Enterprise Applications

Chi Ho Yue
Sorav Bansal
Shivnath Babu
Amin Firoozshahian

EE392C – Emerging Applications Study
Spring 2003

Applications Studied

- 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

Online Transaction Processing (OLTP)

- The delivery of information, products, services, or payments via digital computer networks
- Users/apps interacting with database in real-time
- Example:
  - online banking, online payment
  - eBay, Paypal, etc...

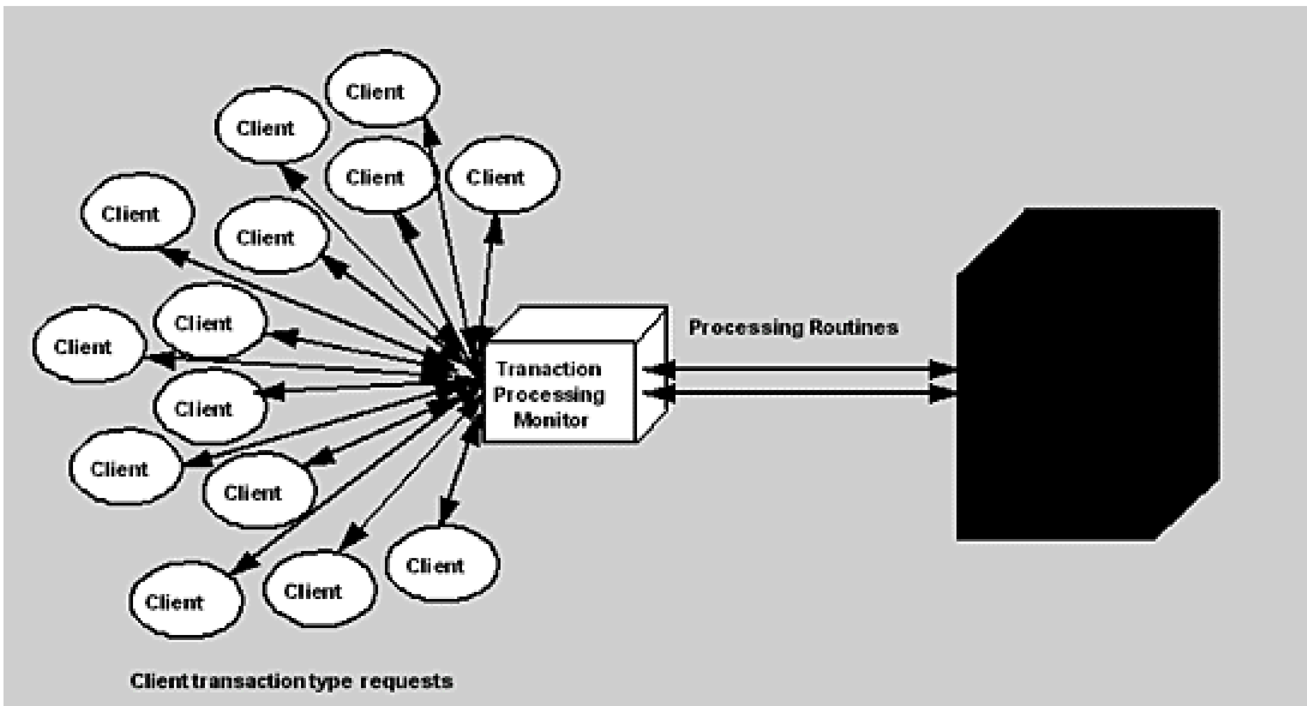
Architectural Requirements - I

- Data volume is very large, requires large storage for client account information
- Computational complexity of OLTP is usually minimal, depending on the particular application

Architectural Requirements - II

- Memory operations to arithmetic operations ratio is high; data of each individual client will be loaded in every transaction
- High bandwidth to storage and to network is favorable because both end can be the bottleneck of the system

Example of OLTP system



Memory Access Patterns and Behavior

- Distributed access to a single resource of data
- Access to distributed resources from a single application component
- Required properties of memory access
  - Atomicity, Consistency, Isolation, Durability
  - Coherence

