Google

Research Challenges Inspired by Large-Scale Computation at Google

Jeff Dean
Google Fellow
jeff@google.com

User's View of Google

Organizing the world's information and making it universally accessible and useful

A Computer Scientist's View of Google

Problems span a wide range of areas:

Product design	Algorithms & Theory	…and much, much more!
User interfaces		
Machine learning, Statistics, Information retrieval, AI		
Compilers, Programming languages		
Networking, Distributed systems, Fault tolerance		
Hardware, Mechanical engineering		

Google

Overview

- A collection of problems we think are difficult/interesting
  - In some areas, significant work has been done/published
  - In others, topics are relatively new
- Not meant to be exhaustive catalog of problems/areas
  - We care about many other problems, too!
- Roughly ordered from lower-level (hardware design, distributed systems, ...) to higher-level (ML, IR, ...)
- Ideas collected based on suggestions from many colleagues
- Suggestions welcome!

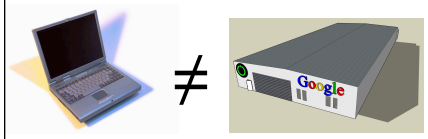
Google

Hardware & Energy Efficiency

- Moore's law is now scaling # cores instead of MHz
  - Fine with us: we love multicore machines for our problems
- Still want more computing capabilities, though...
  - Easy to get more computation by using more energy
  - Proportion of costs for energy will continue to grow, since Moore's law keeps computing cost roughly fixed
- Challenge: for every increase in HW performance, we need a corresponding increase in energy efficiency

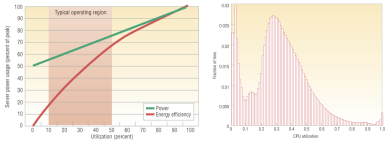
Google

Energy Efficiency



Google

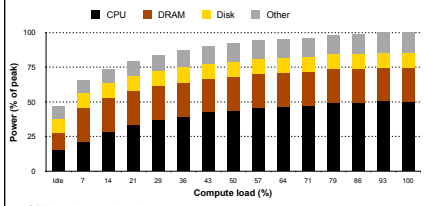
Energy Efficiency at Lower Utilization



- In a datacenter, machine utilization is usually 0.2 to 0.5
- In this range, energy efficiency is less than half the efficiency of a machine at 100% utilization
  - Okay for laptops, not so good for servers
- Challenge: Are there alternative designs that would give better energy efficiency at lower utilization?

Google

Power by component at different activity levels



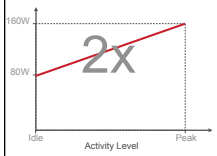
- CPU no longer dominates system power
- CPUs are more energy-proportional than the rest of the system

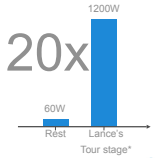
Google

A tale of two machines

==

Average power: ~120W

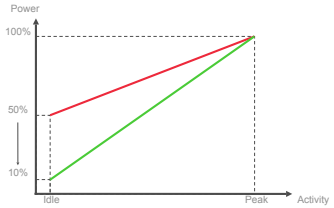




(*) Science World, 10/1/2001

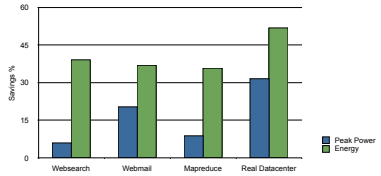
Google

What if we could build machines with a wide activity range?



Google

Estimated impact of energy proportionality



Both energy and power savings are very significant

Source: Fan, Weber, Barroso, ISCA'07

Google

Operating System Design

- Our production machines all run Linux
  - Design largely inspired by UNIX design from the 70s
  - Still works reasonably well for us, but:
    - We don't use many aspects of the system (e.g. paging to disk)
    - Clusters of tens of thousands of machines are pretty removed from original design point

Google

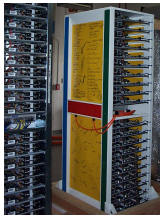
Operating System Design

- Challenge: Server O.S. design aimed at 1000s of highly-connected machines in one building
  - remote paging to other machine's memory?
  - redo networking stack (given RTTs of 0.1 ms, not 100s of ms)?
  - different security model?
  - top-to-bottom performance isolation across apps/machines?

