DEPARTMENT OF
COMPUTER SCIENCE

THE NEW AGE OF DISCOVERY

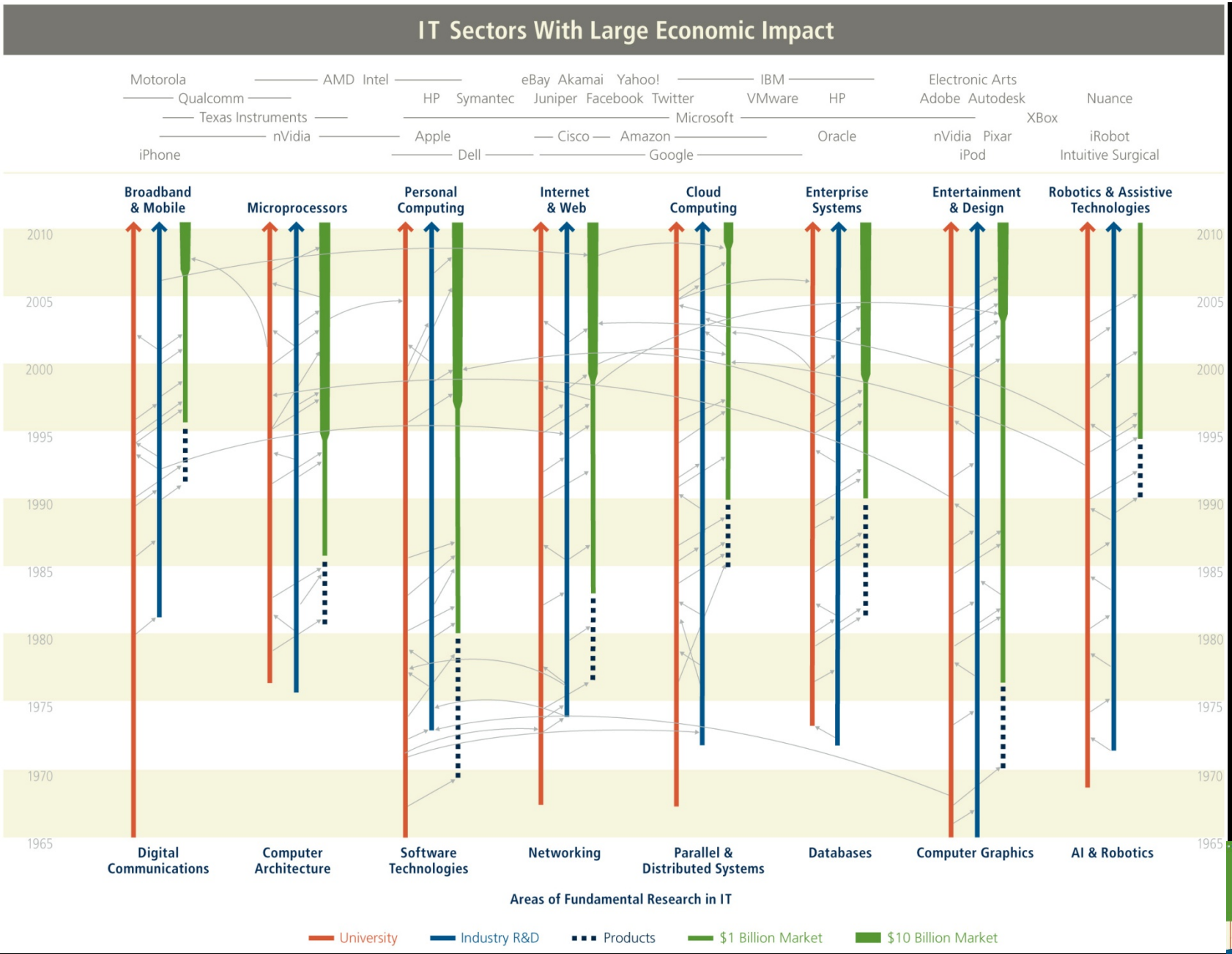
What Is Computing Today? Deconstruction

Gregory D. Hager
Professor and Chair

JOHNS HOPKINS
UNIVERSITY
WHITING SCHOOL OF ENGINEERING

Questions

- How does “stuff work” and how did it come to be?
- What are the basic research areas of CS that impacted it?
- What are commercial needs drove it?
- How has that “stuff” changed with time?



2012 NITRD update by PCAST

Some Background Information

- IT is around 1T$* of US economy (itself 18T$ GDP)
  - **Apple Inc. (Nasdaq: AAPL), (560B/30B)**
  - Exxon Mobil Corporation (NYSE: XOM),
  - **Google Inc (Nasdaq: GOOG), (358B /12B)**
  - **Microsoft Corporation (Nasdaq: MSFT), (344B/20B)**
  - Berkshire Hathaway Inc. (NYSE: BRK.B),
  - Wal-Mart Stores, Inc. (NYSE: WMT),
  - Johnson & Johnson (NYSE: JNJ),
  - General Electric Company (NYSE: GE),
  - Chevron Corporation (NYSE: CVX)
  - Wells Fargo & Co (NYSE: WFC)

*Atkinson, R. D., & Stewart, L. A. (2013). Just the FACTS: The Economic Benefits of Information and Communications Technologies

Deconstructing a Search Query

Challenges in Building Large-Scale Information Retrieval Systems

Jeff Dean
Google Fellow
jeff@google.com

Credits to material used from
static.googleusercontent.com/media/research.google.com/en/us/people/jeff/WSDM09-keynote.pdf

The Origins of PageRank

- Stanford WebBase project (1996 - 1999)
  http://dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/
  http://dbpubs.stanford.edu:8091/diglib/

- funded by NSF through DLI1
  http://www.dli2.nsf.gov/dlione/

"The Initiative's focus is to dramatically advance the means to collect, store, and organize information in digital forms, and make it available for searching, retrieval, and processing via communication networks -- all in user-friendly ways." quote from the DLI1 website

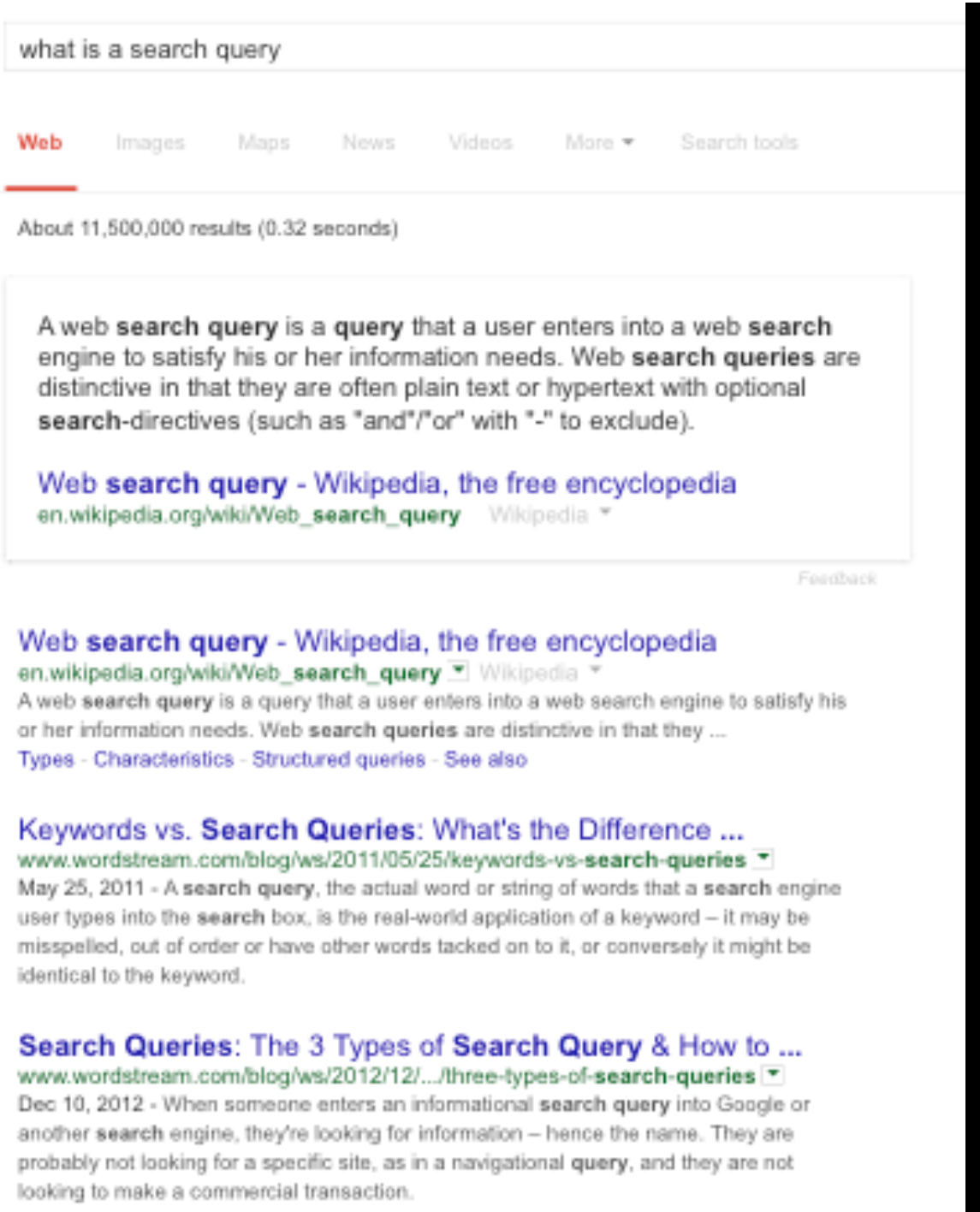
Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. (1999).
Brin, Sergey, and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems* 30, no. 1 (1998): 107-117.

