The Personal Terabyte

Ann L. Chervenak
College of Computing, Georgia Tech
Motivation

Magnetic disks: massive, inexpensive storage

• Capacity increases 60% per year
• 5 to 8 years: Terabyte on a disk
• 10 years: $300 buys a terabyte

A “Personal Terabyte” for the home user

How to manage and exploit the Personal Terabyte?
Outline

• The Home Environment and Workload

• Overview of Research Issues
  • Prefetching from the Personal Terabyte disk
  • Prefetching from the World Wide Web
  • Backup and Reliability
  • Disk System Architecture and File System Issues

• Summary
The Home Environment

- Multiple Network Connections to Outside World
  - High-bandwidth, low-cost broadcast
    (cable, satellite)  Gbits/sec
  - Lower-bandwidth, higher-cost point-to-point
    (wired and wireless)  Tens of Mbits/sec
The Home Environment

• Network within the home
  • Connect appliances, security system, etc.
  • Ethernet, Firewire, CEBus

• Compute engines, displays

• The Personal Terabyte
  • Single disk or disk array?
  • Central server or distributed?
  • Traditional or new disk architecture?
The Home Workload

Prefetch, cache and pre-process WWW data
Archive of personal and family data
  • Home movies, photographs
Storage, playback of entertainment video
Information databases
Games, virtual environments
Work: Simulations, large data sets
Security system: video monitoring of children, pets
Word processors, spreadsheets, etc.
Outline

• The Home Environment and Workload

• Overview of Research Issues
  • Prefetching from the Personal Terabyte disk
  • Prefetching from the World Wide Web
  • Backup and Reliability
  • Disk System Architecture and File System Issues

• Summary
Prefetching from the Personal Terabyte Disk

Increasing gap between CPU and disk speeds
CPU: 60%/year  Disk: 10%/year

Page Faults Take Millions of Clock Cycles:
Prefetch disk pages into memory

Techniques: Disk readahead or application or Operating system prediction
Prefetching From Disk to Main Memory

Two main prefetching approaches:

1. Applications provide deterministic "hints" of what blocks they will access
   - Decide whether to prefetch a block based on cost-benefit analysis
   - Hugo Patterson (CMU)

2. Predict future accesses based on past accesses
   - Probabilistic hints or predictions
   - Probability trees (Duke, Kentucky), Markov models (Illinois)
Our Approach: a Hybrid Scheme

*What to prefetch:* predict based on past  
*Whether to prefetch:* use cost/benefit analysis

Deciding What to Prefetch

- Algorithm from Duke University:  
  Probability tree updated on every access

![Probability tree graphic]

Prefetch candidates:  
high probability of being accessed
Deciding Whether to Prefetch: Cost-Benefit Analysis

Adapted from Hugo Patterson’s Informed Prefetching

Must calculate:

- **Benefit of allocating a cache buffer** to prefetch an additional block

- **Cost to reclaim a buffer** from the *demand cache* or *prefetch cache* to hold the prefetched block

Prefetch block only when *benefit exceeds cost*
Prefetching and Caching World Wide Web Data

Prefetch/cache a subset of WWW
  • Avoid network and server delays

Cache general interest data from broadcasts
  • Filter data based on user interests
  • Even Personal Terabyte can’t store everything

Prefetch data for specific interests over Internet
  • Generate network traffic: responsible prefetching
Prefetching from World Wide Web

WebSnatcher

- Customized prefetching of WWW data
- Periodically prefetch according to user profile
- Store results on local disk
- Avoid delays from network and server loads

Profile includes:

- List of servers that are “functionally equivalent”
- Used for anycasting
Anycasting: Using Experience to Guide Server Selection

With Prof. Ellen Zegura
“Anycasting”: specify a service, can go to any one of a collection of servers

Importance: Responsible prefetching, Performance
Resolver Selection Algorithms

**Average:** Choose server with best mean performance

**Moving Weighted Average:** Choose server with smallest moving weighted average (Weigh recent history more heavily)

- $D_{0,j} = X_{0,j}$
- $D_{i,j} = \alpha X_{i,j} + (1-\alpha)D_{i-1,j}$

**Minimum:** Visit each server periodically, choose server with minimum response times

**Hop Count:** Choose server with smallest hop count from client machine to server (use traceroute)

**Round Robin:** Select servers in round robin fashion
Evaluating Resolver Algorithms

Experiments with four anycast groups:
News stories, Seattle weather forecast,
“Today in history”, Leo Horoscope

Average and moving weighted avg. close to optimal

Hop count, round robin perform poorly
Don’t use past experience

Importance:
• Automatically generate network traffic:
  must be responsible
  • Choose server with quick response to reduce network and server load
• Improve performance for interactive applications
Protecting the Reliability of Personal Terabyte Data

Disk Array Techniques

| D1 | D2 | D3 | D4 | Parity |

• Protect against individual component failures
• Will consumers buy extra disks for reliability?

Still need recovery from disasters, recovery of accidentally-deleted files

Traditional full backups will take longer
• Capacity increases 60% per year
• Transfer rate 40% per year
Desirable Backup Features for the Personal Terabyte

**Incremental-only** backup schemes
- Write file when it is created
- Then only write incremental changes

**Snapshots** and **copy-on-write**
- On-line backup, save old versions of files

**Selective backup, compression**

**Automated network backup** off-site
- Few backup home data

Measuring **College of Computing backup system**, evaluating incremental-only algorithm
Personal Terabyte Storage Architecture

Centralized or distributed server

- Information furnace
- One disk or an array

Network-Attached Disks

- Data need not pass through host

Active or Intelligent Disks

- Partition applications, run part on disk’s CPU
- Home applications:
  - Optimize disk layout for backup
  - Background reorganization of data
  - Processing, delivering multimedia data
File System Organization

Must support:
  • Large files
  • Large numbers of files
  • Efficient storage and retrieval

Block sizes
  • Fixed blocks, multiple block sizes or extents

Metadata
  • Inodes
  • Limited levels of indirection (Frangipani)
  • B-Trees (XFS)
Summary: Managing and Exploiting the Personal Terabyte

• Prefetching from the Personal Terabyte disk
  • Cost-benefit analysis, predictive prefetching

• Prefetching from the World Wide Web
  • WebSnatcher, Anycasting paradigm

• Backup and Reliability
  • Experiments with incremental-only; snapshots

• Disk System Architecture and File System Issues
  • NASD, active disks; Data and metadata layout