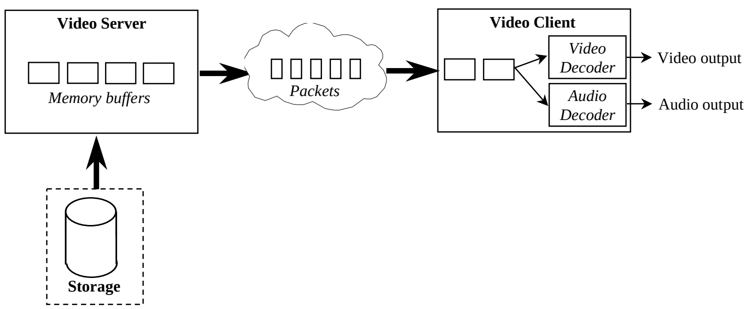
Distributed Video Systems
Chapter 7
Parallel Video Servers
Part 1 - Introduction and Overview

Jack Yiu-bun Lee
Department of Information Engineering
The Chinese University of Hong Kong

Contents

5.1 Introduction

- Primary Challenges in VoD System Design
  - High throughput and capacity ;
  - But low cost!
- Conventional VoD System
  - Video Server + Network + Video Clients



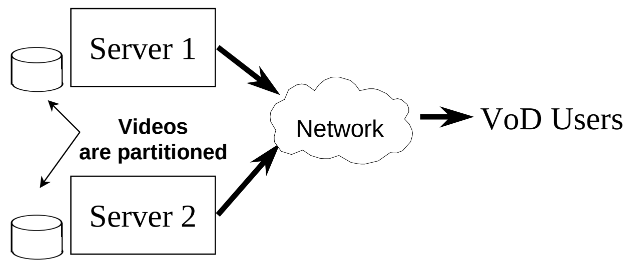
5.1 Introduction

- Bottlenecks at Video Server
  - Protocol and I/O processing
    - CPU time could be exhausted
  - Data retrievals
    - Disk bandwidth could be exhausted
  - Network transmissions
    - Network bandwidth could be exhausted
  - Others
    - System bus bandwidth could be exhausted
    - System's I/O interfaces could be exhausted
    - ...

- Traditional Approaches to More Capacity
  - Upgrade server with
    - Faster disk array with more disks
      - Leads to reliability problem
      - The disk-array controller becomes the bottleneck
    - Multiple disk-array controller
      - Number of expansion slots is limited
      - The system bus or the CPU becomes the bottleneck
    - Multiple faster CPU
      - The gain in multiprocessor system is sub-linear.
    - Faster network interface
      - Limited by number of expansion slots and system bus capacity
    - ...

- Examples
  - Small-scale systems (~100 streams)
    - Starlight Networks
      - PC-based, serves up to 100 users on a dedicated machine with a disk array and fast network connections.
    - Microsoft NetShow
      - Wintel-based, serves up to 60 users on a Wintel machine with a disk array and fast network connections.
  - Large-scale systems (~1000 streams)
    - The Magic Video-on-Demand System
      - Proprietary massively-parallel supercomputer with custom hardware and interconnection networks.
    - Oracle nCube Video-on-Demand System
      - nCube-based, massively-parallel supercomputer.

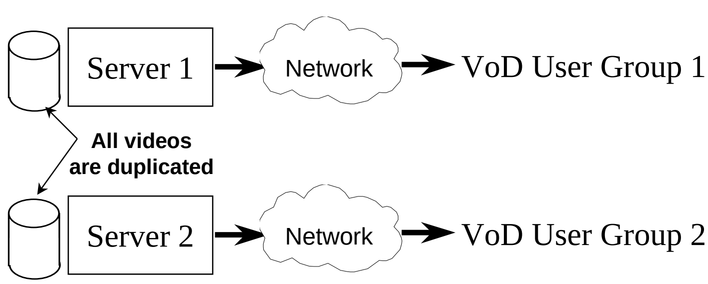
- Problems
  - Limited Scalability
    - How to support more than 1000 streams? 10K? 100K?
    - Partition:



    - Load balancing problem.

5.1 Introduction
Jack Y.B. Lee

- Problems
  - Limited Scalability
    - How to support more than 1000 streams? 10K? 100K?
    - Replication:



    - Cost-effectiveness problem.