Some Other Research Ideas

- Cache (M. Wilkes, 1965, Cambridge)
- The internet (Cerf, Kahn, 1969, ARPA)
- The Web and HTML (T. Berners-Lee, 1989, CERN)
- PageRank (Brin, Page, Motwani, Winograd, Stanford, 1997)
- SIFT Image Features (Lowe, UBC, 1999)
- Hadoop (Cutting, Cafarella, Yahoo/UW, 2005)
- Deep Learning (Hinton+others, Toronto+ others, ??)
  - GPUs ….

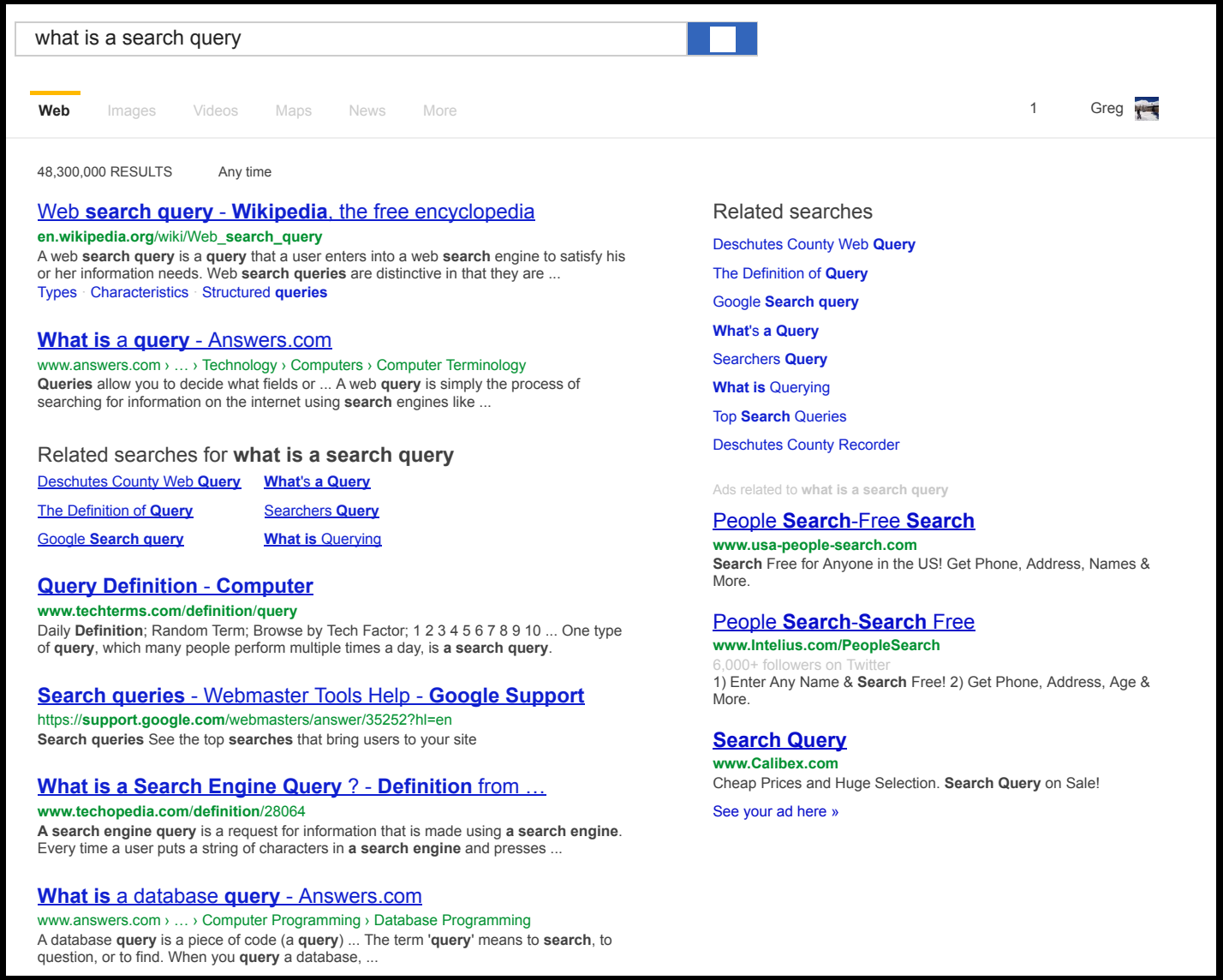
What Is a Search Query?

Google's answer



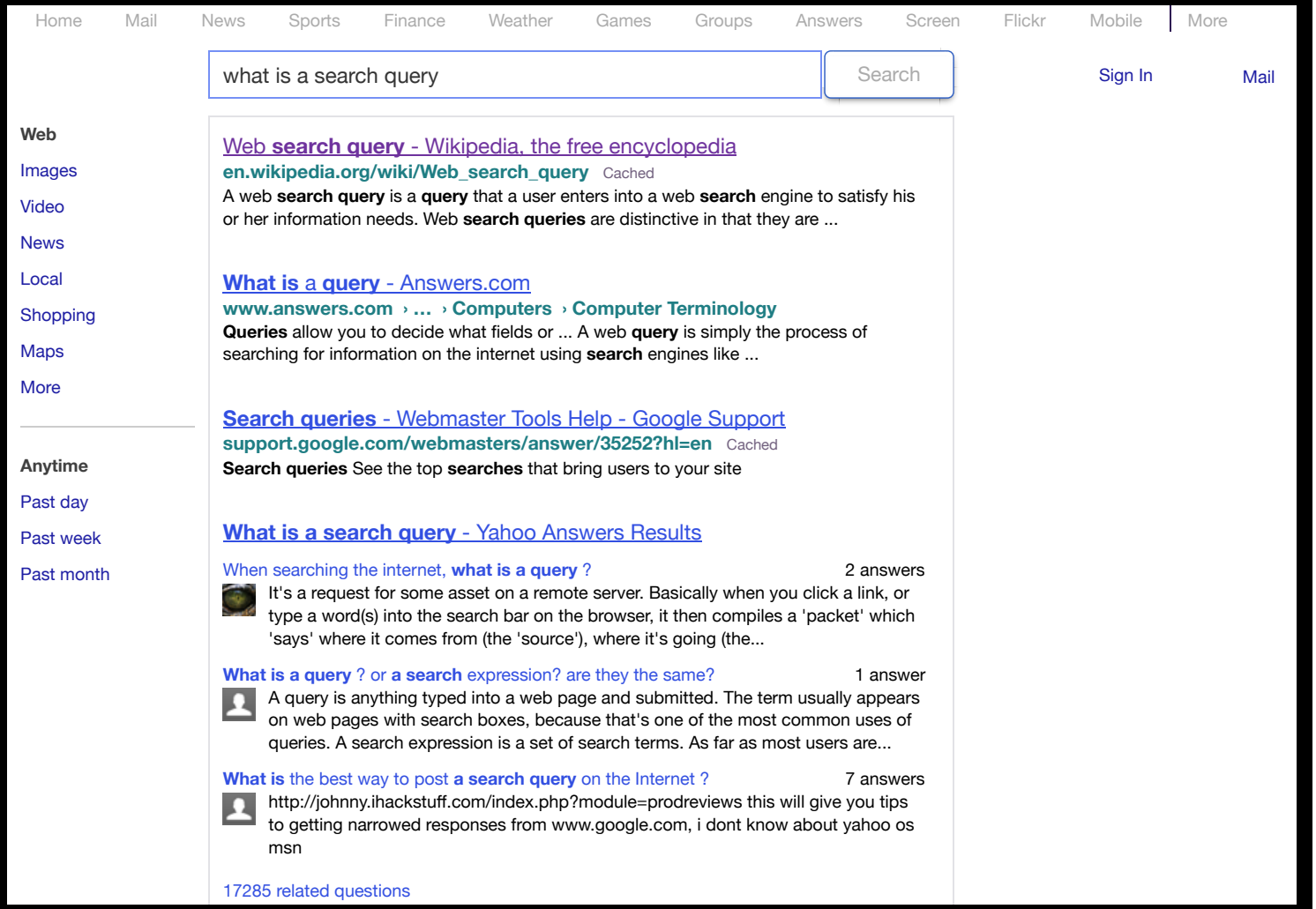
What Is a Search Query?

