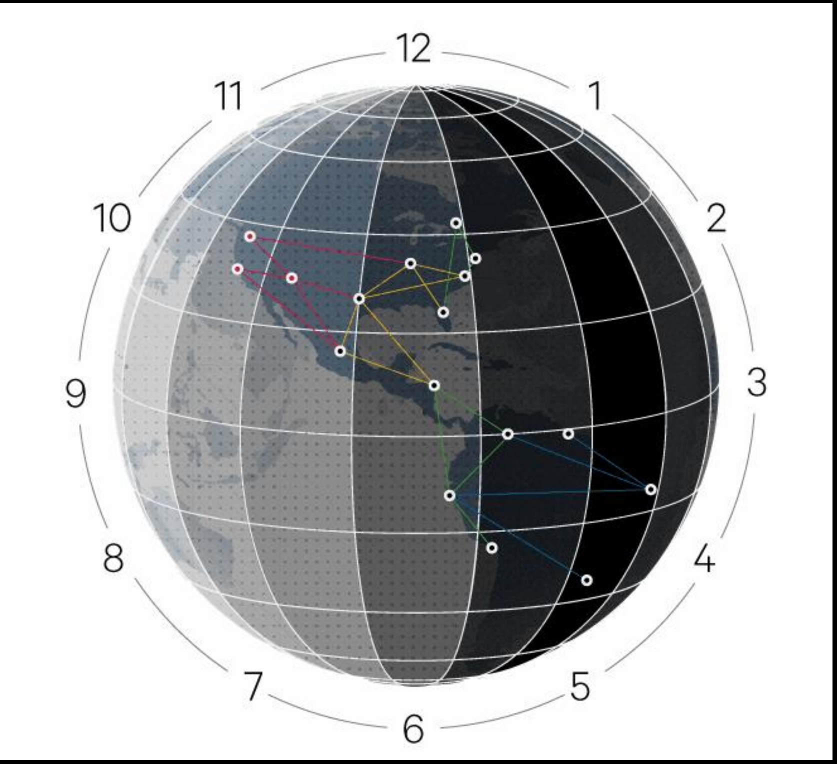
Designing applications to cope with concurrency anomalies in their data is very error-prone, time-consuming, and ultimately not worth the performance gains.

developers spend a significant fraction of their time building extremely complex and error-prone mechanisms to cope with eventual consistency and handle data that may be out of date. We think this is an unacceptable burden to place on developers and that consistency problems should be solved at the database level. Full transactional consistency is one

"F1: A Distributed SQL Database That Scales", Proceedings of the VLDB Endowment, Vol. 6, No. 11, 2013

Third Nature

Google F1: Another Evolution



- Distributed SQL database
- ACID compliance, 2PC and row-level locking (!)
- Transparent data distribution
- Synchronous replication across data centers
- Table interleaving (hierarchies)
- Queryable protobufs
- MapReduce access to underlying data
- Average user-facing latency of ~200ms with small deviation

Third Nature

The holy grail of databases under current market hype

We're talking mostly about computation over data when we talk about "big data" and analytics.

The goal is combining data storage, retrieval and analysis into one system, *a potential mismatch for both relational and nosql.*

Hadoop & NoSQL Adoption

Some people can't resist getting the next new thing because it's new.

Many organizations are like this, promoting a solution and hunting for the problem that matches it.

Better to ask "What is the problem for which this technology is the answer?"

Third Nature

Unintended consequences

Disruption vs Destabilization

Disruption is a bad framing for most software.

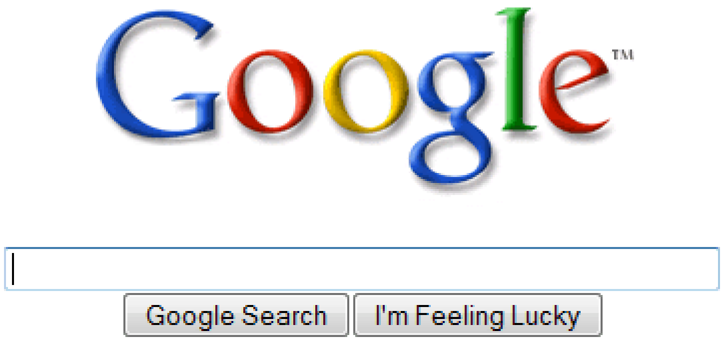
New technology exists in a software ecosystem with many dependencies.