- Problems
  - Upgrade Path
    - Single-server VoD systems
      - not incrementally upgradable;
      - requires replacement of hardware to upgrade;
      - less cost effective since existing hardware has to be discarded.
    - Partitioned VoD system
      - incrementally upgradable by adding more servers and repartitioning videos among them.
    - Replicated VoD system
      - incrementally upgradable by adding more servers and replicating videos among them.

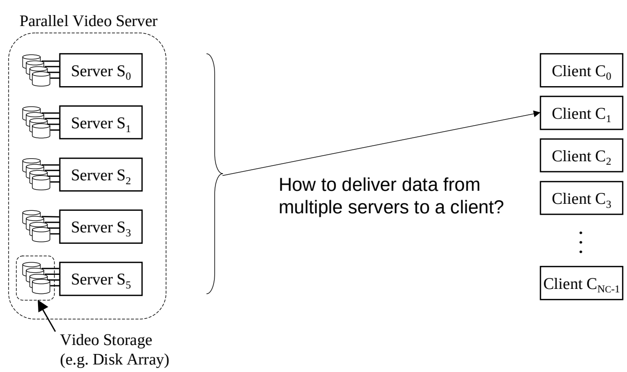
- Problems
  - Fault Tolerance
    - Single-server VoD systems
      - Can survive disk failures using RAID
      - Can survive power failures using UPS and redundant power supplies
      - Can survive memory failures using ECC memory
      - Very difficult to survive network failures
      - Impossible to survive server-level failures
    - Partitioned VoD system
      - Failures could be isolated, some video titles are affected and becomes unavailable.
    - Replicated VoD system
      - Failures could be isolated, some users are affected with service unavailability.

5.1 Introduction

- Motivation
  - The scalability and fault tolerant problems have been encountered before in
    - Disk Storage
      - Solution is disk array for scalability; and
      - RAID for both scalability and fault tolerance.
    - Tape Storage
      - Solution is tape arrays.
    - Network Communications
      - Solution is network striping.
    - So server arrays for VoD?

5.2 Video Distribution Architectures

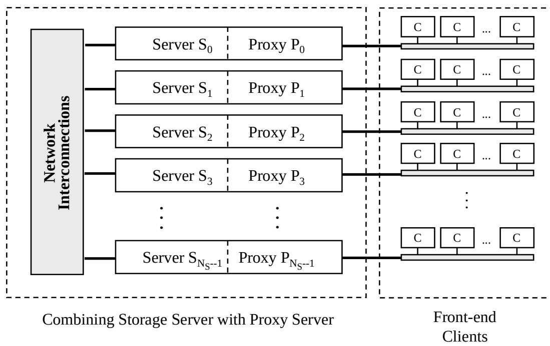
- Server-Level Striping and Video Playback



5.2 Video Distribution Architectures

Jack Y.B. Lee

- Proxy-At-Server Architecture
  - A proxy is used to retrieve data blocks from all servers and merges them for delivery to a video client.
  - Each server also runs a proxy process.

