SNIA Education

File Systems: What they are and why you care

Paul Massiglia, Symantec
SW Worth, Microsoft

SNIA Legal Notice

Abstract

Title: File systems: whatthey are and why you care

Abstract: File systems—the software that transforms bits stored on disks into business objects—are so much a part of the IT landscape that we hardly even notice them. We fire up the computer, and immediately start working with "folders" and "files," without ever reflecting on the complex software that makes it so easy and natural to store and process data electronically. But new requirements and new file system capabilities are making it necessary for decision makers to understand the "gets" and "don't gets" of different types of file systems. This tutorial looks inside the file system to describe how it does its magic, explores new and emerging file system capabilities, and addresses the question, "As an IT decision maker, why do I care what kind of file systems my department is using.

Learning objectives:

- Understand at an overview level the role file systems play in information technology.
- Appreciate the applications for advanced and emerging features offered by some file systems.
- Become equipped to make decisions about file system technology and usage in the IT environment.

- **Audience**: IT managers and decision makers

Agenda

- What’s a file system
- Why might I want one rather than another ?
- Things to watch for

Education SNIA

We hope you have checked out SNIA tutorial

The File System Evolution

A good technical foundation for this session

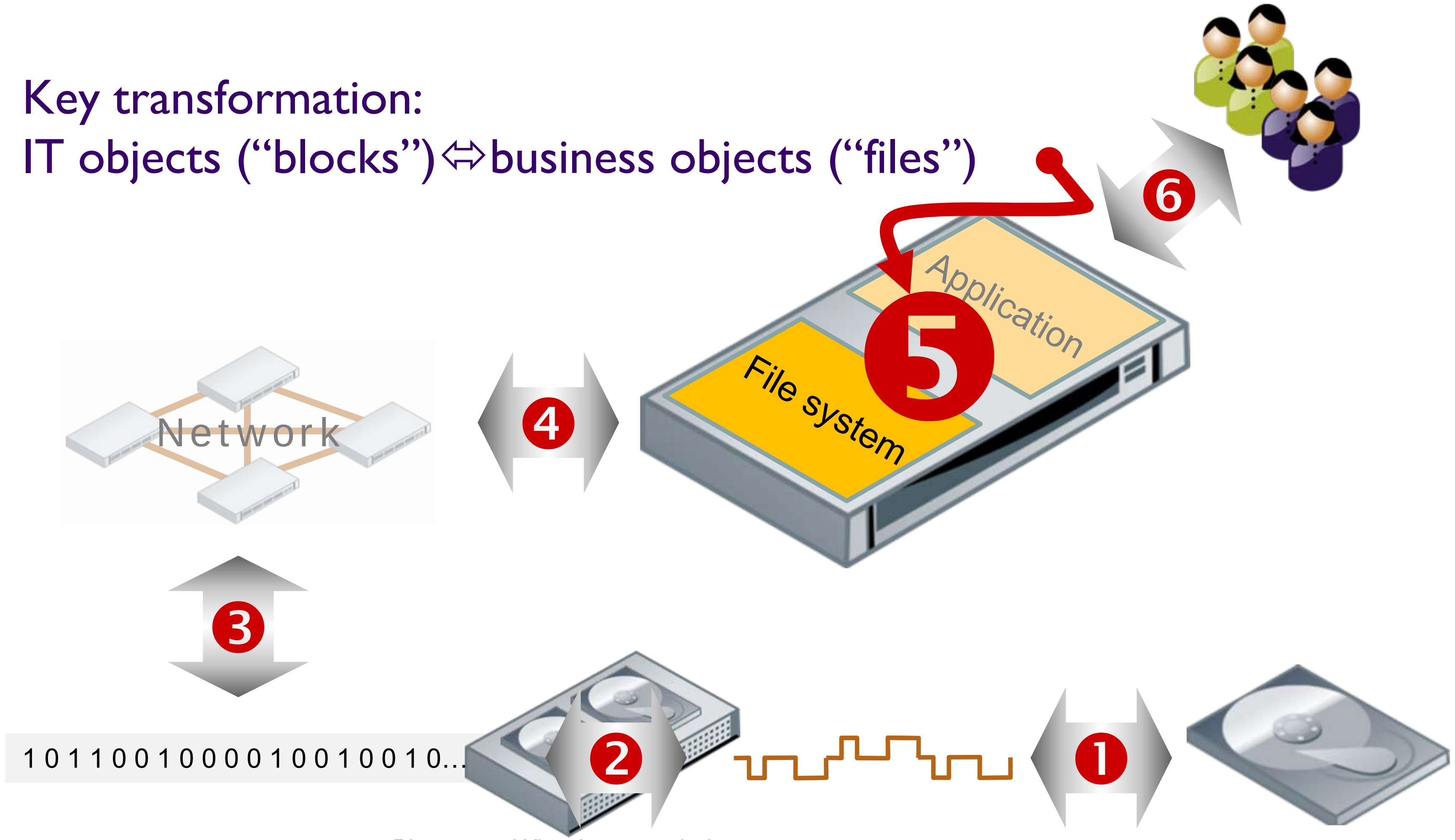
We bet you've already heard

- "Online data (particularly 'unstructured' data) is exploding"
