Microsoft Windows Azure Storage, Facebook's Data Storage and Google Search Architecture

Hung-Wei Tseng

Web search for a planet: The Google cluster architecture

Luiz Andre Barroso, Jeffery Dean ; Urs Holzle
Google

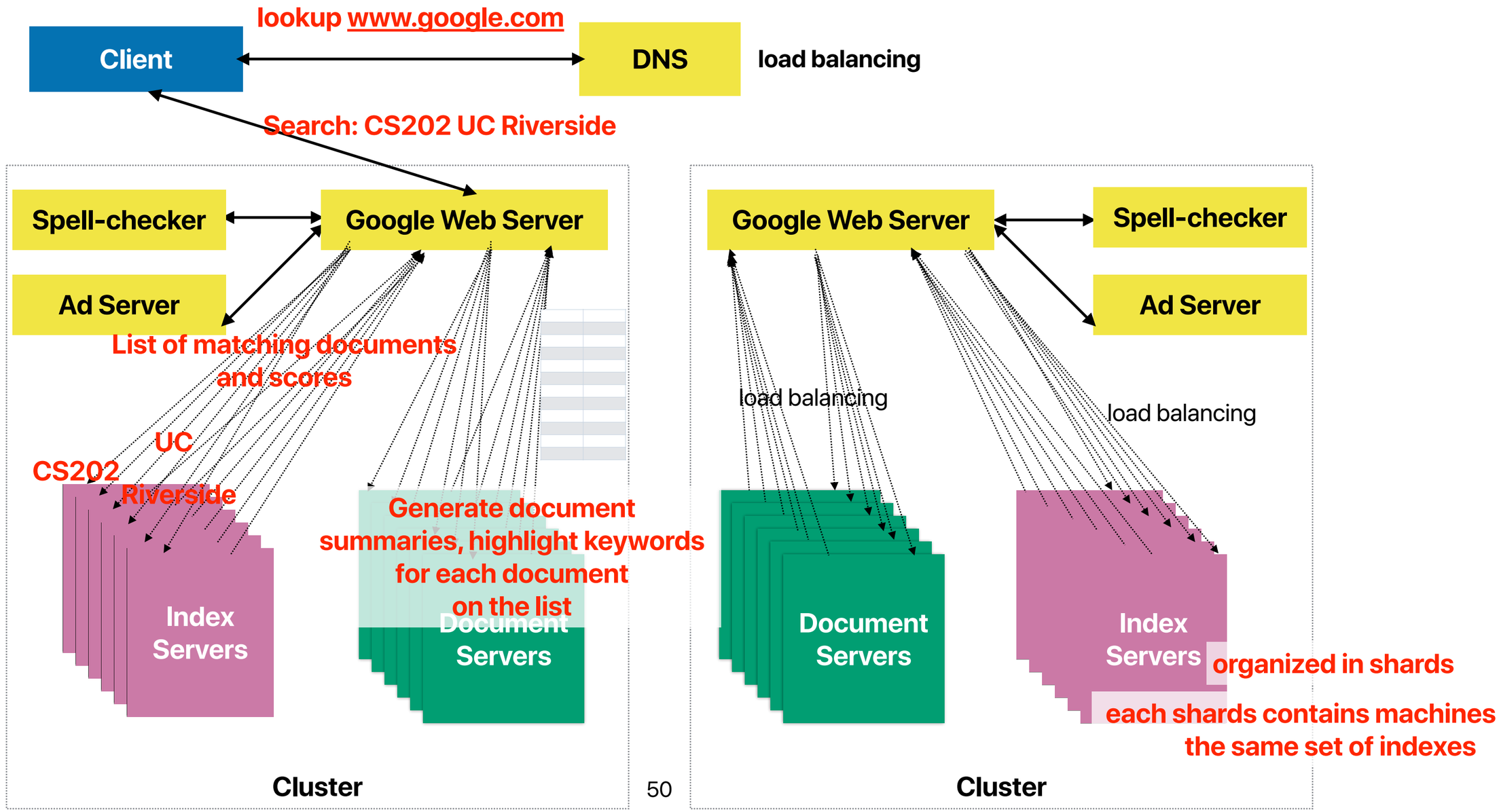
Why Google Search Architecture?

- The demand of performing search queries efficiently
  - Each query reads hundreds of MBs of data
  - Support the peak traffic would require expensive supercomputers or high-end servers
- We need a **cost-effective** approach to address this demand
  - Google search is compare against “AltaVista” search engine that uses DEC's high-performance alpha-based multiprocessor systems
  - AltaVista is later acquired by Yahoo! and you know the later story…

What Google proposes?

- Using commodity-class / unreliable PCs
- Provide reliability in software rather in hardware
- Target the best aggregate request throughput, not peak server response time

Google query-serving architecture



Replication is the key

- Scalability: simply add more replicas, the service capacity can improve
- Availability: even though one machine fails, another replica to take over

Hardware

- Processor
  - Index search has little ILPs — doesn't need complex cores
  - Index search can be highly parallelized — processors with thread-level parallelism would be a good fit (e.g. Simultaneous Multithreading, SMT and Chip Multicprocessor, CMP)
  - Branch predictor matters
- Memory: Good spatial locality. Moderate cache size will suffice
- Storage: No SCSI, No RAID — not worth it
- Power: is an issue, but only $1,500/mo operating bill vs $7,700 capital expense

Will their architecture work for other things?

As mentioned earlier, our infrastructure consists of a massively large cluster of inexpensive desktop-class machines, as opposed to a smaller number of large-scale shared-memory machines. Large shared-memory machines are most useful when the computation-to-communication ratio is low; communication patterns or data partitioning are dynamic or hard to predict; or when total cost of ownership dwarfs hardware costs (due to management overhead and software licensing prices). In those situations they justify their high price tags.

At Google's scale, some limits of massive server parallelism do become apparent, such as the limited cooling capacity of commercial data centers and the less-than-optimal fit of current CPUs for throughput-oriented applications. Nevertheless, using inexpensive PCs to handle Google's large-scale computations has drastically increased the amount of computation we can afford to spend per query, thus helping to improve the Internet search experience of tens of millions of users. MICRO

At first sight, it might appear that there are few applications that share Google's characteristics, because there are few services that require many thousands of servers and petabytes of storage. However, many applications share the essential traits that allow for a PC-based cluster architecture. As long as an application orientation focuses on the price/performance and can run on servers that have no private state (so servers can be replicated), it might benefit from using a similar architecture. Common examples include high-volume Web servers or application servers that are computationally intensive but essentially stateless. All of these applications have plenty of request-level parallelism, a characteristic exploitable by running individual requests on separate servers. In fact, larger Web sites already commonly use such architectures.

Metrics we care about data center design

- Costs — machine architecture, distributed system architecture, replication strategies
- Power — machine architecture
- Energy — machine architecture
- Space-efficiency — erasure coding, replication, distributed
- Throughput — replication, distributed
- Reliability — replication