Enterprise Applications

Chi Ho Yue
Sorav Bansal
Shivnath Babu
Amin Firoozshahian

EE392C – Emerging Applications Study
Spring 2003

Functionality

- Online Transaction Processing (OLTP)
  - Users/apps interacting with database in real-time
- Online Analytical Processing(OLAP) / data mining
  - Experts doing “offline” data analysis
- Web servers
  - Serves static HTML / dynamically generated pages
- File servers
  - Provide access to stored data over the network
- Video servers
  - Special type of file servers

Data Set and Working Set

- OLTP: few GB
- OLAP: few hundred TB / 1-2 PB
- Web servers
  - 10k-500k hits/hr. Around 100 concurrent connections
  - BBC sees 200 hits/sec. Peak: 2000 hits/sec
  - File size: frequently 4KB, avg. 18KB
- File/video servers
  - Huge amounts of stored data (TB/PB)
  - Request size widely varies

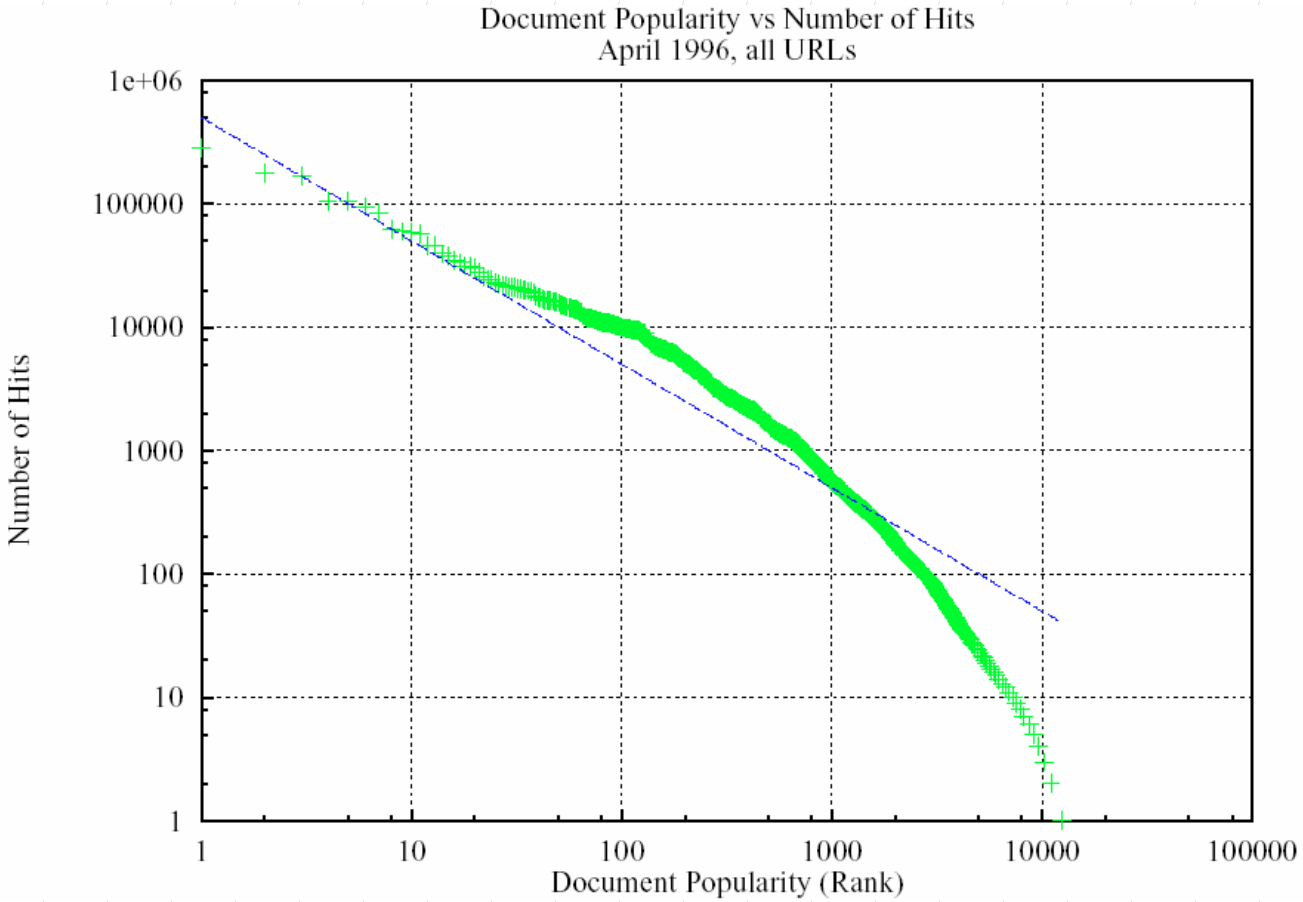
Computational Complexity

- OLAP/data mining
  - Could get complex: e.g., sort, pattern-matching
- OLTP
  - Usually minimal, depends on applications
  - Mostly transaction code
- Web server: 1 parent, 40 child processes
  - Each child process parses request, retrieves content, writes to TCP connection
  - Tree like path (1-out of-5 instructions is a branch)
- File & video servers
  - Algorithms like : scheduling, pre-fetching, data distribution, and fault tolerance

Memory Access Behavior

- OLTP and OLAP/mining
  - Indexed and sequential access patterns
  - Good spatial locality, limited temporal locality
  - Large memory footprints ➔ conflict misses
- File and video servers
  - Memory largely used as caches and buffers
  - Large spatial locality in video servers
  - Good pre-fetching behavior for video servers

Webservers: Strong Locality of Access Patterns



Server side temporal Locality:

LFU caching Works best

I/O Accesses

- OLTP
  - I/O bus on multithread SMP memory system, bottleneck for scalability
  - High bandwidth for simultaneous accesses
- OLAP/mining: I/O is not primary bottleneck
  - High performance disk arrays (RAID)
  - Exploit TLP to hide latency
- File and video servers
  - Huge volumes of storage with high bandwidth to stored data
  - Fast network connections