- "Business processes have to be integrated with each other"
- "Data center complexity is increasing"
- "Managing IT costs too much"
- "There's no time for backup"

File systems have a powerful effect on all of these

The long and winding data path

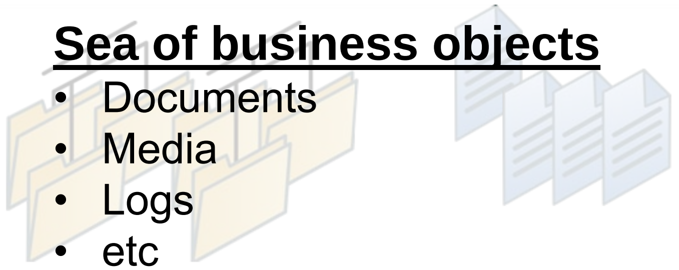
- Every step is a transformation to a different "abstraction"
- Key transformation:
  IT objects ("blocks")⇔business objects ("files")



What file systems do

Education SNIA

- Two things:



Sea of business objects

- Documents
- Media
- Logs
- etc

Name space management

Can I name a file **`/sales/2010/March`** ?

Who is allowed to read **`/hr/employees/performance`** ?

Storage management

Where is the data for **`/sales/2010/March`** stored ?

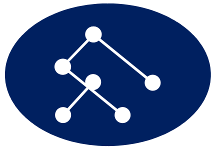
Where is there some free space for a new file ?

1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...
1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...	1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...
0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...
1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	1 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0...	0 0 0 0 0 free space 0 0 0 0 0 0 0 0 ...

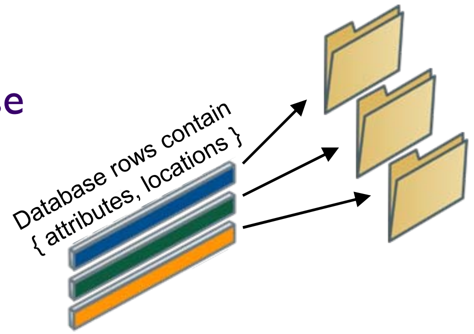
Ocean of blocks

What's in a name (space) ?

- Conventional name space
  - Syntax
  - Implied tree structure ("directories" or "folders")
  - Inheritance
  - Properties (ownership, etc.)



- Emerging
  - "RESTful" file access via the Internet "cloud"
    - Example: file name = URL (`http://mydomain.com/mydirectory/myfile.dat/read`)
  - "Flatten" the name space
    - Example: file management by relational database



Storage management
What's the big deal ?

