CSci 5980 Advanced Storage Systems

Spring 2017

David Hung-Chang Du

Qwest Chair Professor
Computer Science and Engineering
University of Minnesota
du@cs.umn.edu

CRIS: NSF I/UCRC Center on Intelligent Storage
Class Emphases

• Reading + Discussing
• Thinking Skill
• Building Research Experiences
• Class Interactions
• Research Report
Research Basics

• Technology Trend
• Application Trend
• Systems Trend
• Understand the overall picture
• Identify new issues with sound motivations
• Formulate the problem and propose solutions
• Verify or validate your findings
Instrument and Connect the World!

Data Are Continuously Generated with Coordinates and Time

Bridge Monitoring
Building
  Environment
  Controls
Earthquake
  Monitoring
Elder Care
Factories
Fire Response
First Responders
Forest Management
Soil Monitoring
Supply Chain
Wind Response
... and more more
Digital Explosion: Data Centric

- The digital universe will grow over six-fold, from 281 exabytes in 2007 to 1,773 exabytes in 2011.

- > 90% of the information in the digital universe is unstructured and absolute # of files growing faster than the TBs.

----from IDC Survey presented in ISW 2008
We Are Collecting All Kinds of Information

- Indoor/outdoor temperature sensors
- Oil sensor
- Water coolant temperature
- Oxygen sensor
- Accelerometer
- Passenger Occupancy
- Fuel level
- Seat belt tension
- Rain sensor
- Parking sensor
- Radar sensor
- Anti thief sensors
- Speedometer Tachometer Odometer
- Wheel speed
- Tire pressure monitor
- GPS
- Water coolant temperature
- We Are Collecting All Kinds of Information
Rapid Evolution of Mobile Devices and Applications (More Data)
More Examples

1. Surveillance video data collected by Air Force drones
   - One year’s collection requires 24 years’ processing (NYT, January 10, 2010)

2. Analog TV broadcasting becomes digital

3. Surveillance video cameras are widely deployed
Research Methodologies Are Rapidly Changing

ITS modeller: A modelling environment for Intelligent Transport Systems, including co-operative systems
NASA Satellite and In Situ Observations

SRTM

LANDSAT

MODIS

ALOS PALSAR

ICESat

USDA FS, FIA Data
**Terrestrial Observation and prediction system (TOPS)**

A data-modeling system for integrating satellite, surface data with simulation models to produce ecological nowcasts and forecasts.

**Key elements:**

- Collect & Extract Data
- Monitoring
- Modeling
- Forecasting
- Local to Global

Focus on biogeo-chemical cycles

Nemani et al., 2009, RSE
Information Based Medicine will require unprecedented access to diverse, integrated information

Challenges
• Volume and complexity of data
• Integrating massive volumes of disparate data
• Need for sophisticated analytics
• Growing collaboration across ecosystem

Access to Diverse Heterogeneous Distributed Data

1. Patient Information

Expression Arrays (various tissues)

Personal genomics

Analysis lab notes

Clinical Record

Hospital events ....admission, surgery, recovery, discharge

X-rays, MRI, mamograms, etc
IBM Healthcare and Life Sciences Clinical Genomics Solution Conceptual Architecture

Medical Research
- Expression, SNPs, Clinical Studies & Trials, Proteomic

Clinical Care
- HIS, RIS, CIS, Pathology, Rx, Patient Charts

Adherence to Standards
- HL7, BSML/HapMap, CDISC/ODM, MAGE-ML, CDA, etc.

Medical Information Gateway
- Deidentification of Patient Data & Anonymous Global Patient Identifier Assigned

Medical Information Broker

Data Mining/Statistical Analysis/Visualization

WebSphere

DB2 Information Integrator

Medical Information Repository
- Source scientific data & unstructured text files e.g. MS Access, MS Excel, EST/ GenBank, XML, Medline, dbSNP

University of Minnesota  Digital Technology Center  Intelligent Storage Consortium
Challenges in the New Environment

• Network is no longer just end-to-end
• Data is no longer just structured
• Drowning in Data; Loss in Internet
• A New Internet Architecture Is Required
• Data -> Information -> Knowledge
• Data/Network Security
• Data/Information Privacy
• Long-Term Data Preservation
• Scalability (Internet, cloud storage and exascale computing)
More Challenges

• How to model dynamically changed data relationship?
  – Relationship can be changed by an event
  – Relationship can be changed by situations
  – Relationship can be changed by interests

• How to make network to be situational-aware?