Bing's answer

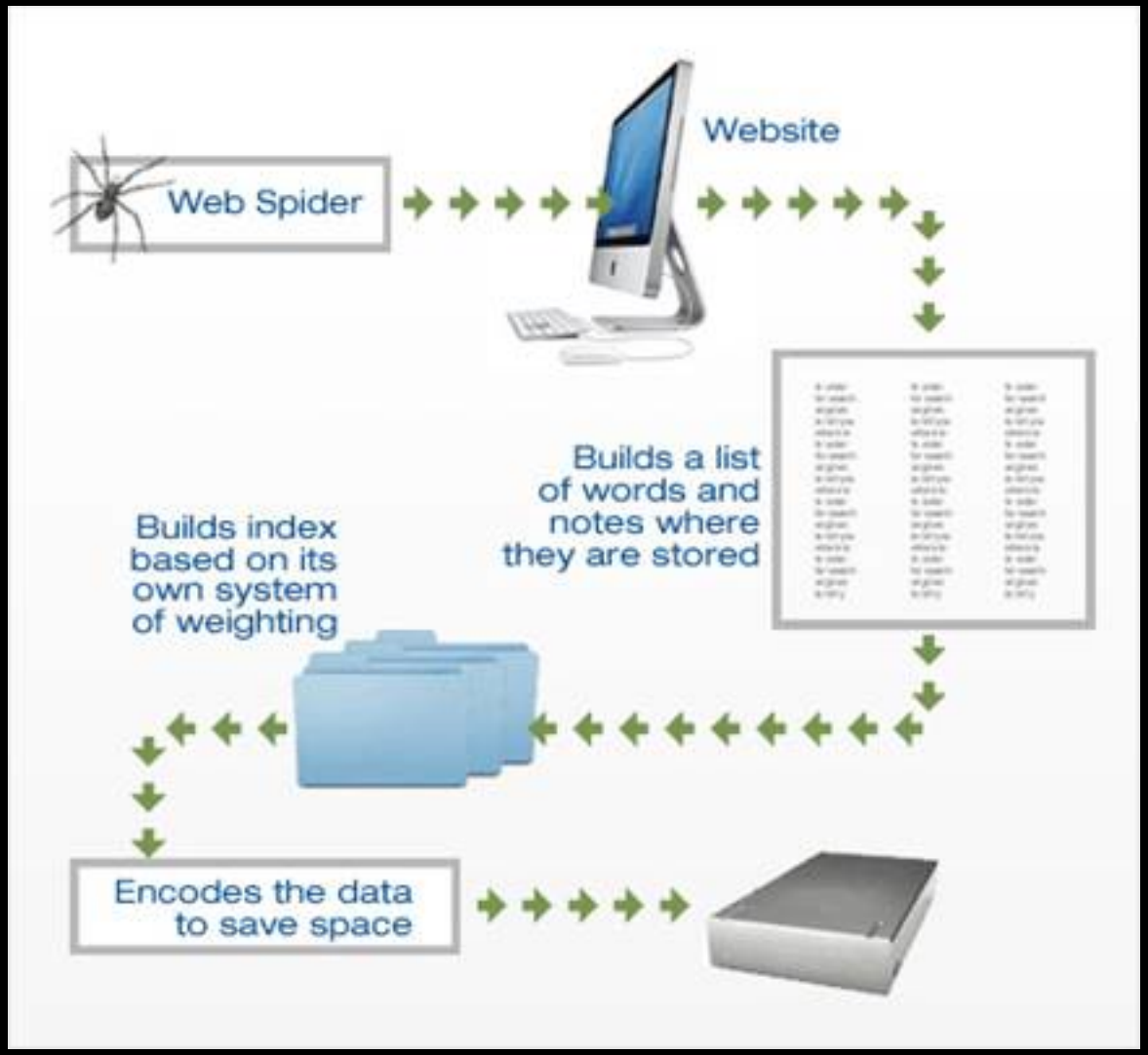


What Is a Search Query?

Yahoo's answer



It All Starts With a Spider



http://programming4.us/website/15366.aspx

Inverted Index

For example, let's say we have two documents, each with a `content` field containing:

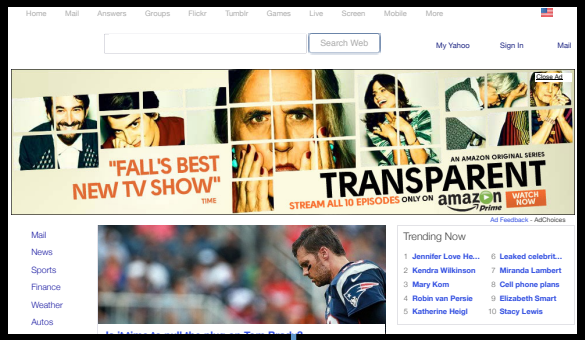
1. "The quick brown fox jumped over the lazy dog"
2. "Quick brown foxes leap over lazy dogs in summer"

To create an inverted index, we first split the `content` field of each document into separate words (which we call *terms* or *tokens*), create a sorted list of all the unique terms, then list in which document each term appears. The result looks something like this:

```
Term      Doc_1  Doc_2
-------------------------
Quick   |       |  X
The     |   X   |
brown   |   X   |  X
dog     |   X   |
dogs    |       |  X
fox     |   X   |
foxes   |       |  X
in      |       |  X
jumped  |   X   |
lazy    |   X   |  X
leap    |       |  X
over    |   X   |  X
quick   |   X   |
summer  |       |  X
the     |   X   |
-------------------------
```


http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/inverted-index.html

Browser to Computer to Internet to Search Engine to Cloud



Then We Need Horsepower …

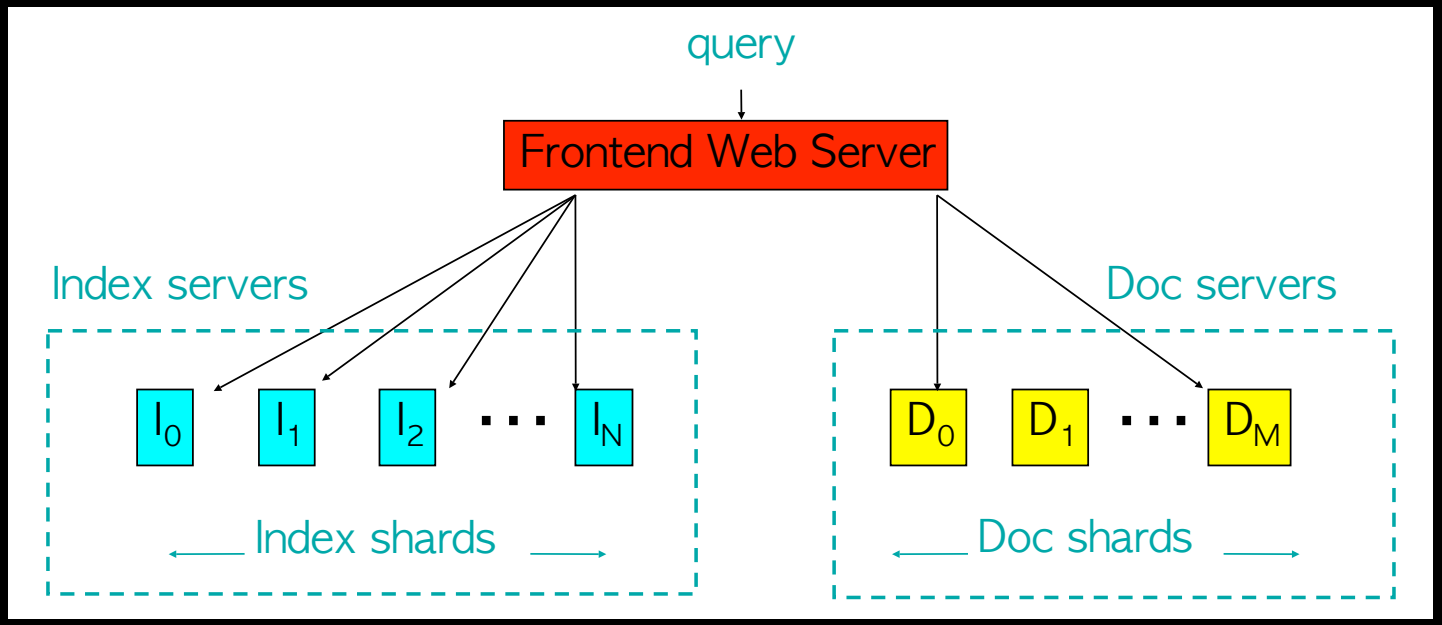


Image from Dean, Google 1997

- Given search terms e.g. dog, cat
- Return pair <docid, score>
- Score is the "secret sauce" for ranking docs
- Doc servers return pre-formatted snippets plus doc address

Some Questions

- Where are the bottlenecks?
- What is missing?