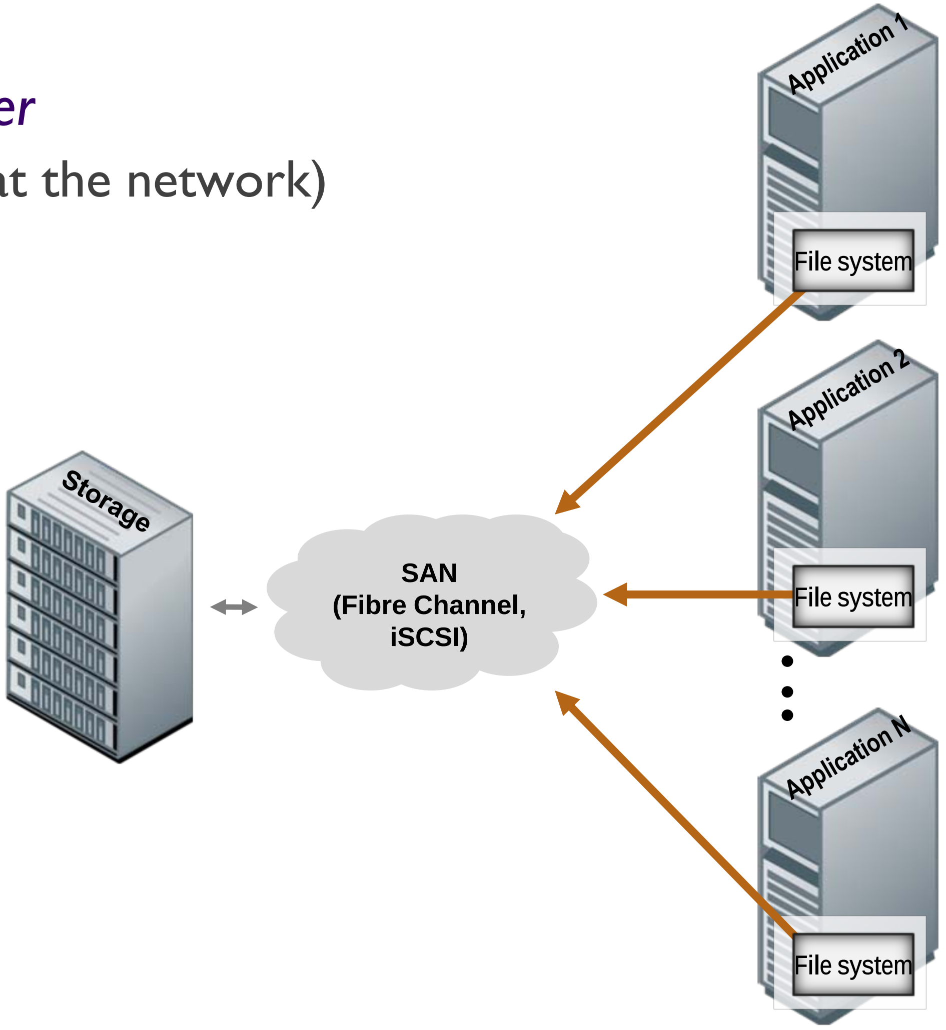
- A 1-terabyte disk contains ~2,000,000,000 blocks
- A 1-petabyte storage system contains ~2 trillion blocks
  - Spread across a thousand disks
- Now, don't lose track of any of them, even if,…
  - A disk can't read the data that's on it
  - A disk fails entirely
  - A server fails while it's writing data to a disk
  - Power fails while 10 servers are writing data to 50 disks
  - etc.
- …and deliver acceptable performance while you're doing it

Where file systems live

- "SAN"
  File system in the application server
  - High performance (measured at the network)
  - Excellent scaling

But…

- Every application has its own file system
  - Each one has to be managed
  - Any data sharing has to be coordinated
  - These are either
    - "Exercises for the user"
    - Automated by "cluster file systems"