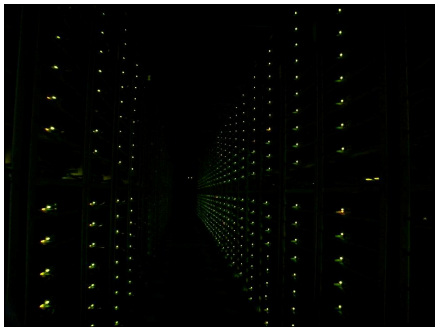
Google

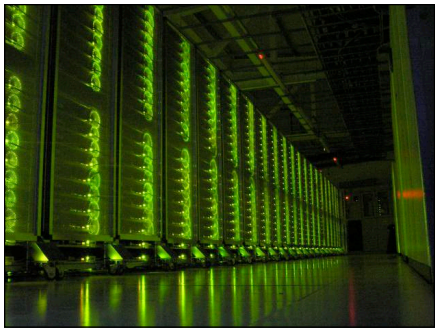
Current Machines

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



Google





The Joys of Real Hardware

Typical first year for a new cluster:

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
~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
plus slow disks, bad memory, misconfigured machines, flaky machines, etc.

- Long-haul networking breaks for unusual reasons, too:
  - Wild dogs, dead horses, thieves, blasphemy, drunken hunters and sharks

Google

Automated Systems Management via Machine Learning

- Challenge: machine learning techniques applied to monitoring/controlling/diagnosing such systems
  - automatic monitoring
    - learn to spot potential problems before they happen?
    - learn to spot unexpected failure modes?
    - automatically identify root causes of problems based on observed symptoms?
  - automatically figure out right strategies to adapt?

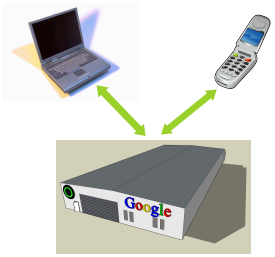
Google

Distributed Systems Abstractions

- High-level tools/languages/abstractions for building distributed systems
  - e.g. For batch processing, MapReduce handles parallelization, load balancing, fault tolerance, I/O scheduling automatically within a simple programming model
- Challenge: Are there unifying abstractions for other kinds of distributed systems problems?
  - e.g. systems for handling interactive requests & dealing with ***intra***-operation parallelism
    - load balancing, fault-tolerance, service location & request distribution, ...
  - e.g. client-side AJAX apps with rich server-side APIs
    - better ways of constructing client-side applications?

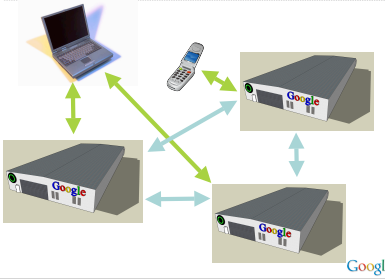
Google

Consistency in Internet-services



Google

Consistency in Internet-services (actually)



Google

Apps and Strong vs. Weak Consistency

- Many applications need state replicated across a wide area
  - For reliability, availability, and low latency access
- Two main choices:
  - consistent operations (e.g. use Paxos)
    - often imposes additional latency for common case
  - inconsistent operations
    - better performance/availability, but apps harder to write and reason about in this model
- Many apps need to use a mix of both of these:
  - e.g. Gmail: marking a message as read is asynchronous, sending a message is a heavier-weight consistent operation

Google

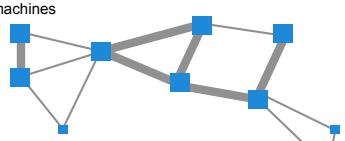
Using Weakly Consistent Storage Systems

- Challenge: General model of consistency choices, explained and codified
  - ideally would have one or more "knobs" controlling performance vs. consistency
  - "knob" would provide easy-to-understand tradeoffs
- Challenge: Easy-to-use abstractions for resolving conflicting updates to multiple versions of a piece of state
  - Useful for reconciling client state with servers after disconnected operation
  - Also useful for reconciling replicated state in different data centers after repairing a network partition

Google

Design of Very Large-Scale Computer Systems

- Future scale: ~10^6 to 10^7 machines, spread at 100s to 1000s of locations around the world, ~10^9 client machines



  - zones of semi-autonomous control
  - consistency after disconnected operation
  - power adaptivity

Google

Adaptivity in World-Wide Systems

- Challenge: automatic, dynamic world-wide placement of data & computation to minimize latency and/or cost, given constraints on:
  - bandwidth
  - packet loss
  - power
  - resource usage
  - failure modes
  - ...
- Users specify high-level desires:
  "99%ile latency for accessing this data should be <50ms"
  "Store this data on at least 2 disks in EU, 2 in U.S. & 1 in Asia"

Google

Privacy vs. Sharing

- There are undoubtedly people in the world that share some of your interests that you don't know
  - e.g. others with interests in "`kite-based power generation`"
- Data driven services are growing and increasingly important
  - ... but policy and technical reasons can limit what can be done
- Challenge: How can we build useful services that match people together based on their interests/ behavior (for example) in ways that preserve their privacy?