Types of Parallelism

- OLTP and OLAP
  - High TLP: thread per query/transaction
  - Limited ILP: more dependencies, less loops
- Web servers:
  - TLP (24-40 child threads run concurrently)
  - Limited ILP (studies revealed IPC on Pentium pro 1.5 times better than on Pentium)
- File and video servers
  - Limited ILP, mostly control code

Architectural Requirements

- OLAP
  - High TLP ➔ SMP, SMT work well
  - Need high performance disk arrays to hide I/O latency
  - Memory stalls are biggest bottleneck: large footprint
- OLTP
  - SMP systems: parallel processing of shared data
  - Precise exceptions, atomicity, coherence, fault tolerance
- Web Server
  - Cache Size impacts performance greatly
  - I/O bandwidth
  - Branch Prediction/Speculation does not work too well
- File and video servers
  - Huge storage
  - High bandwidth to stored data

Benchmarks

- OLTP
  - TPC-C : throughput for a mixture of read-only and update intensive transactions
- OLAP
  - TPC-R, TPC-H : models a decision support system in a manufacturing application; 22 complex SQL queries
  - Throughput, price/performance
- Web servers
  - SPECweb96, SPECweb99, Webstone
- File servers
  - SPEC SFS : Throughput vs. Response time
  - Postmark : pool of dynamic and small files

Scaling Trends

- OLTP, Web Servers
  - Increasing number of simultaneous requests
- OLAP
  - Increasing data sets
  - Trying to close gap between data and decision
- What to do?
  - Distributed Servers
  - RAID – have many disks per processor
  - Bus bandwidth needs to scale

Emerging Applications

- Streaming applications
  - Processing continuous streams instead of stored data
  - Monitoring applications: network monitoring, intrusion detection, stock monitoring
  - Run-time profiling and adaptivity is key
- Sensor data management
  - Traffic monitoring and military applications
- Peer-to-peer systems

Conclusion

- Lots of TLP
- Mostly control code ➔ limited ILP
- Bottleneck
  - Cache misses (greatly impacts performance)
  - I/O bandwidth (disk to memory)
  - Network bandwidth (memory to network)
  - Branch mis-prediction rate (tree-like path)
  - Speculative and OOO execution would be less useful