INF5070 – Media Storage and Distribution Systems:

Brute force vs. QoS

17/11 – 2003

Brute force

No brute force	Brute force
Think	Pay

LANs

- On-demand QoS
  - Asynchronous
    - all
  - Synchronous
    - FDDI, DQDB, ATM, Token Ring
    - Real-time ethernet
  - Isochronous
    - FDDI, DQDB, ATM
    - Real-time ethernet
- Non-QoS
  - Gigabit ethernet
  - Overprovisioning
  - Point-to-point
  - Switching

LANs

- On-demand QoS
  - Distinguish traffic types
    - Asynchronous
      - Normal traffic
    - Synchronous
      - Reserved
      - Bursty traffic
    - Isochronous
      - Reserved
      - Long sequences of short packets (telephony, …)
  - A&S
    - TokenRing, FDDI
  - A&S&I
    - FDDI-II, QPSX, DQDB, ATM
- Non-QoS
  - Ignore
    - Ethernet
  - Planning and setup-time decision
    - Real-time ethernet
    - Etherchannel
  - Use point-to-point connections

WANs

- On-demand QoS
  - RSVP
  - ATM
    - Directly
    - Setup e.g. with RSVP
  - MPLS
    - Setup e.g. with RSVP
  - DiffServ
    - By software selection that remains below planned-purchased resources
    - By adding RSVP in the access network
- Non-QoS
  - Don't rely on the net
    - Caching, pre-distribution, etc.
    - Prefetching
  - Overprovisioning
    - Dedicated line
      - Use DiffServ, ATM for reservation
      - Truly dedicated lines
    - Rely on ISPs to overprovision
  - Planning
    - DiffServ
    - ATM

WANs

- On-demand QoS
  - Idea
    - Reserve network and router resources
    - Short-term
    - On-demand
    - With exact traffic specification
  - Approaches
    - IntServ
    - ATM
  - Problems
    - Formulation of traffic specification
    - No consideration for cost
- Overprovisioning
  - Historical view
  - "Reservations about reservations" discussion
- Planning and setup
  - Idea
    - Reserve-by-phone
    - Reserve only aggregates
  - Approaches
    - DiffServ
    - MPLS
    - Actual use of ATM
  - Problems
    - Inflexible

Distribution systems

- Delayed on-demand delivery
- Prescheduled distribution
  - Possible with DVB
- Client-side caching
- All of them
  - Needs new receiver box
  - Must make more money
  - No business model
  - Must be able to receive 2 channels concurrently
  - Can add caching to TV sets
  - Use computers
  - External boxes exist (TiVo)
- No scheduling
  - Situation today

Distribution systems

- Prefix caching
  - Used to reduce jitter and startup latency
  - Works well with centralized control
  - Actually in use
- Optimized prefix caching
  - Can use resources more efficiently
- Gleaning
  - Can use regular clients with patching
- Periodic Multicasting with Pre-Storage
  - Can optimize delivery
- Nothing
  - Central control
  - Limited scalability
- Pre-distribution
  - Tight control
  - CDNs provide scalability
- Caching
  - Well-understood
  - Saves considerable
  - Simple to maintain and implement
- P2P Systems
  - Saves resources
  - Management problems

Distribution systems

- Prefix caching
  - is in use
  - This is a variation of predistribution
- Optimized prefix caching
- Gleaning
- Periodic Multicasting with Pre-Storage
  - Tried by CNN?
  - Similar to Real's cache for live content
- No scheduling
  - Caching can be attractive to ISPs to be preferrable over other ISPs
- No caching
- Predistribution
  - Content provider keeps control over the delivery

Distribution systems

- Delayed on-demand delivery
  - Allows TVoD
- Prescheduled distribution
  - Requires at least 2x client bandwidth
  - Reduced startup latency
- Client-side caching
  - Requires at least 2x client bandwidth
  - Requires client buffers
  - Allows TVoD
- All
  - Save server resources
  - Save bandwidth
  - Reduce startup latency
- Cable and satellite broadcast systems
  - NVoD is marketed now
  - Lack good programming anyway
  - Customer decisions are not desirable (TiVo)
- Internet
  - Requires multicast
    - Not supported by most ISPs
  - Content providers dislike client buffers
  - Jitter increases required buffer size
- Build bigger servers
- Limit the programming

OS Mechanisms Design: Memory

- Memory (buffer) caching
  - ☺ saves disk accesses (next client served from memory)
  - ☹ requires more memory
  - ☹ complex algorithms
  - ☹ often no gain due to high data rates
- Memory prefetching
  - ☺ saves disk accesses (large vs. many)
  - ☹ requires more memory
- Zero-copy
  - ☺ saves memory
  - ☺ saves CPU cycles
  - ☹ breaks OS layering
  - ☹ complex access to data (if at all)
- No memory caching
  - More disks
  - More disk controllers
  - More main memory / LRU
  - More or faster bus systems
  - (more machines)
- No memory prefetching
  - More disks
- No zero-copy
  - More memory
  - Faster CPU
  - More or faster system bus

OS Mechanisms Design: Disk

- Disk scheduling
  - time-based
    - ☺ timely delivery
    - ☹ low efficiency (much seeks)
  - seek-based
    - ☺ minimal disk arm movement
    - ☹ no provision of time or deadlines
    - ☹ disks hide layout (but usually a linear relationship with OS logical block number)
  - the disks themselves perform sorting based on placement
- No special disk scheduling
  - More disks
  - (More disk controllers)
  - More memory
    - But only
      - With high load
      - Accesses to same files
      - At slightly different times

OS Mechanisms Design: Disk

- Continuous block placement
  - ☺ may reduce arm movement by reading large data chunks
  - ☹ serve other requests as well
  - ☹ disks hide layout (but usually a linear relationship with OS logical block number)
  - ☹ requires much memory for efficiency
- No particular placement
  - More disks
  - Less memory
  - Buy RAM drives
    - 2GB - $3000 (in 2002)
  - No reasonable solution possible
    - To intra-request seeks

OS Mechanisms Design: CPU

- CPU scheduling
  - time-based
    - ☺ timely delivery
    - ☹ may be unfair
  - fairness-based
    - ☺ fair
    - ☹ no provision of time or deadlines
- No special CPU scheduling
  - More CPUs
  - ☺ CPU often not main bottleneck in optimized streaming servers

Combinations

- CPU scheduling and memory caching
- Disk scheduling and prefetching
- Disk scheduling and data placement
- Disk placement and zero-copy
- CPU scheduling and zero-copy
- CPU scheduling and disk scheduling

Combinations

- Prescheduled delivery and disk scheduling
  - Access patterns change, are not as expected
  - Prescheduled delivery allows good capacity planning
- Prescheduled delivery and network resource reservation
  - Prescheduled delivery allows capacity planning and needs no online decision making
  - Fits well with planning approaches, DiffServ
- P2P and disk scheduling
  - Current P2P systems are supposed to be network I/O-bound
  - Synchronized streaming from several clients increases jitter
- P2P and network resource reservation
  - Current P2P systems expect normal customers as peer nodes – no network resource reservation available
  - Distribution network using P2P technology may benefit