Adding Speed and Income

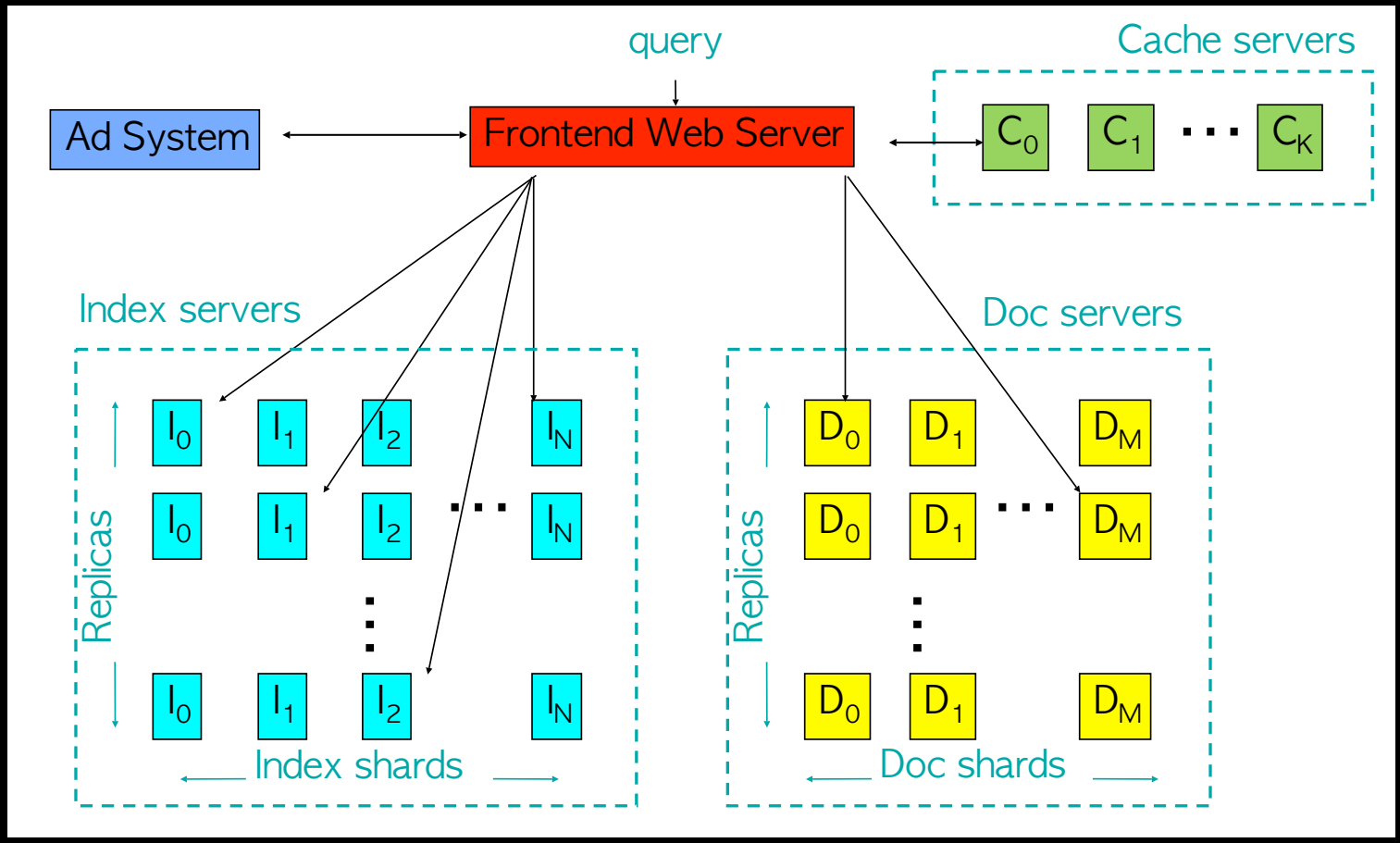


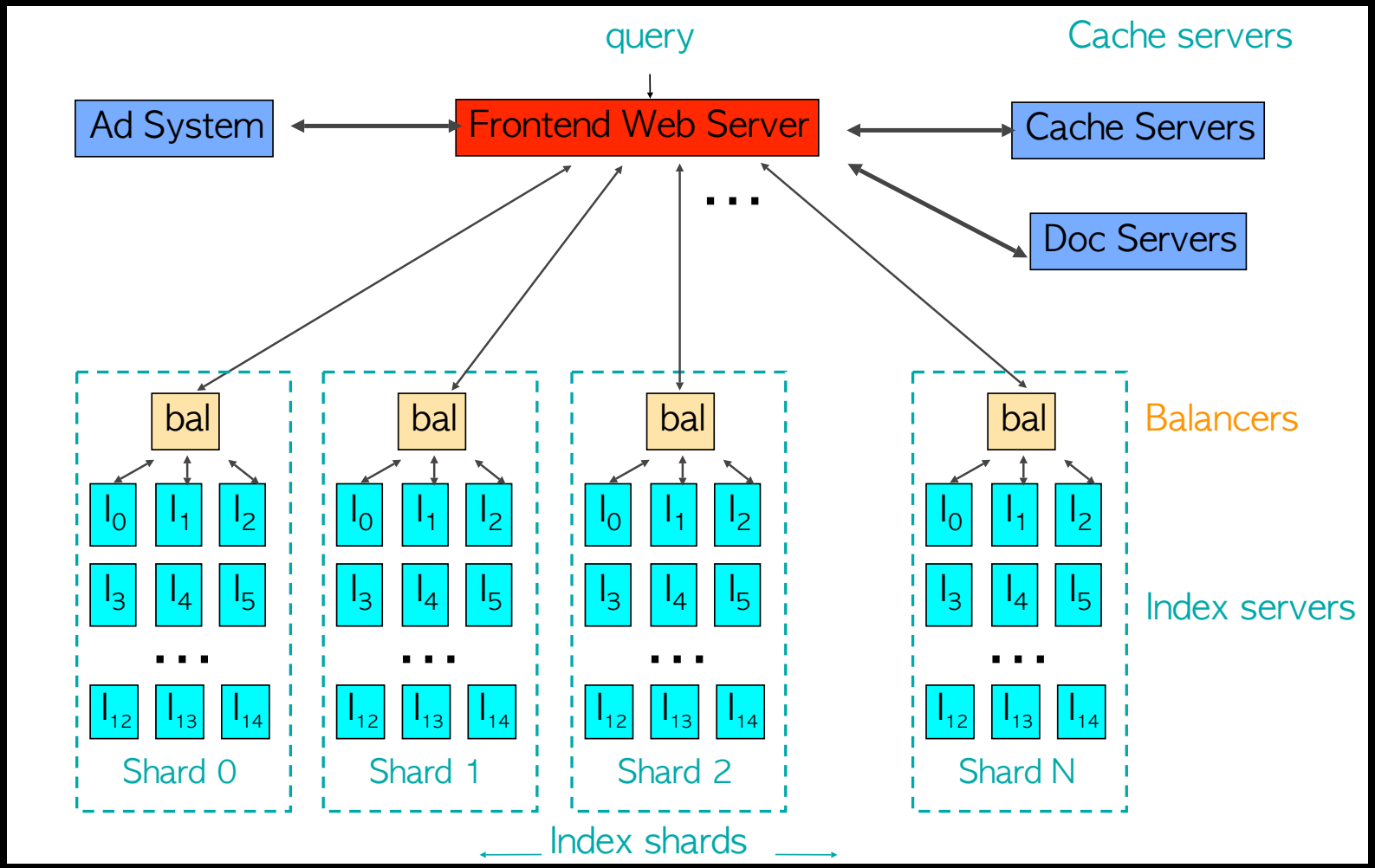
Image from Dean, Google 1999

What Makes an Effective …

- Search?

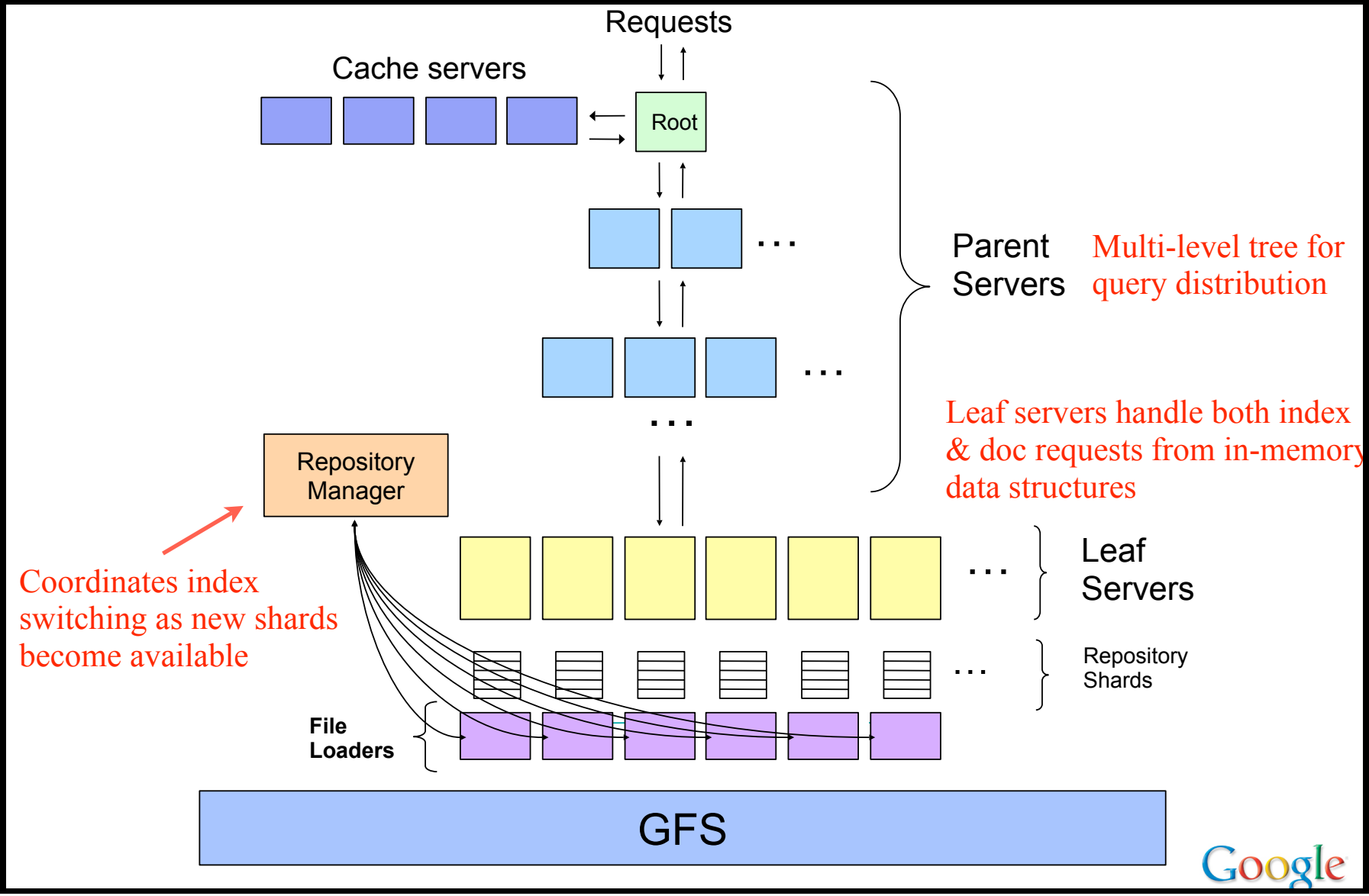
- Ad?