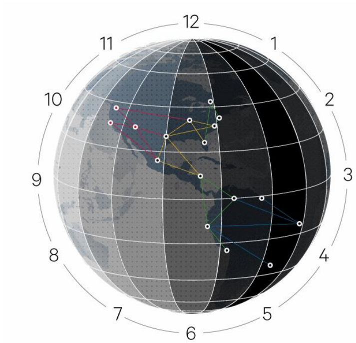
Better to frame new technology as destabilizing. This does not imply direct replacement.

The big data revolution, more of an evolution

Be pragmatic, not dogmatic

Who are you following?





hadoop

And why are you following them?

Third Nature

Conclusion


IF YOU
PROCRASTINATE
LONG ENOUGH
MOST PROBLEMS
SOLVE THEMSELVES
CRACKED.com

Summary

1. All design decisions are tradeoffs. Be aware of what you trade away for the thing you get.
2. Pay attention to workloads and try to isolate them when you can. Mixed workloads are very hard.
3. Relational algebra is still useful. The key is distributed parallel database implementations.
4. Declarative models enable optimizers. Optimizers save you work.

References (things worth reading on the way home)

A relational model for large shared data banks, Communications of the ACM, June, 1970, http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf

Column-Oriented Database Systems, Stavros Harizopoulos, Daniel Abadi, Peter Boncz, VLDB 2009 Tutorial http://cs-www.cs.yale.edu/homes/dna/talks/Column_Store_Tutorial_VLDB09.pdf

Nobody ever got fired for using Hadoop on a cluster, 1st International Workshop on Hot Topics in Cloud Data ProcessingApril 10, 2012, Bern, Switzerland.

A co-Relational Model of Data for Large Shared Data Banks, ACM Queue, 2012, http://queue.acm.org/detail.cfm?id=1961297

A query language for multidimensional arrays: design, implementation and optimization techniques, SIGMOD, 1996

Probabilistically Bounded Staleness for Practical Partial Quorums, Proceedings of the VLDB Endowment, Vol. 5, No. 8, http://vldb.org/pvldb/vol5/p776_peterbailis_vldb2012.pdf

"Amorphous Data-parallelism in Irregular Algorithms", Keshav Pingali et al

MapReduce: Simplified Data Processing on Large Clusters, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//archive/mapreduce-osdi04.pdf

Dremel: Interactive Analysis of Web-Scale Datasets, Proceedings of the VLDB Endowment, Vol. 3, No. 1, 2010 http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//pubs/archive/36632.pdf

Spanner: Google's Globally-Distributed Database, SIGMOD, May, 2012, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/es//archive/spanner-osdi2012.pdf

F1: A Distributed SQL Database That Scales, Proceedings of the VLDB Endowment, Vol. 6, No. 11, 2013, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/41344.pdf

Third Nature

CC Image Attributions

Thanks to the people who supplied the creative commons licensed images used in this presentation:

shady_puppy_sales.jpg - http://www.flickr.com/photos/brizzlebornandbred/5001120150

cuneiform_proto_3000bc.jpg - http://www.flickr.com/photos/takomabibelot/3124619443/

cuneiform_undo.jpg - http://www.flickr.com/photos/charlestilford/2552654321/

scroll_kerouac.jpg - http://www.flickr.com/photos/ari/93966538/

House on fire - http://flickr.com/photos/oldonliner/1485881035/

Manuscripts on shelf - http://flickr.com/photos/peterkaminski/1688635/

manuscript_illum.jpg - http://www.flickr.com/photos/diorama_sky/2975796332/

manuscript_page.jpg - http://www.flickr.com/photos/calliope/306564541/

subway dc metro - http://flickr.com/photos/musaeum/509899161/

Third Nature

About the Presenter

Mark Madsen is president of Third Nature, a technology research and consulting firm focused on analytics, business intelligence and data management. Mark is an award-winning author, architect and CTO whose work has been featured in numerous industry publications. Over the past ten years Mark received awards for his work from the American Productivity & Quality Center, TDWI, and the Smithsonian Institute. He is an international speaker, a contributor to Forbes Online and on the O'Reilly Strata program committee. For more information or to contact Mark, follow @markmadsen on Twitter or visit http://ThirdNature.net

Third Nature

About Third Nature

Third Nature is a research and consulting firm focused on new and emerging technology and practices in business intelligence, analytics and performance management. If your question is related to BI, analytics, information strategy and data then you‘re at the right place.

Our goal is to help companies take advantage of information-driven management practices and applications. We offer education, consulting and research services to support business and IT organizations as well as technology vendors.

We fill the gap between what the industry analyst firms cover and what IT needs. We specialize in product and technology analysis, so we look at emerging technologies and markets, evaluating technology and hw it is applied rather than vendor market positions.