Required Memory Properties I

- Atomicity
  - Transactions should be done completely and unambiguously
  - Transactions should be undone and data should be rolled back when a failure in operation occurs
  - Requires precise exception handling

Required Memory Properties II

- Coherence
  - During the course of a transaction, intermediate (possibly inconsistent)
  - State of the data should not be exposed to all other transactions
  - Two concurrent transactions should not be able to operate on the same data
  - Database management systems usually implement this feature using locking

Type of Parallelism

- Thread-level parallelism
  - Same instructions, different data sets
- The participating operations are executed sequentially or in parallel threads requiring coordination and/or synchronization
- Symmetric multiprocessor (SMP) is currently the most popular server product for commercial application

Benchmarks

- Transaction Processing Performance Council (TPC)
- TPC-C for commercial workload
  - Mixture of read-only and update intensive transaction
  - Simulate a complete computing environment: terminal operators and database
- Measured performance metric
  - Throughput: transactions per minutes (TPC-C transaction)

Future Trends

- I/O system connected to the SMP is a potential bottleneck
- Scalability is a major limit on bus-based shared memory multiprocessors
- New research on alternative effective configuration of I/O system

Summary

- Lots of thread level parallelism
- Atomicity requires precise exception handling
- Data on memory and cache requires coherence
- I/O is the bottleneck for SMP based OLTP system
- Research on alternative effective I/O configurations

Databases for Decision Support

- On-Line Analytical Processing (OLAP)
- Data warehousing and mining

OLTP

- Users/apps interacting with database in real-time
- E.g., customers buying/selling books at Amazon
- As old as databases
- Large number of concurrent users, queries, connections

OLAP/Mining

- Experts doing "offline" data analysis
- E.g., Amazon wants to know which books are hot
- Emerged around late 80's
- Small number of concurrent users, queries, connections

OLTP Vs. OLAP/Mining (contd.)

OLTP	OLAP/Mining
Simple queries, often predetermined,	Complex, ad-hoc queries
Relatively little data (~GB)	Very large data sets (~TB)
Each query touches little data	Queries touch large data sets
Little computation per query	Mining is compute intensive
Simple computation	Complex operations in mining
Data is continually updated	Data is mostly read-only
Accuracy and recovery important, hence strict transactions	Strict transactional semantics is not needed
Throughput is most important	Latency is more important

OLAP/Mining: Data, Computation, and I/O

- Very large data sets
  - E.g., Walmart data warehouse is >24 TB
- Range of computational complexity
  - Compute-intensive data preprocessing, e.g., sort, indexing
  - Most queries perform simple computation
  - Complex mining tasks, e.g., pattern analysis
- OLAP/Mining is no longer I/O bound
  - Highly parallel disk arrays (RAID)
  - Asynchronous I/O with sequential log writes
  - Autonomous DMA engines, larger memory
  - Aggressively exploit thread-level parallelism

Memory Behavior and Parallelism

- Memory Behavior
  - Indexed and Sequential access patterns
  - Good spatial locality
  - Little data reuse across queries
  - Low ratio of arithmetic operations to memory accesses. Exception: some mining tasks
- Parallelism
  - ILP not very effective
    - Instruction dependencies are high
    - Lesser number of loops compared to other software
  - High DLP and TLP

Main Performance Bottleneck: Memory Stalls

- High L1 instruction and L2 data cache misses
  - Large memory footprints
  - Significant conflict misses
- But, memory stalls less severe than in OLTP
  - Smaller instruction footprint
    - Reduced transactional, security components
  - More computation and data reuse
  - Less synchronization

Memory Stalls: Some Observations

- Poor OS page mapping policy causes cache conflicts
- Page mapping based on reference order works best
- Offset conflicting virtual-address-space structures
- Small cacheable “critical” working sets exist
- Larger caches help, but not much
- Multiple contexts and prefetching very effective
- Use cache-conscious page layouts and structures

Benchmarks

- TPC-H is the popular OLAP benchmark
  - Models decision support for a large manufacturer
  - 22 complex SQL queries
  - Metrics: queries per hour, price/performance
- Many data mining benchmarks
  - Yearly KDD Cup
  - Intrusion detection benchmark
  - Metrics: precision, recall