More Scale?



Dean, Google 2001

A More Complete Picture



Some Interesting Plusses and Minuses

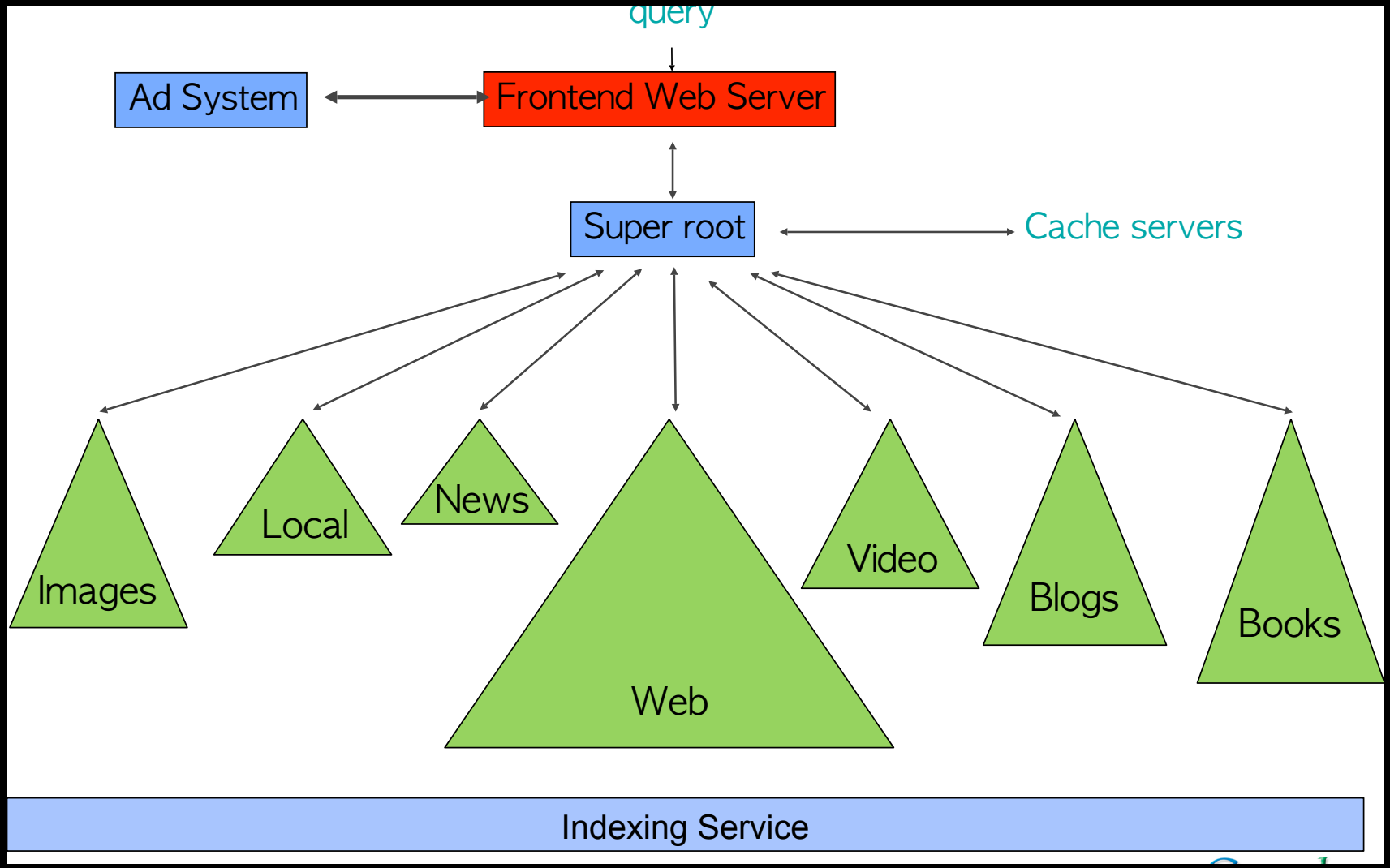
- Queries are now fast – particularly at tail
- Throughput is high

- Now depending on a very large # of machines – one key machine down and that query fails or takes a long time
- What if a query kills a machine – you can mow down the entire cluster

Canary Requests

- Send a request to one machine and see if it dies
- If not, go ahead
- If it does, try a couple more; if they die, give up

Google 2007 Architecture



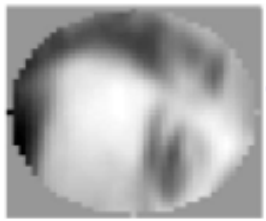
An Aside – Visual Search

- Key breakthrough due to Lowe (SIFT, 1999)
- Second key technology: use of weak labels from the Web
- Third key technology: learning technologies that can be applied at scale

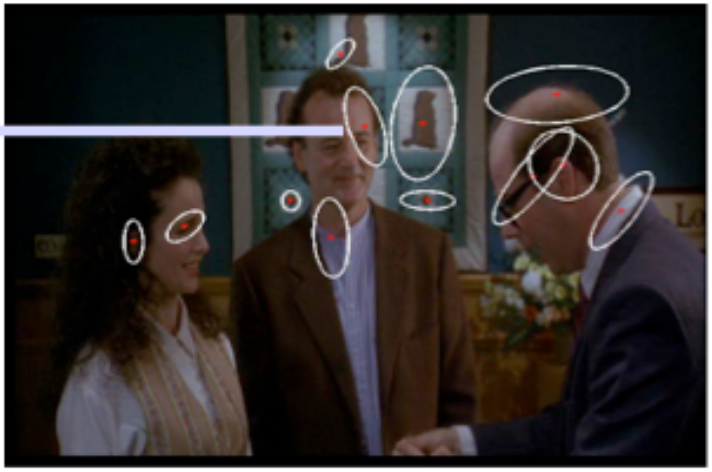
1.Feature detection and representation

Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

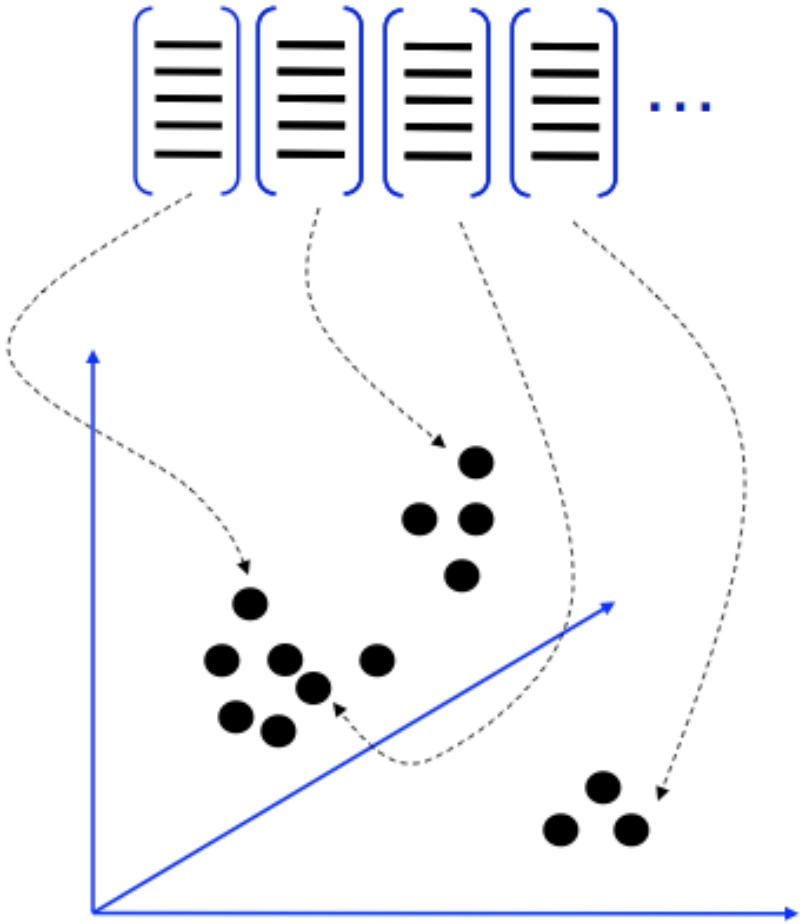
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