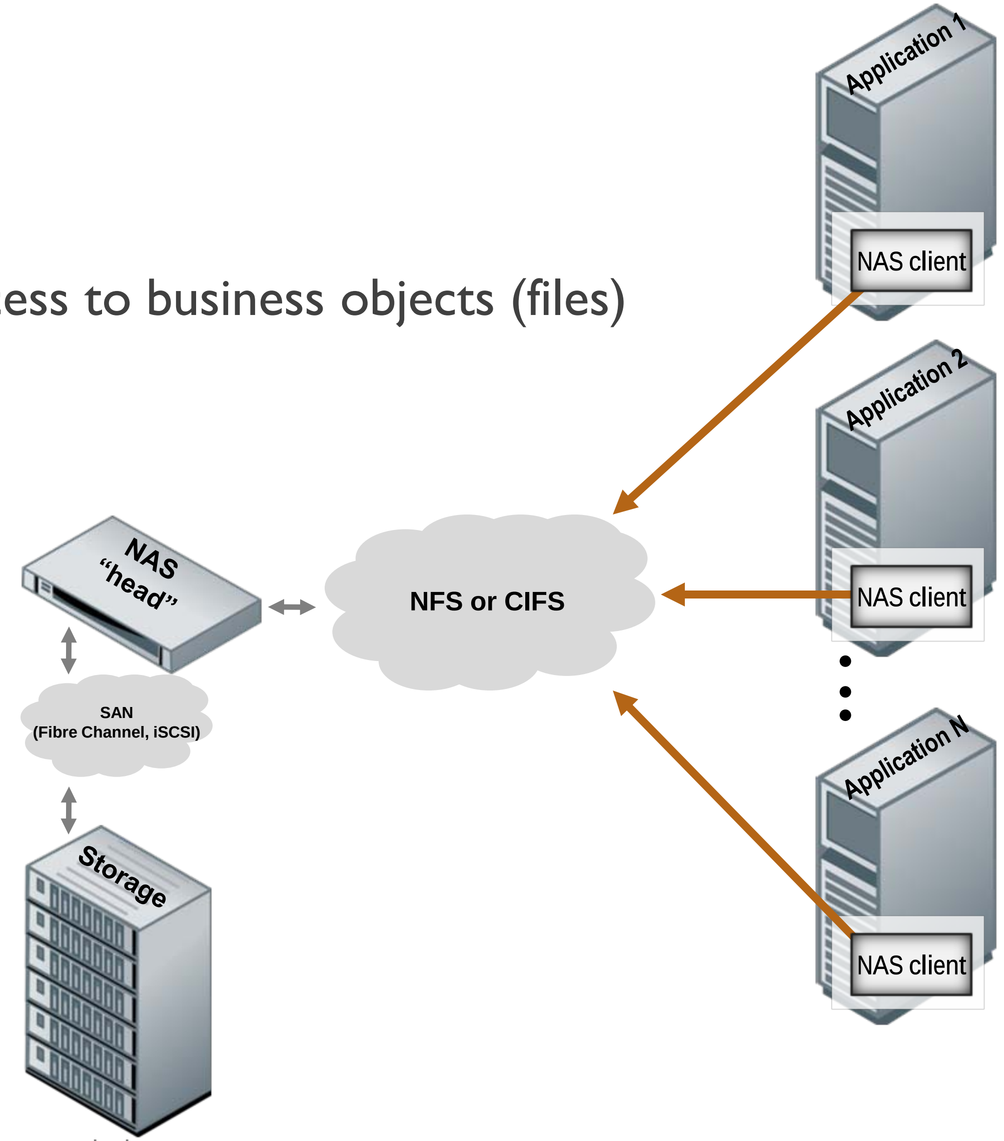


Where file systems live

- "NAS"
  File system in the NAS box
  - Easy to administer
  - Consistent, coordinated access to business objects (files)

But…

- Bottleneck-rich
  - Higher-latency protocols
  - Long I/O paths
  - The "NAS head"
- Conventional wisdom
  - SAN for transactions
  - NAS for "content"



OK,
I'm impressed
with what file systems do.

Now, why do I care ?

I have technicians who worry about this for me

Why do I care ?

- Well, for the most part, you don't

Why do I care ?