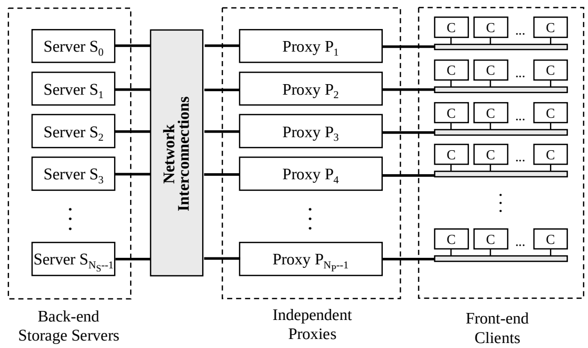


5.2 Video Distribution Architectures

Jack Y.B. Lee

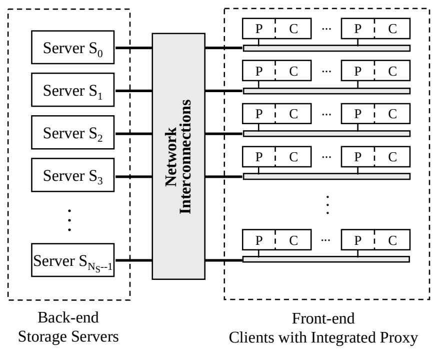
- Proxy-At-Server Architecture
  - Observations
    - A video client communicates with a specific proxy only.
    - No knowledge of the servers is required, hence transparent to the clients.
    - To deliver *B* bytes of data from servers to a client, on the average we need:
      - $B(2N_S-1)/N_S$ bytes of data transmission (server-to-proxy, proxy-to-client) and
      - $B(2N_S-1)/N_S$ bytes of data reception (proxy and client).
    - A server node failure will disrupt all clients connected at the proxy. Fault tolerance is only partial.

- Independent Proxy Architecture
  - The proxy runs at a separate node/host.



5.2 Video Distribution Architectures
Jack Y.B. Lee

- Proxy-At-Client Architecture
  - The proxy runs at client host.



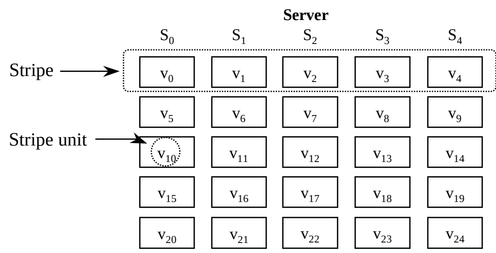
5.2 Video Distribution Architectures

- Proxy-At-Client Architecture
  - Observations
    - A proxy serves one client only.
    - Each client communicates with all servers directly.
    - The parallel servers are not transparent to the clients.
    - To deliver *B* bytes of data from servers to a client, we need:
      - *B* bytes of data transmission (server-to-client) and
      - *B* bytes of data reception (client).
    - A proxy failure affects one client only.
    - A server failure can be masked by redundancy. I.e. complete fault tolerance is possible.

5.3 Server Striping Policies

- Scope of Striping
  - Wide Striping
    - Stripe a video title over all servers in the system.
  - Short Striping
    - Stripe a video title over a subset of servers only.
- Striping Units
  - Time Striping
    - Stripe units are of the same duration in terms of playback.
  - Space Striping
    - Stripe units are of the same size.

- Time Striping
  - *k* frames per stripe unit. (Also called frame striping.)



v_i is stripe unit i, containing frames ki to $k(i+1)-1$

  - However *k* can be smaller than 1, i.e. use fragment of a frame as stripe unit. (Also called sub-frame striping.)

- Time Striping
  - Advantages
    - Scheduling may be simpler due to the constant-time nature of the stripe units.
    - May be easier to support interactive control such as fast-forward by frame skipping.
  - Disadvantages
    - Potential load imbalance among servers. For example, MPEG has I, P, B frames of generally different sizes.
    - Stripe using GOP can improve load balance.
    - More complicated storage and retrieval scheduling due to varying stripe unit size.
    - Note that sub-frame striping can achieve perfect load balancing using equal-sized frame fragments.

5.3 Server Striping Policies

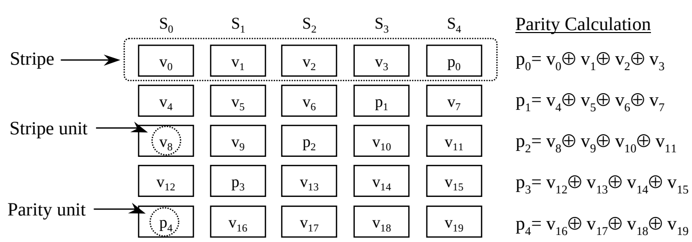
Jack Y.B. Lee

- Space Striping
  - Fixed-size striping units.
  - Advantages
    - Balanced storage;
    - Simplified retrieval scheduling;
    - Independent of the video compression formats.
  - Disadvantages
    - Variations in video block consumption time must be compensated by client buffering.

5.3 Server Striping Policies

Jack Y.B. Lee

- Data Redundancy
  - To sustain server-level failures, we need to introduce data redundancies among the servers.
  - Similar to RAID, we can use parity blocks to sustain single-server failure.