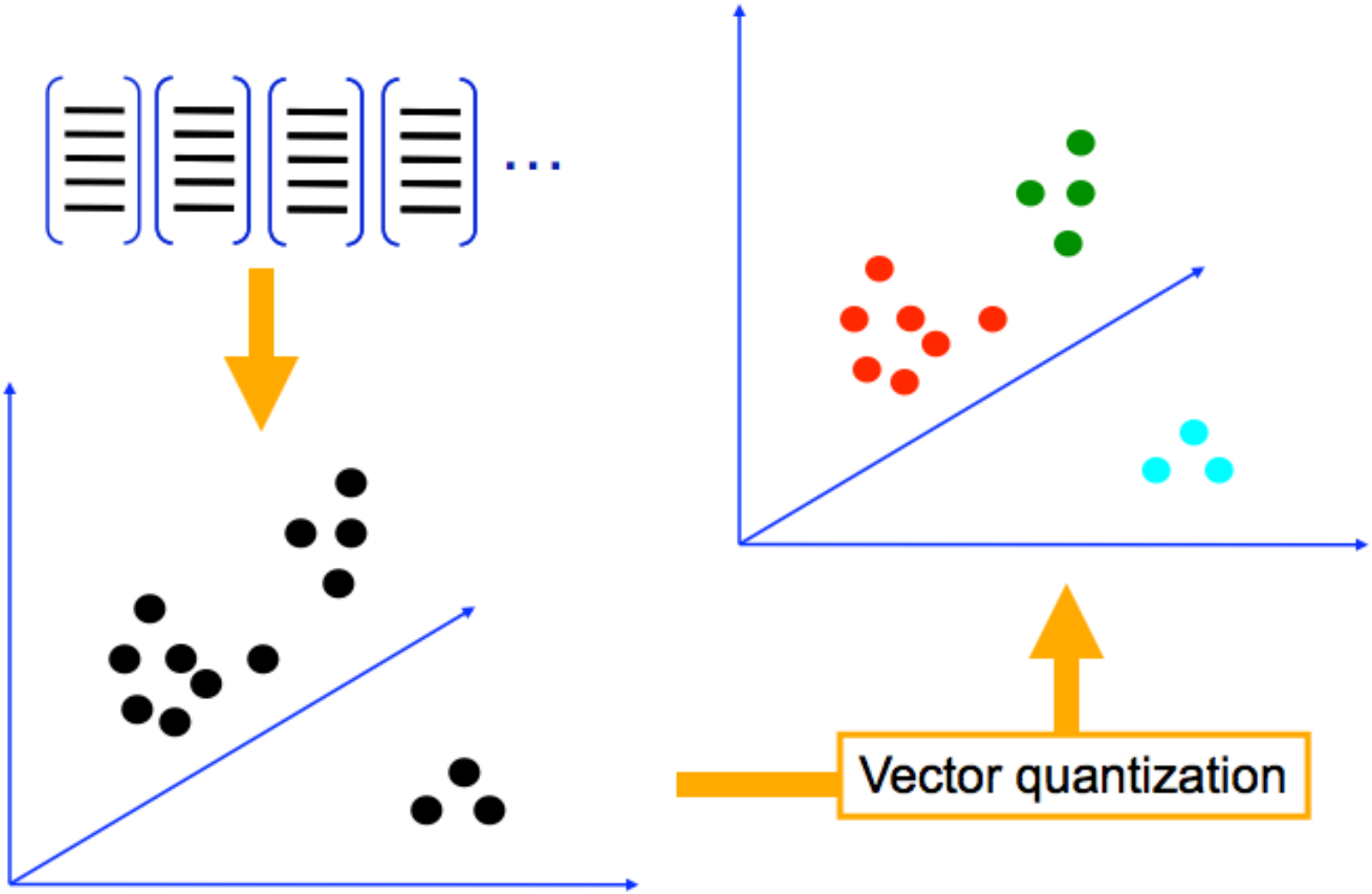
[Sivic & Zisserman, '03]

Slide credit: Josef Sivic

2. Codewords dictionary formation



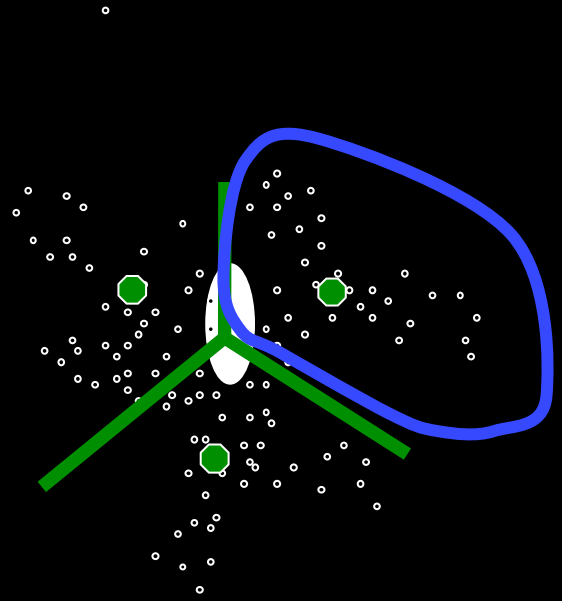
2. Codewords dictionary formation



Slide credit: Josef Sivic

Visual Words

- Example: each group of patches belongs to the same visual word



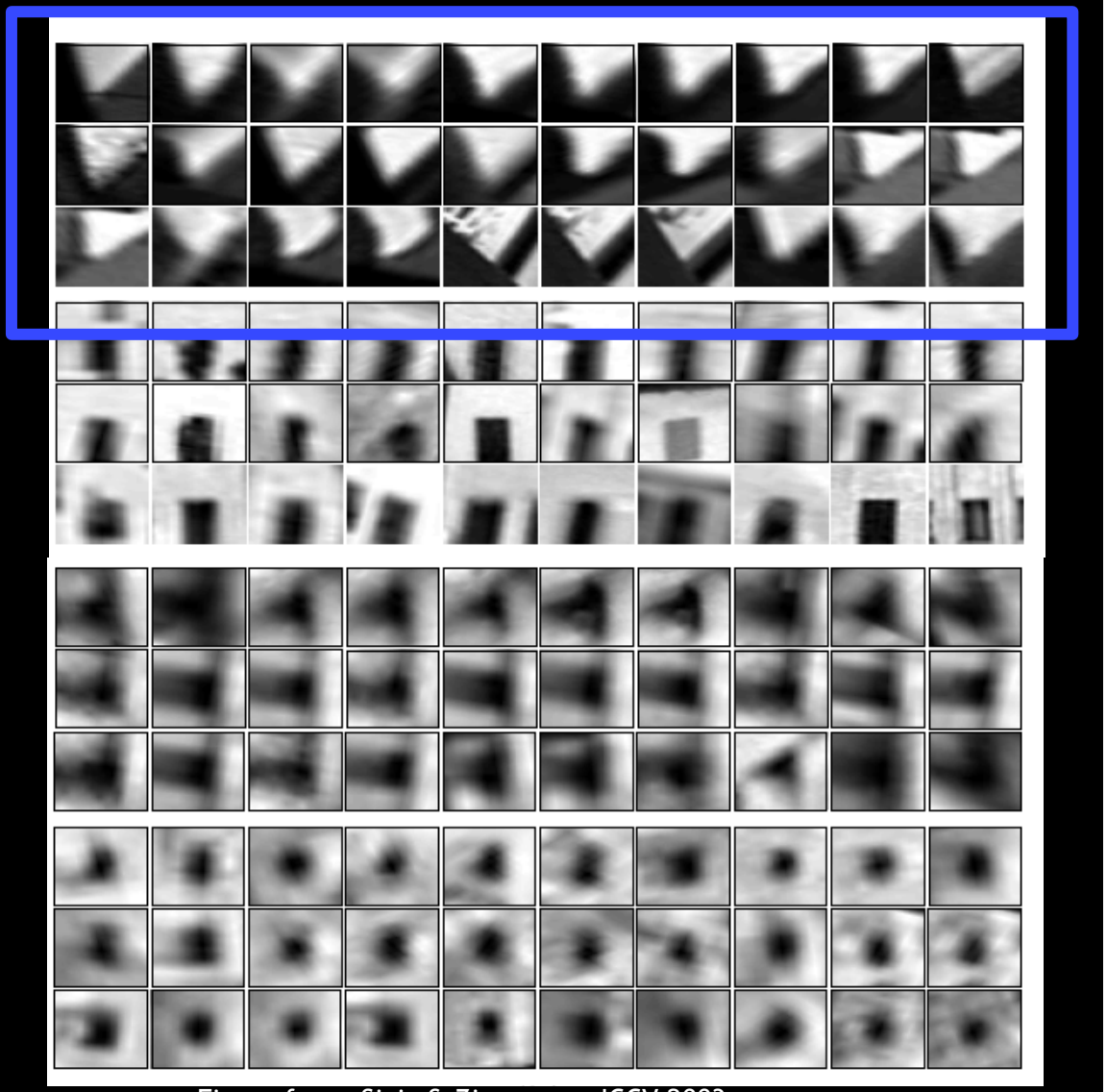
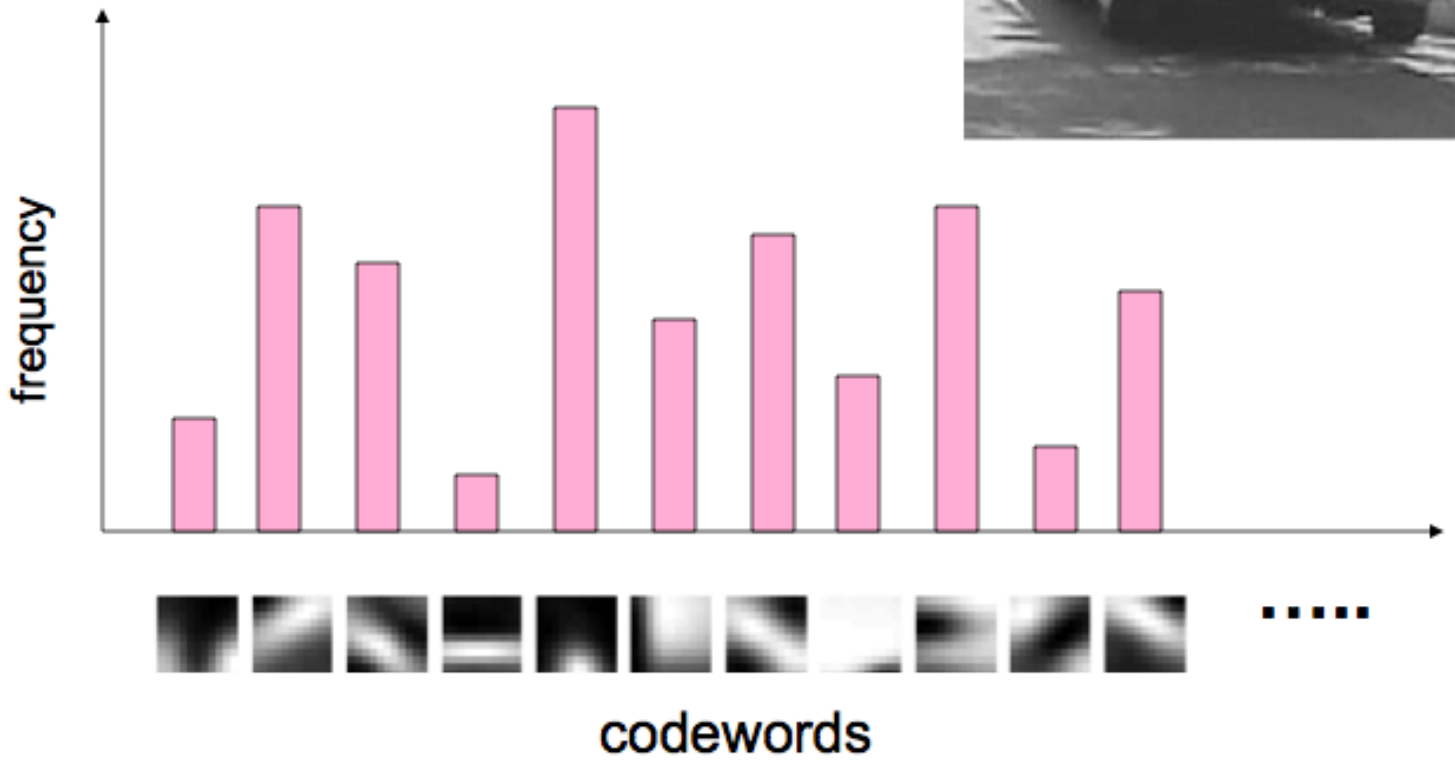


