Google Searches

1000s of queries per second

Each query reads 100s MBytes data

How achieve that kind of performance?

15000 commodity PCs

Highly parallel, non-transactional design
Well, Not Really ...

Each Google cluster = a few thousand PCs

DNS Load Balancing scheme sends each client query to nearest Google cluster

Hardware Load Balancer sends arriving query to least loaded Google Web Server in cluster
Processing a Query

Index server holds *shard* -- random subset
Multiple server replicas for each shard
Route requests through load balancer
Now have ordered list of matching docids
Doc servers hold random subsets
Route requests through load balancer
Design Principles

Get reliability from fault-tolerant software

Use replication to improve throughput and availability

Price/performance beats peak performance

Commodity PCs provide cheapest cycles
“Maintaining 1000 servers is not more expensive than 100 if they all have identical configurations.” Phew!

Power: densely packed PCs = 400-700 w/ft
data center = 150 w/ft

Claim: Pentium 4 speculative execution already beyond point of diminishing return
DNS Load Balancing

Ordinary DNS Lookup

User enters www.loadbalancedsite.com in the address box of the web browser.

DNS Server

www.loadbalancedsite.com
IP address is
203.24.23.3

Web Browser

Internet

Web Browser

GET 203.24.23.3

Web Browser

Internet
DNS Load Balancing

Round-Robin DNS Lookup

Cluster

203.34.23.3

Round Robin DNS

203.34.23.4

203.34.23.5

www.loadbalancedsite.com
Hardware Load Balancing

Client always sees same address

Load balancer interprets session data to choose destination

Cluster

Hardware Load Balancer
Virtual IP: 203.34.23.10
Redirects the user requests to a node in the cluster based on the information in the header, Cookies or URL data
Encryption?

Must decrypt to interpret session data
Apache Load Balancing

Apache serves static pages

Requests for dynamic pages routed to back-end Tomcat
Encryption?

Apache serves static pages
Also decrypts HTTPS
Requests for dynamic pages routed to back-end Tomcat cluster
Hardware for that too ...

Works like the Apache/Tomcat but faster
And it costs a lot more