• How to integrate data delivery with dynamically changed situations/interests/events?

• How to decide data to be sent (to whom?), to be stored (for how long?) and to be dropped?
Why VIA – Hardware structure
A Case for Data and Control Flow between Host and NIC
Intelligent Storage
based on Object-based Storage Devices (OSD)

Captured in the attributes of an object → Exploited to store and retrieve data more efficiently with Indexing/Search capability

[ INTELLIGENCE ]

← Extended attributes augmented view high level semantics associated.

← Traditional storage device view - raw bits, no associated semantics.
Intelligence on the Drives

- Magnetic storage / Solid State Storage
- Processor & DRAM
- SAN/IP attachment
- Execution environment
- Semantic-aware
- Application-aware
- Object storage device
- Search and indexing
- Query and response

Data+Data Semantic, Data Location Transparency, Storage Layer Reduction
Evolution of Data Storage

Direct Attached Storage

SAN Attached Storage

SAN Attached Storage

Network Attached Storage

Object-based Storage Devices (OSD)

Intelligent Storage Devices (ISD)
Overview of the OSD Architecture

Clients → Access Request → Object-based Storage Devices

Data

MANAGEMENT

SECRET KEY

SECRET KEY

Validate Capability

Overview of the OSD Architecture

Managers

Network
OSD - Phase I Architecture

Initiator

- Application
- OSD Commands
- iSCSI Initiator Driver

IP Network - SCSI Commands

OSD Target

- iSCSI Target Driver
- OSD Cmd Emulator
- Local File System

Linux-based Host

- Local OS
- Disk

Local OS

Security Agent

MySQL

ACL

Data

Security manager

Policy Manager

MySQL

ACL

Data

Security manager

Policy Manager
DISC OSD Reference Architecture

OSD Client
- File System Clients
- OSD API / I/O Path

Meta Data Server
- Meta Data Manager
- Security

OSD Initiator
- Intel iSCSI Initiator / Driver
- Linux TCP/IP Stack

OSD Target - Phase I
- Linux TCP/IP Stack
- Intel iSCSI Target
- T10 OSD Command Parser
- Simple OSD
- Linux File System
- Linux Disk Driver

OSD Target – Phase II / III
- OSD File System / Command Processor

Future Global File System Architecture

Clients can directly access the objects after receiving credentials from the regional manager.

P2P interaction
Our View of Intelligent Storage

• Storage that:

  – Is *aware* of the data *objects* that it stores
  – Can *interpret* the *attributes* of the data *objects*
  – Can independently *manipulate* the *objects*
  – Can independently build a search and indexing mechanism based on the *objects contents*
  – Can be (dynamically) *taught* new methods for understanding and manipulating objects, attributes and their content
Active Data Objects

- Problem: How can the OSD environment be extended to be even more flexible?
- Approach: Allow objects to include executable methods in addition to the data, attributes and metadata. These methods can be invoked when a pre-set condition is met.
- Uniqueness: Data objects are truly autonomic. Intelligent storage devices have to designed to provide such a capability.
OSD Implementation

System Configuration

- 1 file client: Blade server node (linux)
- 2 or more OST: Blade node (linux)
- 1 MDS: Blade node (linux)

- Integrates method API into Lustre file system
- Develops new distributed grep program
Semantic Gulf

• Data is organized based on data characteristics
  – Efficiency and scalability are primary
• User intentions and context are not considered
  – Query environment does not maintain the state of the user

Possible Solutions

Keywords, Ontology, Query by Example, Relevance Feedback, Event-based Indexing
Two-Level Search & Indexing

1. **Client**
   - Content-based query
   - Authenticate client

2. **Home Region**
   - IP Network
   - Meta Data Server
   - SIE (High-level Searching & Indexing Engine)
   - SIE (Low-level Searching & Indexing Engine)
   - Response

3. **Meta Data Server**
   - Content-based query

4. **IP Network**
   - SIE
   - Content-based query

5. **Region**
   - GUID
   - SIE
   - Content-based query

6. **Response**
   - 6"
Extended Attributes (EAs) in Intelligent Storage

• Motivations
  – Attributes are important extension of objects to traditional files
  – EAs can support semantic-aware and application-aware storage
  – Preliminary solutions in existing file systems: Ext2, Ext3, XFS, JFS, ReiserFS, etc.