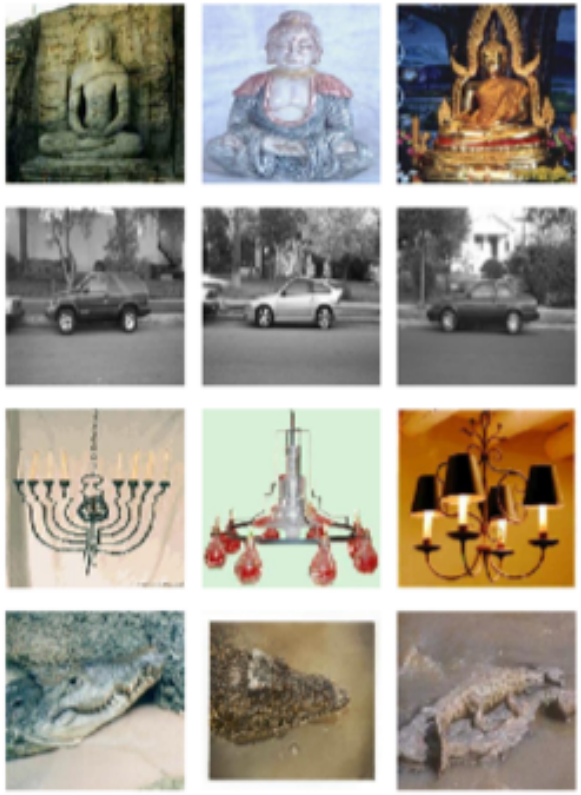
Figure from Sivic & Zisserman, ICCV 2003

3. Image representation



Object recognition results

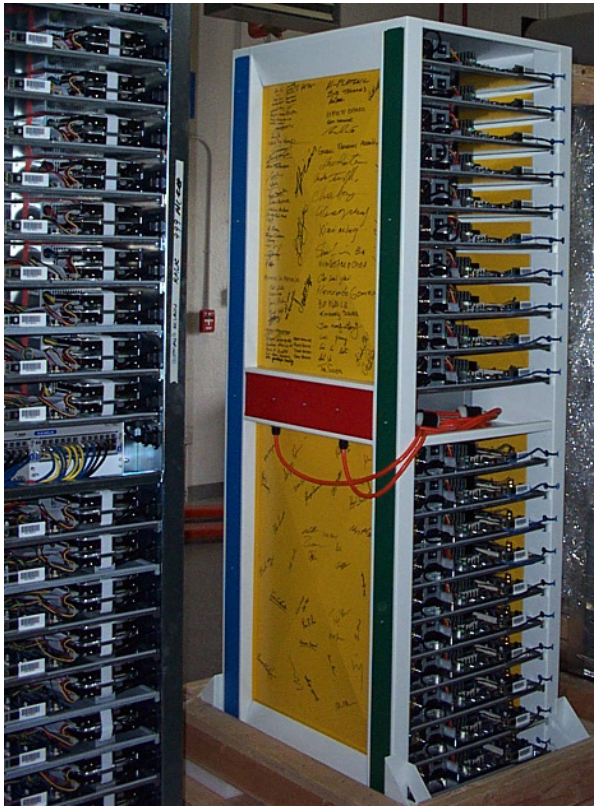
- Caltech objects database
  101 object classes
- Features:
  - SIFT detector
  - PCA-SIFT descriptor, *d*=10
- 30 training images / class
- 43% recognition rate
  (1% chance performance)
- 0.002 seconds per match



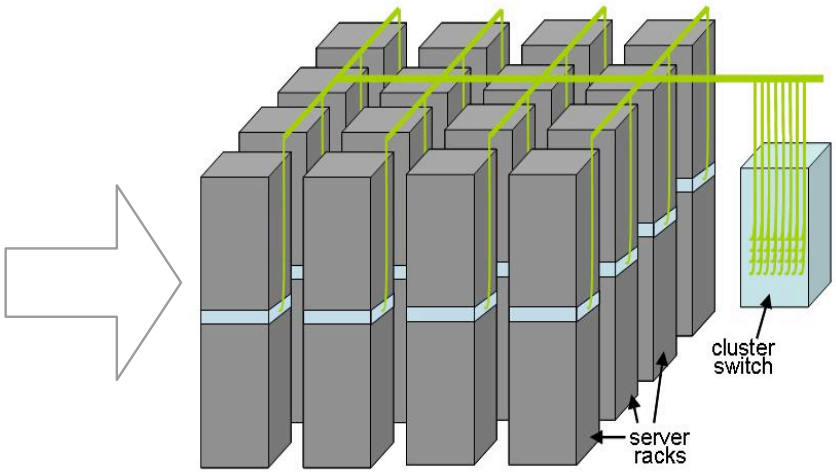
Slide credit: Kristen Grauman

Running in The Real World

Machines + Racks



Clusters



- In-house rack design
- PC-class motherboards
- Low-end storage & networking hardware
- Linux
- + in-house software

From Dean

Running in the Real World

Typical first year for a new cluster:

~1 network rewiring (rolling ~5% of machines down over 2-day span)

~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)

~5 racks go wonky (40-80 machines see 50% packetloss)

~8 network maintenances (4 might cause ~30-minute random connectivity losses)

~12 router reloads (takes out DNS and external vips for a couple minutes)

~3 router failures (have to immediately pull traffic for an hour)

~dozens of minor 30-second blips for dns

~1000 individual machine failures

~thousands of hard drive failures

slow disks, bad memory, misconfigured machines, flaky machines, etc.