Hot App: Processing Continuous Streams

- Monitoring applications, real-time needs
  - Network monitoring and intrusion detection
  - Processing sensor data in military applications

Database System

- Queries pull stored data
- OLTP, OLAP-style (one-time) queries
- Statistics available on stored data
- Traditional one-time query optimization

Data Stream System

- Streams pushed at system
- Long running (continuous) queries
- Stream characteristics often unknown and time-varying
- Online profiling and adapting necessary

Data, Computation, and I/O

- Continuously arriving data streams
  - Up to gigabits per second in network monitoring
- Would like to run continuous OLAP/mining queries
- Most processing on recent windows over streams
  - E.g., stock ticks in the last hour
- Working set for typical systems might fit in memory
  - Disk mostly for archiving purposes
- Disk latency hiding like OLAP should work

Data Stream Systems: Performance Characterization

- No real data available. System development in progress
- Workload characteristics between traditional OLAP and Imagine-style media processing
  - Large windows over streams require non-sequential access
- Fast streams will stress cache performance
- Stream data and arrival characteristics change
  - Continuous profiling and adaptivity will be important

WebServers

- A WebServer typically serves
  - Static HTML Pages (including images, a very small number of media files)
  - Runs CGI scripts to dynamically generate pages
  - Recent Webservers run a JVM to run Java servlets which serve dynamic web pages