5.3 Server Striping Policies

Jack Y.B. Lee

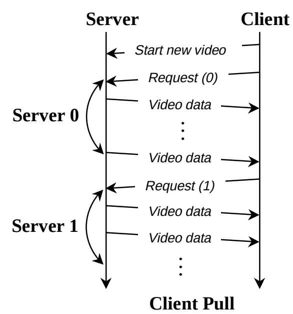
- Data Redundancy
  - Time Striping
    - Difficult to introduce data redundancy (unless it is sub-frame striping) as erasure-correction codes work on fixed-size parity groups only.
    - Error concealment could be applied to sub-frame striping as only a small fragment of each video frame will be lost.
  - Space Striping
    - Parity or RS-Code can be used to generate the redundant video blocks.
    - The recovery of lost video blocks must be performed at the proxy in real-time.

5.4 Video Delivery Protocols

Jack Y.B. Lee

- Service Models
  - Server Push
    - The servers schedule transmissions to a client.
    - Problem
      - If multiple servers transmit to the same client at the same time, congestion will occur, leading to packet losses.
    - Solution
      - Some form of co-ordination (i.e. synchronization) between the server must be performed to avoid congestion.
    - Additional Problems
      - The synchronization protocol must be scalable;
      - and tolerance to node failures.

- Service Models
  - Client Pull



- No need for server synchronization;
- Truly autonomous servers;
- Simpler server design.

- Detecting and Masking Server Failures
  - Problem
    - Given there are redundant data at the servers, how do we deliver these redundant data to the client for recovery in case a server failure occurs?
  - Solution 1: Forward Error Correction (FEC)
    - Retrieve and transmit redundant data all the time.
    - Constant bandwidth overhead of $K/(N_S-K)$ where K is the number of redundant video blocks per parity group in a N_S-servers system.
    - No failure detection is necessary, the redundant data will be ready at the client when a server fails.

5.4 Video Delivery Protocols

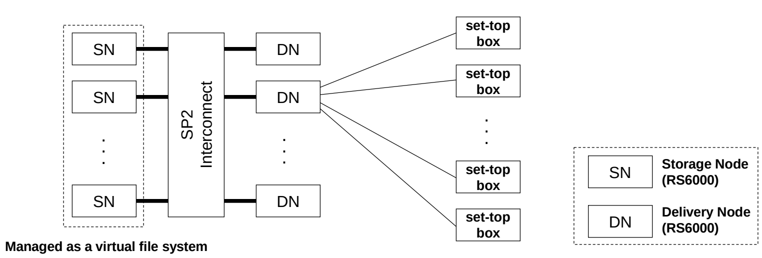
- Detecting and Masking Server Failures
  - Solution 2: On-Demand Correction (ODC)
    - Retrieve and transmit redundant data only after a server failure is detected.
    - No overhead when there is no server failure.
    - Even after a server failed, the total bandwidth requirement remains the same. (Why?)
    - Server failure detection is required, though.
    - Additional client buffering will be required to sustain continuous video playback while the system reconfigures itself for failure-mode operation.

5.5 Representative Studies

- SPIFFI (Freedman et al. 1995)
  - Architectural Highlights
    - Proxy-At-Client
    - Space Striping
    - Client-Pull with Predictive Prefetch
  - Methodology
    - Performance evaluation using simulation.
    - Studies real-time disk scheduling, predictive prefetch algorithms, and server buffer pool management (caching using various cache replacement algorithms).
    - Provides statistical performance guarantees.

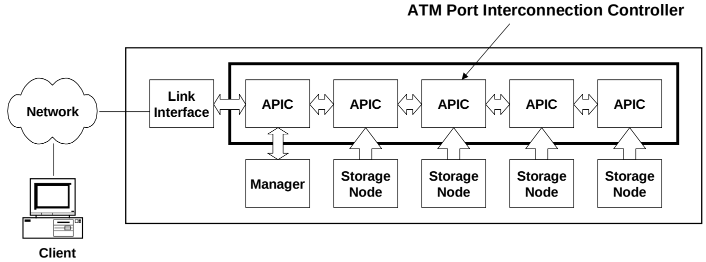
- SPIFFI (Freedman et al. 1995)
  - Major Results
    - Optimal striping size ~ 512KB.
    - Server buffer requirement 128MB~2GB.
    - A 4-servers, 64-disks system can support 760 4Mbps streams.
    - No implementation.