Dean

Time for a Reality Check

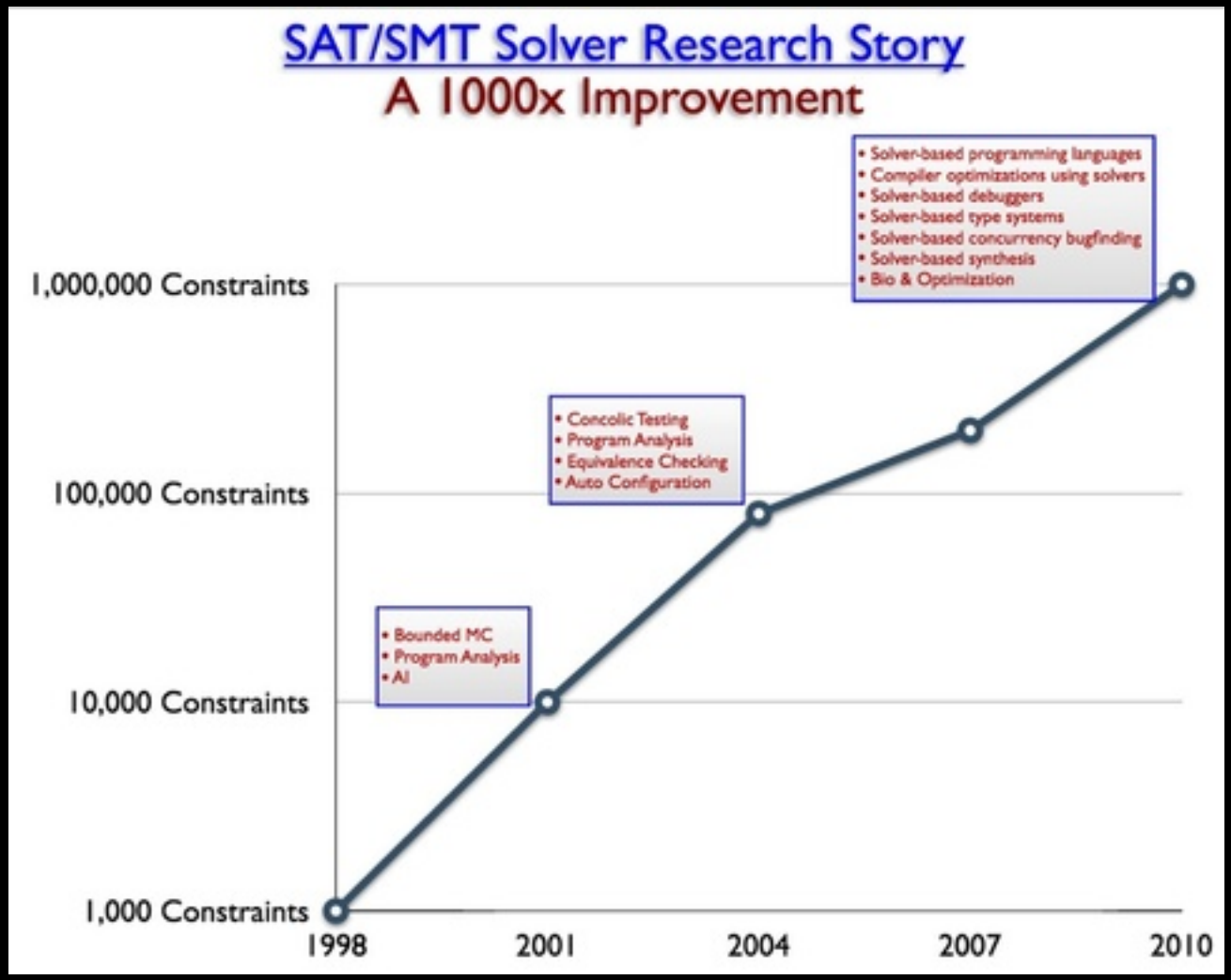
- Suppose you have a bug that is exercised once in a million queries
- How often will that bug be exercised in a day at Google?

3.5 billion queries/day -> 40k/second -> **every 25 seconds**,

A Slight Digression – SAT and Program Verification

- Simple problem: is a boolean formula satisfiable?
  - A & B === yes A=1, B=1
  - A & ~A === no!
- Original NP complete problem (Cook 1971)
- So What?

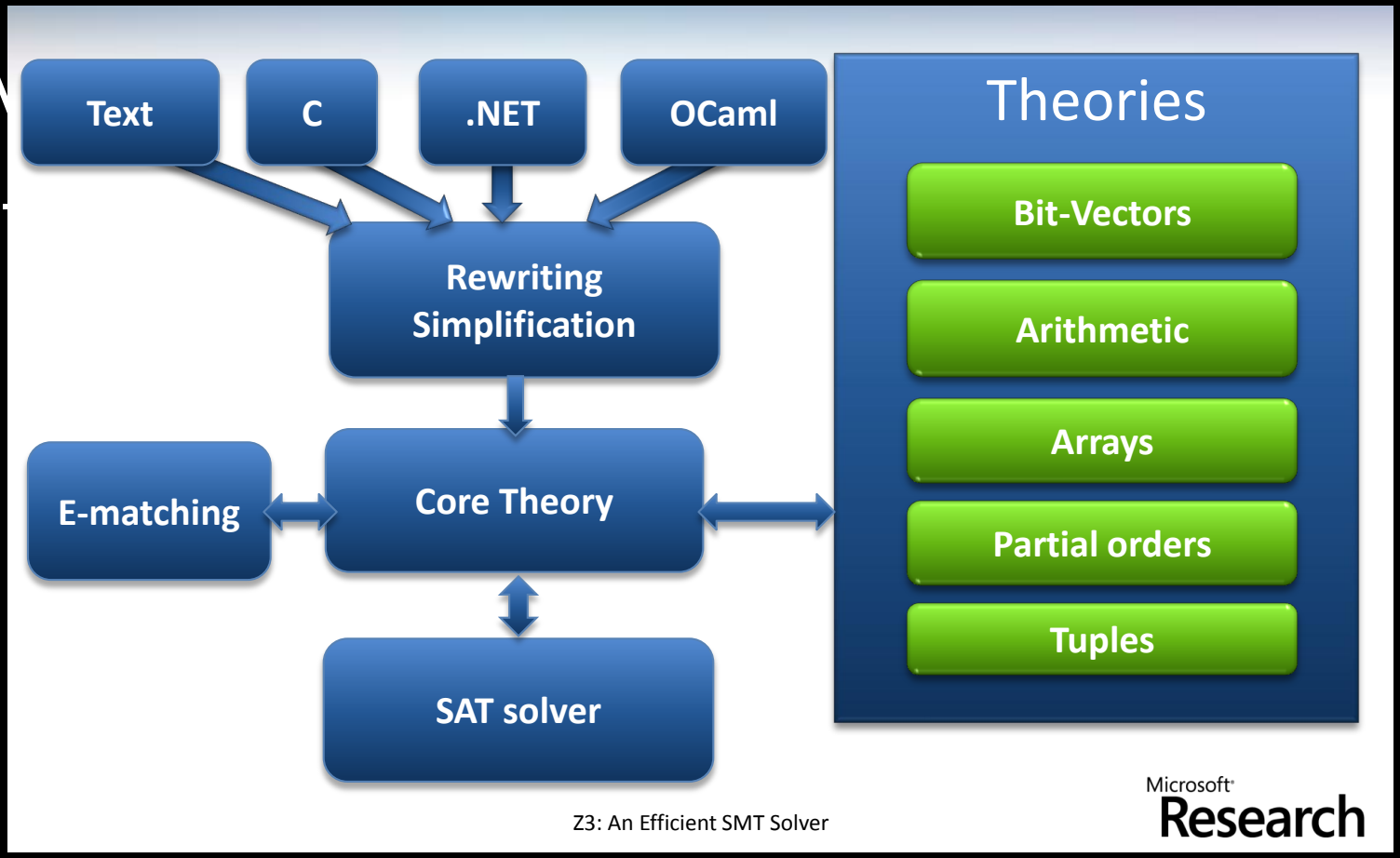
A Slight Digression -- SAT



Courtesy Vijay Ganesh

Add Some Horsepower

- N



A large fraction of windows bugs now found by program verifications

http://research.microsoft.com/en-us/um/redmond/projects/z3/Z3_System.pdf

Time for a Reality Check

- Suppose you have a bug that is exercised once in a million queries
- How often will that bug be exercised in a day at Google?

  3.5 billion queries/day -> 40k/second -> **every 25 seconds**,

- Even more complicated because queries are highly parallel!

A Paradigm Is Born

- Every problem has to deal with all of the possible hardware and software exceptions – lots of work!
- BUT, many of the underlying computations are "embarrassingly parallel"
  - Send a query to server (e.g. "do you have this term")
  - Aggregate the results
- The idea of **Map-Reduce**

Map Reduce

- User writes two main functions
  - Map -> the work each worker has to do – e.g. find docs for an index term
  - Reduce -> the work to combine the results – e.g. find the top n queries based on ranking
- System handles
  - Distribution, load balancing, communication, checkpointing …
- Apache Hadoop a common (open source) system for Map-Reduce

Computing at Scale

	Aug, '04	Mar, '06	Sep, '07	May, '10
Number of jobs	29K	171K	2,217K	4,474K
Average completion time (secs)	634	874	395	748
Machine years used	217	2,002	11,081	39,121
Input data read (TB)	3,288	52,254	403,152	946,460
Intermediate data (TB)	758	6,743	34,774	132,960
Output data written (TB)	193	2,970	14,018	45,720
Average worker machines	157	268	394	368

Dean: Map Reduce Statistics

Some Lessons

- Reality is a harsh taskmaster – many of the best ideas are forged from real problems
- It's usually not a single idea – borrow from the best!
- It's hard to trace the impact of ideas to fruition – at best we can do an anecdotal approximation; don't be fooled by an overly simplistic view!
- There are few truly failed ideas, just failed applications thereof – persevere!