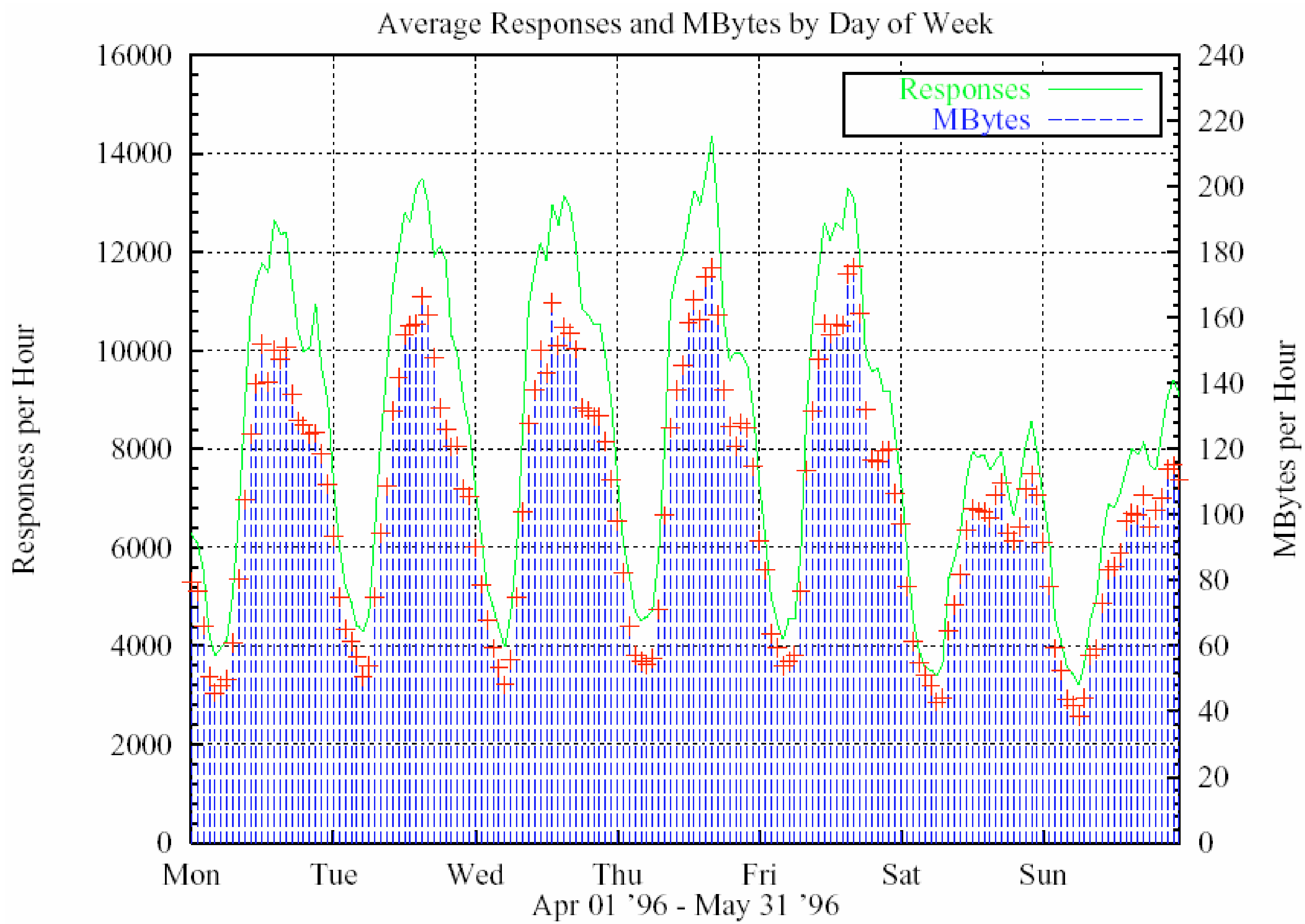
Typical WebServer Workloads

- 2.5 Gbytes/day or 30 KB/sec
- 8400 hits/hour. For popular websites like BBC: 200hits/sec. Peak: 2000hits/s
- Typically 24-40 concurrent connections
- Peak 80-100 concurrent connections
- Throughput per connection typically 1.5KB/sec
- Images constitute 90% of Byte Traffic
- Most Frequent File Size = 4KB. Average File Size = 18KB