- Clustered Video Server (Tewari et al. 1995)
  - Architectural Highlights
    - Proxy-At-Server (Flat) and Independent Proxy (Two Tiered)
    - Space Striping
    - Client-Pull (Server-to-Proxy) Server-Push (Proxy-to-Client)



- Clustered Video Server (Tewari et al. 1995)
  - Methodology
    - Performance analysis using queueing theory and simulation.
    - Studies effect of striping size, proxy buffer requirement, and system scalability.
    - Provides statistical performance guarantees.
    - The proposed architecture has been implemented using a cluster of RS6000 workstations.
  - Key Results
    - Optimal striping size ~256KB.
    - Near-linear scalability (90% at 128 nodes).
    - A delivery node can support ~50 MPEG2 streams.

- MARS (Buddhikot et al. 1995)
  - Architectural Highlights
    - Independent Proxy Using a Proprietary ATM Switch
    - Time Striping (Frame Striping)
    - Server Push
    - Closely-coupled, clock-synchronized



5.5 Representative Studies

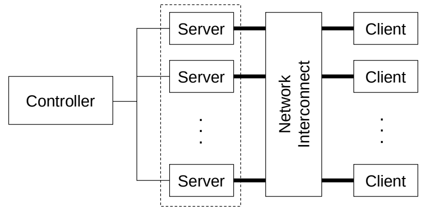
- MARS (Buddhikot et al. 1995)
  - Methodology
    - Performance analysis using worst-case analysis to provide deterministic performance guarantees.
    - Studies data layout policy, scheduling at the servers and the custom ATM switch, and playout control to support VCR-like interactions.
    - The proposed system has been implemented.
  - Key Results
    - Designs for a closely-coupled parallel video server.
    - Proved conditions to maintain load balance in normal and FF, RW playback.
    - No benchmark results are given.

5.5 Representative Studies

Jack Y.B. Lee

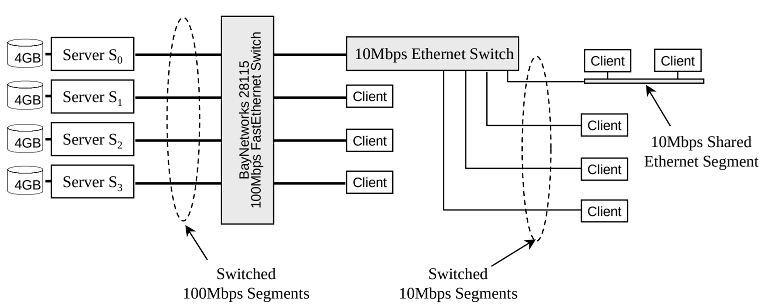
- Microsoft NetShow Theater (Bolosky et al. 1996)
  - Architectural Highlights
    - Proxy-At-Client
    - Space Striping
    - Server Push
    - Fault Tolerance via Mirroring with Declustering
  - Methodology
    - Performance evaluation using experimentation and benchmarking.
    - Studies system capacity, inter-server scheduling, data placement policy for mirroring, and fault-detection protocol.
    - The proposed design has been implemented and is available commercially.

- Microsoft NetShow Theater (Bolosky et al. 1996)



- Key Results
  - Implementation runs on Windows, delivers video over UDP.
  - Block size from 64KB ~ 1MB
  - A system with 5 servers, 3 disks/server, and OC-3 ATM card can support 68 6Mbps streams.
  - Can tolerate single-server failure using mirroring and 20% reserve in capacity.

- Server Array and RAIS (Lee et al. 1996)
  - Architectural Highlights
    - Proxy-At-Client, Space Striping, and Client Pull
    - Fault Tolerance by
      Redundant Array of Inexpensive Servers (RAIS)



- Server Array and RAIS (Lee et al. 1996)
  - Methodology
    - Performance analysis using worst-case analysis to provide deterministic guarantees.
    - Performance evaluation through experimentation and benchmarking.
    - Studies system capacity, scalability, striping and placement policy, fault tolerance algorithms, etc.
    - The proposed designs have been implemented and is available commercially.