Google

ACLs in Information Retrieval Systems

- Retrieval systems with mix of private, semi-private, widely shared and public documents
  - Spectrum from:
    - private e-mail
    - shared doc among 10 people
    - message in group with 100,000 members
    - public web pages
- Challenge: building retrieval systems that efficiently deal with ACLs that vary widely in size
  - best solution for doc shared with 10 people is different than for doc shared with the world
  - sharing patterns of a document might change over time

Google

Automatic Construction of Efficient IR Systems

- Currently use several retrieval systems
  - e.g. one system for sub-second update latencies, one for very large # of documents but daily updates, ...
  - common interfaces, but very different implementations primarily for efficiency
  - works well, but lots of effort to build, maintain and extend different systems
- Challenge: can we have a single parameterizable system that automatically constructs efficient retrieval system based on these parameters?

Google

Information Extraction from Semi-structured Data

- Data with clearly labelled semantic meaning is a tiny fraction of all the data in the world
- But there's lots semi-structured data
  - books & web pages with tables, data behind forms, ...
- Challenge: algorithms/techniques for improved extraction of structured information from unstructured/ semi-structured sources
  - noisy data, but lots of redundancy
  - want to be able to correlate/combine/aggregate info from different sources

Google

Machine Learning Algorithms

- Challenge: Non-brittle/understandable ML systems
  - develop algorithms that can account for:
    - systematic differences between characteristics of sources of training data and data encountered in the field
    - incorrect/uncertain information in training data (due to human rater error or use of proxies)
  - provide concise explanations of their reasoning
- Challenge: Computationally efficient ML algorithms
  - data is often very large, so efficient algorithms are essential
  - are principled nearly-linear-time algorithms possible, e.g. for clustering?

Google

Long Distance Dependencies in Text Processing

- N-gram/phrase-based methods are pretty good for local dependencies (trained on gigantic amounts of data)
  - ...but very poor quality for nonlocal dependencies or long-distance reordering
- Challenge: Exploiting/handling non-local dependencies in text processing
  - Fundamental issue in machine translation, speech recognition, natural language processing, ...
  - Combinatorial explosion of possible dependencies; most of them irrelevant; how to filter relevant ones?

Google

Robust Speech Recognition Systems

- In addition to better language/semantics models, we need much better low-level acoustic models (probably with more features)
- Current speech recognition systems are very brittle:
  - e.g. system that does 15% WER on broadcast news drops to 30-40% on (somewhat) mismatched data such as YouTube news-like video
- Challenge: give speech recognition systems "domain" independence/robustness through:
  - larger acoustic/language models

  and/or
  - "domain" adaptation, for some definition of "domain"

Google

Computer Vision

- Explosion of digital images and videos
  - will be more useful if we could extract/summarize/search it
- Many successful algorithms in restricted domains:
  - e.g. porn detection, face detection, OCR
- Initial promising work on modeling visual cortex
- Challenge: general-purpose machine vision systems
  - examine image/video & generate human-like summary:
    - "a brown horse in a meadow"
  - requires substantial progress in scene geometry analysis, robust object detection, recognition systems, ...
  - ... or maybe completely different approaches

Google

Conclusions

- Connectivity + computational power + explosion of available information have created new classes of challenging computing problems
- Solving these will require substantial progress across many disciplines
  - most individual problems span multiple sub-disciplines

Google

Thanks!

- Helpful suggestions from Luiz Barroso, Ciprian Chelba, Tom Dean, Alon Halevy, Urs Hölzle, Waldemar Horwat, Phil Long, Dick Lyon, Mike Marty, Muthu Muthukrishnan, Franz Och, Rob Pike, Alfred Spector, Brian Strope, and others.

- Questions? Thoughts?
- More info:

  http://labs.google.com/people/jeff

  http://labs.google.com/papers

Google