• Objective
  – How to use EAs and how to efficiently store and access EAs?
    • Access control of EAs
    • Fast retrieval of any EA by name
    • Fast bulk copy of EAs
    • Efficient space utilization for variable-sized EAs
    • Help on search and indexing
Storing EA on Storage Media

- Inline (within i-node) space for small EA
- External pages for more EAs
- Differentiate per-instance EAs (IEAs) and per-object EAs
  - Example: When copying, “audit trace” EA should not be copied to the new object
- Separated EA headers and EA values
  - Headers are organized into index for fast lookup
  - Headers have inline space for storage small values
  - EA value pages support efficient variable-sized EA values
QoS-based Scheduling:
• dynamically requests breakdown based on current load, buffer space
• storage and network BW monitoring
• Schedule based on urgency and expected access time

Assumptions:
• dynamic network BW
• adjustable data layout

QoS Specification
• Object inherent QoS
• Operation level QoS
• Session level QoS

Integrating network QoS with Storage QoS; Shorten delay with RDMA

Server
Computer
Laptop

Clients
Network Cloud
Data Center

Gig E
IB
FC
Y1  

#\ljslqj/#;2625338
Automatic Configuration of RAID Arrays

- Problem: Designing storage subsystem → very complex
  - Huge search space
  - Complex tradeoffs [stripe size, stripe width, RAID level]
  - More complicated if objects have QoS requirements.

- Solution: Autonomous configuration
  - QoS attributes → capture the requirements of the object.
  - Algorithms to efficiently explore the search space.
  - Automatically choosing RAID level mapping.
  - Assigning objects to physical RAID devices such that QoS requirements are met.
Secure File Sharing with Long Term Key Management

• Secure and scalable file sharing
• End-to-end data security
  – Confidentiality
  – Integrity
• Efficient key revocation
Secure File Sharing with Long Term Key Management

- Long term key management
  - Secure keys → secure data
- Secure key backups and retrieval
- Efficient key recovery
Facts in New Environment

• Extremely large Scale Internet reaches every corner of the earth
• Support massive mobility
• Large number of smart embedded sensors are in our daily objects
• Each sensor can compute, communicate and sense
• They form the largest network ever and generate huge amount of data
• Each sensor generates data continuously with two important dimensions: time and location coordinate
• We are drowning in data and loss in Internet

A new Internet architecture and new ways to handle data are required
More Future Trends

• Embedded processors with great processing power and big storage capacity will be deployed all over the places

• Computers start to show intelligence over human in certain aspects (Considering Watson the computer becomes the winner of Jeopardy and AlphaGo defeats legendary Go player)

• Huge amount of useful data are available and more are generated from every corner of the earth (Data Collection, Extraction, Transformation, and Analytic Are Require)
Storage Research Trends

• New Storage Devices (Active Devices): Small & Big Capacity with Various Performance, Longer Lasting Duration, Better Support to Applications (Fewer Software Layers, but More Varieties of Storage Devices)

• Emerging Storage Hierarchies (Multi-Tier): Cloud Storage (OpenStack Based?), Key-Value Stores, Tiered Storage (not really tiered), Extremely High Performance, More Integrated with File Systems/Applications, Software Defined Storage, Software Defined Data Center, Global Storage

• New Applications (Analytics & Decision Making): Data Analytics, Archive vs. Backup
Challenges of Data Management and Data Analytics

• Dealing with highly distributed data sources & types
• Tracking data provenance, from data generation to data preparation
• Validating data
• Working with different data formats and structures
• Ensuring data integrity, security, privacy and sharing
• Developing methods for visualization massively data
• Coping with real-time analysis and decision making
Solutions?

• New Internet Infrastructure (fully integrating networking with storage: SDN+SDS?)

• Intelligent Storage (object-based? Active devices?)

• Store Data, Meta-Data, and Semantic Together (extended attributes? Adaptive key-value store?)

• Merging HPC and Cloud in Data Centers (performance+capacity)

• Global Storage and File Systems (Internet Scale?)
4 Vs of Big Data

**Volume**

- **40 ZETabytes** (40,000,000,000,000,000,000 bytes) of data will be created by 2020, an increase of 300 times from 2005.
- **6 Billion People** have cell phones.
- **World Population: 7 Billion**

**Value**

- **2.5 Quintillion Bytes** of data are created each day.
- **100 Terabytes** of data are stored.
- Most companies in the U.S. have at least **100 Sensors** that monitor items such as fuel level and tire pressure.

**Velocity**

- The New York Stock Exchange captures **1 TB of Trade Information** during each trading session.
- Modern cars have close to 100 SENSORS that monitor items such as fuel level and tire pressure.
- By 2016, it is projected there will be **16.9 Billion Network Connections** – almost 2.5 connections per person on earth.