- Server Array and RAIS (Lee et al. 1996)
  - Key Results
    - Experimental and Benchmarks
      - Linear capacity scaling from 1 to 4 servers.
      - A PC server can support ~50 MPEG1 video streams.
      - Server memory requirement (incl. OS and everything) is 64MB.
      - Client buffer requirement is <1MB.
      - Fault tolerant to single-server failure.
    - Theoretical
      - System capacity is linearly scalable with the help of admission scheduling.
      - Server and client buffer requirement is fixed irrespective of scale of the system (i.e. number of servers).

5.5 Representative Studies

- Comparisons

Researchers	Video Distribution Architecture	Server Striping Policy	Video Delivery Protocol	Server Fault Tolerance
Biersack et al. (Video Server Array)	Proxy-At-Client	Time Striping	Server Push	Striping w/ Parity; FEC
Bolosky et al. (Tiger Video Fileserver)	Proxy-At-Client	Space Striping	Server Push	Mirroring with Declustering
Buddhikot et al. (MARS)	Independent Proxy	Time Striping	Server Push	–
Freedman et al. (SPIFFI)	Proxy-At-Client	Space Striping	–	–
Lee et al. (Server Array & RAIS)	Proxy-At-Client	Space Striping	Client Pull	Striping w/ Parity; FEC and ODC
Lougher et al.	Independent Proxy	Space Striping	–	–
Reddy et al.	Proxy-At-Server, Independent Proxy	Space Striping	Server Push	–
Tewari et al. (Clustered Video Server)	Proxy-At-Server, Independent Proxy	Space Striping	Server Push	–
Wu and Shu	Proxy-At-Server, Independent Proxy	Space Striping & Time Striping	Server Push	–

References

This chapter's materials are based on:

[1] Jack Y.B.Lee, "Parallel Video Servers - A Tutorial," *IEEE Multimedia*, vol.5(2), June 1998, pp.20-28.

Other useful references:

[2] C. Bernhardt, and E. Biersack, "The Server Array: A Scalable Video Server Architecture," *High-Speed Networks for Multimedia Applications*, Kluwer Press, Boston, 1996.

[3] W.J. Bolosky, J.S. Barrera, III, R.P. Draves, R.P. Fitzgerald, G.A. Gibson, M.B. Jones, S.P. Levi, N.P. Myhrvold, R.F. Rashid, "The Tiger Video Fileserver," *Proc. of the Sixth International Workshop on Network and Operating System Support for Digital Audio and Video*. IEEE Computer Society, Zushi, Japan, April 1996.

[4] M.M. Buddhikot, and G.M. Parulkar, "Efficient Data Layout, Scheduling and Playout Control in MARS," *Proc. NOSSDAV'95*, Springer-Verlag, Berlin, 1995, pp.318-329.

[5] C.S. Freedman, and D.J. DeWitt, "The SPIFFI Scalable Video-on-Demand System," *Proc. ACM SIGMOD'95*, ACM Press, New York, June 1995, pp.352-363.

[6] Y.B.Lee, and P.C.Wong, "A Server Array Approach for Video-on-demand Service on Local Area Networks," *Proc. IEEE INFOCOM '96*, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp.27-34.

[7] P.C.Wong, and Y.B.Lee, "Redundant Array of Inexpensive Servers (RAIS) for On-Demand Multimedia Services," *Proc. ICC'97*, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp.787-792.

[8] P. Lougher, D. Pegler, D. Shepherd, "Scalable Storage Servers for Digital Audio and Video," *Proc. IEE International Conference on Storage and Recording Systems 1994*, IEE Press, London, 1994, pp.140-3.

[9] R. Tewari, R. Mukherjee, and D.M. Dias, "Real-Time Issues for Clustered Multimedia Servers," *IBM Research Report RC20020*, June 1995.

[10] M. Wu, and W. Shu, "Scheduling for Large-Scale Parallel Video Servers," *Proc. Sixth Symposium on the Frontiers of Massively Parallel Computation*, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp.126-133.

[11] C. Brendan, S. Traw, and J.M. Smith, "Striping Within the Network Subsystem," *IEEE Network*, July/August 1995, IEEE Press, New York, NY, 1995, pp.22-32.

[12] A.L. Drapeau, and R.H. Katz, "Striped Tape Arrays," *Proc. 12th IEEE Symposium on Mass Storage Systems*, IEEE Press, 1993, pp.257-65.