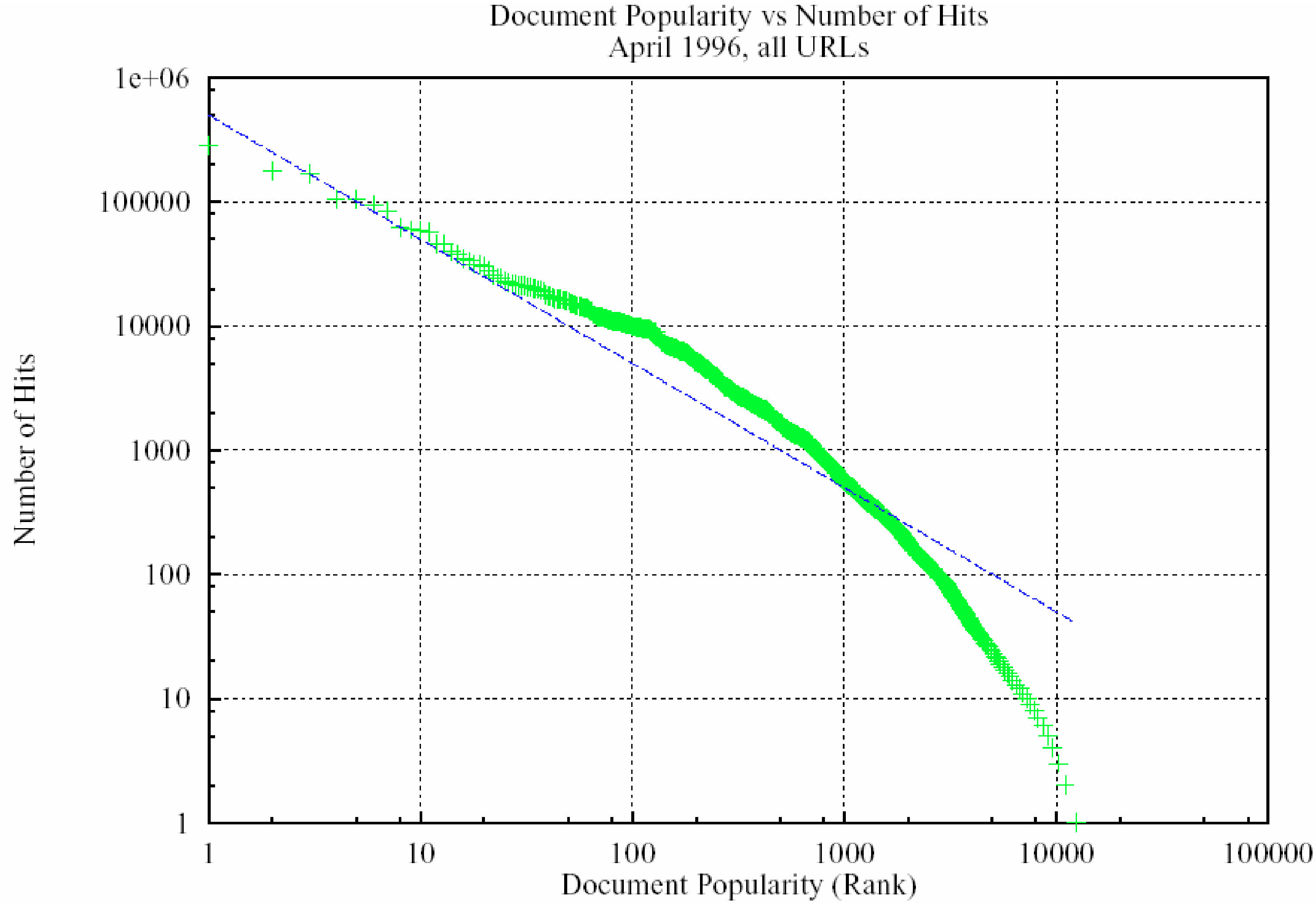
WebServer Tasks

- 1 server PARENT process to receive all incoming requests and spawn children
- Typically 40 “pre-forked” CHILD daemon processes
- Each Child Process:
  - Parses request
  - Retrieves Content (may involve running CGI script)
  - Writes result to TCP connection

Execution Behavior



Locality of access



Workload Comparison On two different machines

- Pentium
  - In order superscalar (2-way)
  - 8KB L1 D-cache. No L2 Cache
  - 64 bit inorder bus
- Pentium Pro
  - Speculative, OOO (upto 3 micro-ops/cycle)
  - 8KB L1 D-cache, 256KB L2 unified cache
  - 64 bit split-transaction bus

Web Server Workload Characteristics

	Pentium	Pentium Pro
CPI	6.65	3.45
Branches/Inst.	0.20	0.20
D-cache miss rate	0.12	0.04

Conclusions:

1. High number of branches
2. Cache Miss Rate crucial
3. There is some amount of ILP

Some Observations

- Key factors to keep in mind
  - Cache Size
  - I/O Bandwidth (disk to memory)
  - Bus Bandwidth (memory to network)
  - Server-side temporal locality (LFU caching works best)
- Obstacles
  - Higher branch misprediction ratio (tree-like execution path
  - Speculative and OOO execution would be less effective

Summary

- Temporal Locality (LFU works well)
- Types of Parallelism
  - TLP (many independent child threads)
  - ILP (demonstrated by Pentium studies)
- Key Factors – Caching, Bus Bandwidth
- Obstacles – Branches, relatively low ILP
- Scaling Trends : Distributed Servers

Benchmarks

- SPECweb96, SPECweb99, Webstone

File and Video Servers

- File servers:
  - Provide access to stored data over the network
  - Used in databases, web servers, mail servers, …
- Video server:
  - Special type of file servers where stored data is multimedia

Architectural Requirements – Highlights

- Storage
  - High volume
  - High bandwidth to stored data
  - Usually magnetic disks (RAID)

- Network
  - High throughput network connections

Computational Requirements

- Processing power required to executes different tasks:
  - Scheduling
  - Pre-fetching
  - Buffering
  - Data distribution (over storage resources)
  - Fault tolerance

Memory

- Memory mostly used for caching and buffering:
  - Small caches can catch large read traffic
  - Used as buffers
    - Buffering video streams
    - For pre-fetching
      - For larger files, pre-fetching shows lightly better performance

- Access patterns depends on application
  - Usually bimodal: files are mostly read or mostly written