**Variety**

- **30 Billion Pieces of Content** are shared on Facebook every month.
- **4 Billion+ Hours of Video** are watched on YouTube each month.
- **400 Million Tweets** are sent per day by about 200 million monthly active users.

**Veracity**

- **1 in 3 Business Leaders** don’t trust the information they use to make decisions.
- **27% of Respondents** in one survey were unsure of how much of their data was inaccurate.

**The FOUR V’s of Big Data**

- From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As of 2011, the global size of data in healthcare was estimated to be **150 Exabytes** (1.5 petabytes).

By 2014, it’s anticipated there will be **420 Million Wearable, Wireless Health Monitors**.

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EVC, SAS, IBM, IDEO, QS.
Current Effort on Big Data

• Flexible & Fast Processing: In Memory Processing
• Multiple Data Sources: Data Steaming
• Decision Making vs. Querying: Machine Learning Approach
  • Identifying Useful Information on Storage: active storage devices: exhaustive search?
  • Long-Term Data Storage: New Archive Architecture?
The Data Centric Future of IT

• What will it look like?
The Changing Storage Landscape:

- Services
- Software
- Systems
- Devices
- Components

- Big Data
- Cloud
- Software Defined Storage
- Open Source

Source: Seagate presentation
Expands I/O Stack to Fit Various Needs

1-2 PB/sec
Residence – hours
Overwritten – continuous

4-6 TB/sec
Residence – hours
Overwritten – hours

1-2 TB/sec
Residence – days/weeks
Flushed – weeks

100-300 GB/sec
Residence – months-year
Flushed – months-year

10s GB/sec (parallel tape)
Residence – forever

1-2 TB/sec
Residence – days/weeks
Flushed – weeks

100-300 GB/sec
Residence – months-year
Flushed – months-year

Parallel File System
80-100PB

High Performance Storage
300-500PB

Tape

Archive
10EB

Host based PCIe cards moving to Shared NVME

Host server based memory architectures- DRAM, 3D Xpoint

PFS based systems with high performance SAS disk

Object based high capacity systems/disks (SATA) – Active Archive

New TAM Expansion Opportunity
New Memory/Storage Technologies
New Storage Technologies

Flash Memory based SSD
FTL Design

PCM Prototype

Shingled Write Disk Design and Layout
## Summary of Memory Technologies

<table>
<thead>
<tr>
<th></th>
<th>HDD</th>
<th>DRAM DIMM</th>
<th>Flash SSD</th>
<th>PCM (25nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Density (μm²/bit)</strong></td>
<td>0.00006</td>
<td>0.00380</td>
<td>0.00210</td>
<td>0.00250</td>
</tr>
<tr>
<td><strong>Read Latency (ns)</strong></td>
<td>3,000,000</td>
<td>55</td>
<td>25,000</td>
<td>48</td>
</tr>
<tr>
<td><strong>Write Latency (ns)</strong></td>
<td>3,000,000</td>
<td>55</td>
<td>200,000</td>
<td>150</td>
</tr>
<tr>
<td><strong>Read Energy (pJ/bit)</strong></td>
<td>2,500</td>
<td>12.5</td>
<td>250</td>
<td>2</td>
</tr>
<tr>
<td><strong>Write Energy (pJ/bit)</strong></td>
<td>2,500</td>
<td>12.5</td>
<td>250</td>
<td>19.2</td>
</tr>
<tr>
<td><strong>Static Power</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Endurance</strong></td>
<td>&gt;10^{15}</td>
<td>&gt;10^{15}</td>
<td>10^{4}</td>
<td>10^{8}</td>
</tr>
<tr>
<td><strong>Nonvolatility</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Why Flash Memory?

• Diversified Application Domains
  – Portable Storage Devices
  – Consumer Electronics
  – Industrial Applications
  – Critical System Components
Flash-based SSD Characteristics

- Random read is the same as sequential.
- Read and write by the unit of pages
- Does not allow overwrite. Need erase before writes. Erase is performed in blocks
- Typical block size is 128 K and page size 2K
- Write is slower than read. Erase is a very slow operation
- Read takes 25 microseconds, write takes 200 microseconds, and erase takes 1500 microseconds
- Limited number of writes per cell. 100 K for SLC and 10K for MLC.
- Flash Translation Layer (FTL) sits in between file system and SSD. FTL provides remapping and wear-leveling

Figure Source: "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage", Hyojun Kim and Seongjun Ahn, FAST 2008
High-Level View of Flash Memory Design
Summary of SSD Research Results

• Robust and Reliable Design of SSDs
• Integrating SSDs into Storage Hierarchy
• New FTL Design: A Convertible FTL Design
• Efficient Wear-Leveling Algorithm
• Optimal/Efficient Read/Write Caching
• Hot and Cold Data Classification
• Bloom Filter Design and Key-Value Store Based on Flash Memory
• Using Sampling Technique for Meta-Data Management in FTL
NV RAM Impacts

- Incorporate NV RAM into current systems as the main memory or part of the main memory
- Design or improve mechanisms to fully explore 1
Motivation

- Traditional HDDs possess their merits as they can provide large capacity with low cost.

- Areal density limit $\rightarrow$ Shingled Magnetic Recording

Figure Source: http://www.popularmechanics.com/technology/engineering/news/434255
Shingled Magnetic Recording

- Squeeze more tracks in the same area
  - Higher Capacity

- Write head is wider than track width
  - Support Sequential write
  - Random write/update interference

Possible Methods

• “In-place Update”: many small bands
  – Protect previously-written data by Read-Modify
  – Behaves similar to regular disks
• “Out-of-place Update”: few large band
  – Maintain data in circular log structure
    • **Data Addition** to head pointer
    • **Data removal** from tail pointer
  – LBA-to-PBA mapping is not fixed
  – Transfer random writes into sequential write
  – Compromise sequential read performance
Indirect Addressing & Defragmentation
(Garbage Collection)

• Similar to Solid State Drives (SSDs)
  – Keep track of valid data location
  – Out-dated data is invalidated
  – Invalid data must be cleaned when necessary
  – Valid data might be migrated during defragmentation (Garbage collection)

• Different from SSDs
  – In SWD, defragmentation requires no explicit action for invalid data erase.
New Memory/Storage Hierarchy

PCM as main memory provides:
1) High capacity
2) standby power

PCM as secondary storage provides:
1) Low access latency
New Memory and Storage Hierarchies

- Data Storage
- Data Migration
- Multi-Level Caching
- Data Prefetching
- Tiered Storage
Storage Layer Management and Caching

Read Queues (RT)

Read Queues (Prefetch)

Write Queues (Offloading)

When/Where/how much

Big Memory with PCM

SATA Disks

SSD

Cloud Storage
Cloud Storage and Big Data

• OpenStack
• VM vs. Container
• Durability, Reliability and Availability
• Private vs. Public Cloud
Project: Storage Systems Prototype with I/O Hints

File System
- QoS-aware IO calls
- QoS to hints

SCSI Device Driver
- I/O Requests
- SCSI Hints

Generic Block Layer
- Persistent Data Structures
- Data Blocks Classifier
- Cache Buffer
- Logical Volume

Device Mapper
- DM Table
- Hints Mapping Table
- Linear devices

HDD
SSD
Cloud
Thin Client

Building Hints Mapping Table
Hints generation

I/O Requests to File System

Cloud Objects Prefetch

Logical Volume vol1
Parallel File Systems and IO Workload Characterization
Why Is This Important?

- **Workload Characterization**
  - Key to *performance analysis* of storage subsystems.
  - Key to the *implementation of simulators*, as captured/synthesized workloads are key inputs.

- **Key Issues**
  - Lack of *widely available tool sets* to capture file system level workloads for parallel file systems
  - Lack of *methods to characterize* parallel workloads (for parallel file systems)
  - Lack of *methods to synthesize* workloads accurately at all levels (Block, File, etc)
  - Understanding of how existing workloads *scale* in the exascale regime is lacking
Goals and Objectives

• A detailed understanding and survey of existing methods in file system tracing, trace replaying, visualization, synthetic workload generators at the file system input levels, and existing mathematical models

• Tools, techniques and methods to analyze parallel file system input traces (require to know more about OS, meta-data server, and applications)

• Models to characterize the above workloads traces (Using statistical and analytical methods)

• Synthetic workload generation at the parallel file system input level – which will be used as inputs to the simulator.

• Understanding of the interactions of workloads at the file system level and making the file system aware of the workloads
Storage system performance cannot be determined by the system alone.

- \( P = f(S, W) \)

Possible Workload Space

Real Workload Space

• Improving system for all possible workload space is difficult.