Benchmarks

- Spec SFS
  - Synthetic benchmark
  - Measures throughput and response time
    - Generates and increases load and observes response time
  - Workload is consisted of different operations:
    - Look up, read, write, get attr., Read link, read dir, create, remove, FS stat, set attr,…

Benchmarks

- Postmark
  - Measures performance for mail and news servers
  - Different working set, a pool of files which are
    - Highly dynamic
    - Small in size
  - Workload:
    - Create or delete
    - Read or append

Video Servers

- Differences with file servers:
  - Access method is mostly sequential
  - Huge storage requirements
- Requirements:
  - Guarantee for timely delivering of data
  - Efficient utilization of storage capacity and bandwidth

Video Transfer

- Real time: data transmitted at the speed of stream requirements
  - Buffering is done in the server
  - Smooth traffic over the network
- Fast-load: larger block are read from storage and sent to client
  - Buffering is done in the network and client
  - Bursty traffic over the network

Some Performance Metrics

- Maximum number of video streams
- Average latency
- Jitter rate
  - Amount of discontinuity in video stream that is allowed by client
- Availability
- Unfairness

Enterprise Applications: Conclusions

- 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

OLTP References

- http://www.sei.cmu.edu/str/descriptions/tpmt.html
- http://www.subrahmanyam.com/articles/transactions/NutsAndBoltsOfTP.html
- Analysis of Commercial Workload on SMP Multiprocessors, X. Zhang, Z. Zhu, X. Du

OLAP/Mining References

- Lo et. al., An Analysis of Database Workload Performance on Simultaneous Multithreaded Processors, ISCA 1998, http://citeseer.nj.nec.com/lo98analysis.html
- Ailamaki et. al., Weaving Relations for Cache Performance, VLDB 2001, http://www-2.cs.cmu.edu/~natassa/dbarch.html
- Ailamaki et. al., DBMSs on a Modern Processor: Where Does Time Go, VLDB 1999, http://www-2.cs.cmu.edu/~natassa/dbarch.html
- Trancoso et. Al., The Memory Performance of DSS Commercial Workloads in Shared-Memory Multiprocessors, HPCA-3, 1997, http://citeseer.nj.nec.com/trancoso97memory.html
- Rao et. al., Cache Conscious Indexing for Decision-Support in Main Memory, VLDB journal, 1999, http://citeseer.nj.nec.com/rao98cache.html

Stream Database References

- Stanford Stream Data Manager Project page: http://www-db.stanford.edu/stream
- Carney et. al., **Monitoring Streams: A New Class of Data Management Applications**, VLDB 2002, http://www.cs.brown.edu/research/aurora/publications.html

WebServer References

- Workshop on Workload Characterization (WWC) 98, 99, 00, 01, 02
- Webserver Workload Characterization, John Diley, Hewlett Packard Labs
- Webserver Workload Characterization: The search for Invariants. M. Arlitt, C. Williamson. Measurement and Modelling of Computer Systems 96
- Performance Impact of Uncached File Accesses in SPECweb99. K. Kant, Y. Won, WWC 99
- Characterizing the behavior of Windows NT Server Workloads using Processor Performance Counters. R. Radhakrishnan, F. Rawson. WWC 98
- Performance Analysis of a WWW Server. V. Almeida, J. Almeida, C. Murta. CMG ’96

File/Video Server References

- Drew Roselli, Jacob R. Lorch, and Thomas E. Anderson, "A comparison of file system workloads", http://research.microsoft.com/~lorch/papers/fs-workloads/fs-workloads.html
- "Spec SFS r1 v3.0 documentation", http://www.specbench.org/sfs97r1/docs/usersguide.html
- Jeffrey Katcher, "postmark: A new file system benchmark", http://www.netapp.com/tech_library/3022.html
- Fouad A. Tobagi, James E. Long, "client-server challenges for digital video", 37th IEEE computer society international conference digest of papers, pp 24-28, February 1992
- C. Bernhardt, E. Biersack, "A scalable video server: architecture, design and implementation", real-time systems conference, Paris, France, January 1995
- C. Bernhardt, E. Biersack, "video server architectures: performance and scalability", 4th open workshop on high speed networks, Brest, September 1994