• If we know the real workload space we can improve performance more efficiently.
Framework of I/O Workload Characterization

Original trace

Workload characterization

Comparison 2

Workload Parameters

Comparison 1

Parameter adjustment

Adjusted Parameters

Workload generation

Synthetic trace

Replayed trace

Replay on same/different storage system

Replay by workload replayer

Changes to applications and/or system (either host or storage)

Arrival pattern, File/Data access pattern in the form of parameters

Comparison 3

Action

Output
Tiered Storage Research

File Users

I/O Storage Subsystem

Units of RAID5 Sets

Active-Tier

Passive-Tier

Tape/Library

Tier 1

Tier 2

Tier 3

q

n-q

A2P–Migrate

P2T–Migrate

Reads
Data Migration, Duplication, and Deduplication

- Tiered Storage Management
- When a file is accessed, we may want to move related data level up to a faster storage provisioning potential near future access requests
- Duplication level optimal for a long-term storage
- Dedup algorithm and how to preserve it long-term (need to make sure we know how to get the data back)
- How to find the right balance between duplication and dedup? How do we validate that data is stored the we think it is?
- Imperfect dedup may be what we are looking for. However, what do we do if we want to have different levels of backup for different data.
Power Management and Adaptation
Power Breakdown

• Data Center Power Consumption
  • 50% HVAC
  • 20-35% Servers
  • 10-25% Storage
  • 5% Networking

• Different Types of data centers
  • Compute Centric (Ex: HPC)
    • 35% Servers, 10% Storage, 5% Networking
  • Data Centric (Ex: Enterprise)
    • 20% Servers, 25% Storage, 5% Networking
  • Average Case
    • 25% Servers, 20% Storage
IT Equipment Efficiency
50% power wasted!

<table>
<thead>
<tr>
<th>Component</th>
<th>Total</th>
<th>Used</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>80</td>
<td>60</td>
<td>Operating at 100% utilization</td>
</tr>
<tr>
<td>Fans</td>
<td>50</td>
<td>25</td>
<td>Temp. directed fan at 100% util</td>
</tr>
<tr>
<td>Memory (32 GB)</td>
<td>88</td>
<td>24</td>
<td>2GB DIMMS, 4W idle, 19W active</td>
</tr>
<tr>
<td>Hard drives</td>
<td>40</td>
<td>10</td>
<td>6 SATA drives, 25% busy</td>
</tr>
<tr>
<td>I/O adapters</td>
<td>20</td>
<td>4</td>
<td>25% disk, 15% network</td>
</tr>
<tr>
<td>Motherboard</td>
<td>22</td>
<td>12</td>
<td>N/S bridges &amp; devices, VR’s, …</td>
</tr>
<tr>
<td><strong>Total DC power</strong></td>
<td><strong>300</strong></td>
<td><strong>135</strong></td>
<td></td>
</tr>
<tr>
<td>Power supply loss</td>
<td>50</td>
<td>7</td>
<td>14% ➔ 5% loss of AC input pwr</td>
</tr>
<tr>
<td><strong>AC input power</strong></td>
<td><strong>350</strong></td>
<td><strong>142</strong></td>
<td>&gt; 50% of power is wasted</td>
</tr>
</tbody>
</table>
High Temperature Operation

- Chiller-less data centers
  - Less energy/materials, but space inefficient

- High temperature operation of comm./computing equipment
  - Smaller $T_{\text{outlet}} - T_{\text{inlet}}$
  - Deal with occasionally hitting temp. limits.

Need smarter thermal adaptability
IT systems fed by Renewable Energy

• Limit or eliminate energy draw from grid
  – Less infrastructure & losses, but variable supply
  – Need to consider impact on both computing & communications

• Similar issues with unreliable grid supply

Need better power adaptability
Applications of Data Deduplication and Cloud Storage
Overview of Chunking Algorithms

- Fixed-sized Chunking
- Content-Defined Chunking

Moving forward

byte stream

Window

\( FP(W) \mod (\text{Divisor}) = r? \)

True

set chunkpoint

False

Move fwd
Data Structures Associated with Chunking Deduplication (stop)

After chunking:

- c1
- c2
- c1
- c3

Index table:

<table>
<thead>
<tr>
<th>ID</th>
<th>loc(chunk)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1</td>
<td>loc(c1)</td>
</tr>
<tr>
<td>ID2</td>
<td>loc(c2)</td>
</tr>
<tr>
<td>ID3</td>
<td>loc(c3)</td>
</tr>
</tbody>
</table>

Chunk list:

- ID1
- ID2
- ID1
- ID3

De-duplicated chunks (stored in chunk store): c1, c2, c3
Thank You!
Questions?