- Well, for the most part, you don't

Unless…

- You have to decide between "free" and expensive
- Your data "absolutely, positively has to be available"
- You can't predict how your data storage needs are going to grow
- You have petabytes to store and manage
- You can't afford a lot of technical talent
- You have to decide whether to spend a lot on "solid-state storage"
- You need to store data "on the Internet"

These are all interrelated

“Free” vs. expensive

- Operating systems include file systems

But…

- Software vendors charge handsomely for add-on file systems
- Should I pay, or is “free” good enough ?

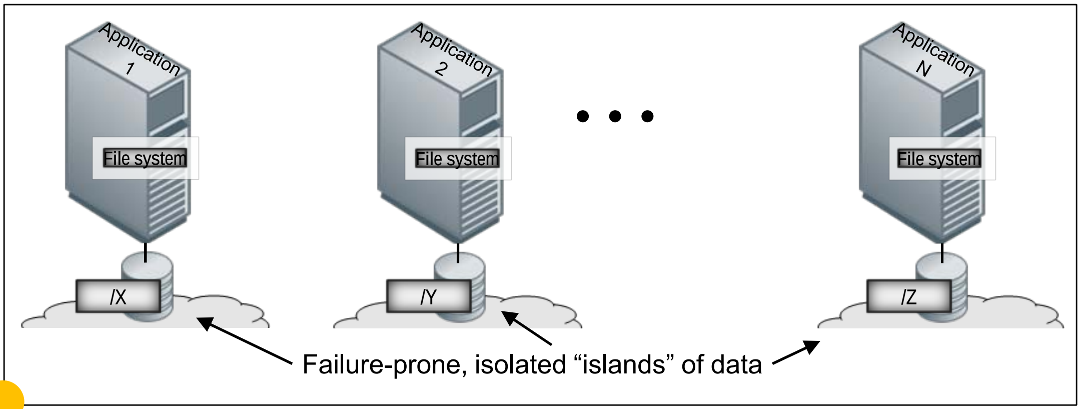
Usually, more $$$ = more features

Feature	When it's important
Clustering and replication	My business can't afford outages if • Something breaks • A disaster happens
Snapshots and clones	• Protection from malice and error • Testing & analysis
File tiering	• I have a lot of "idle" data that I want to store cheaply • I also have a little active data

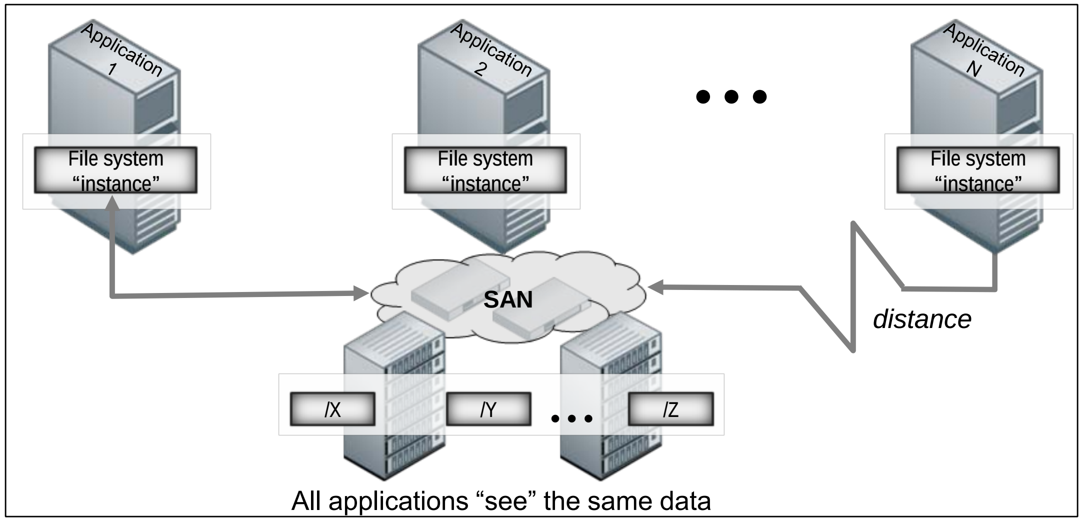
Let's look at each of these

Why clustering ?

From:

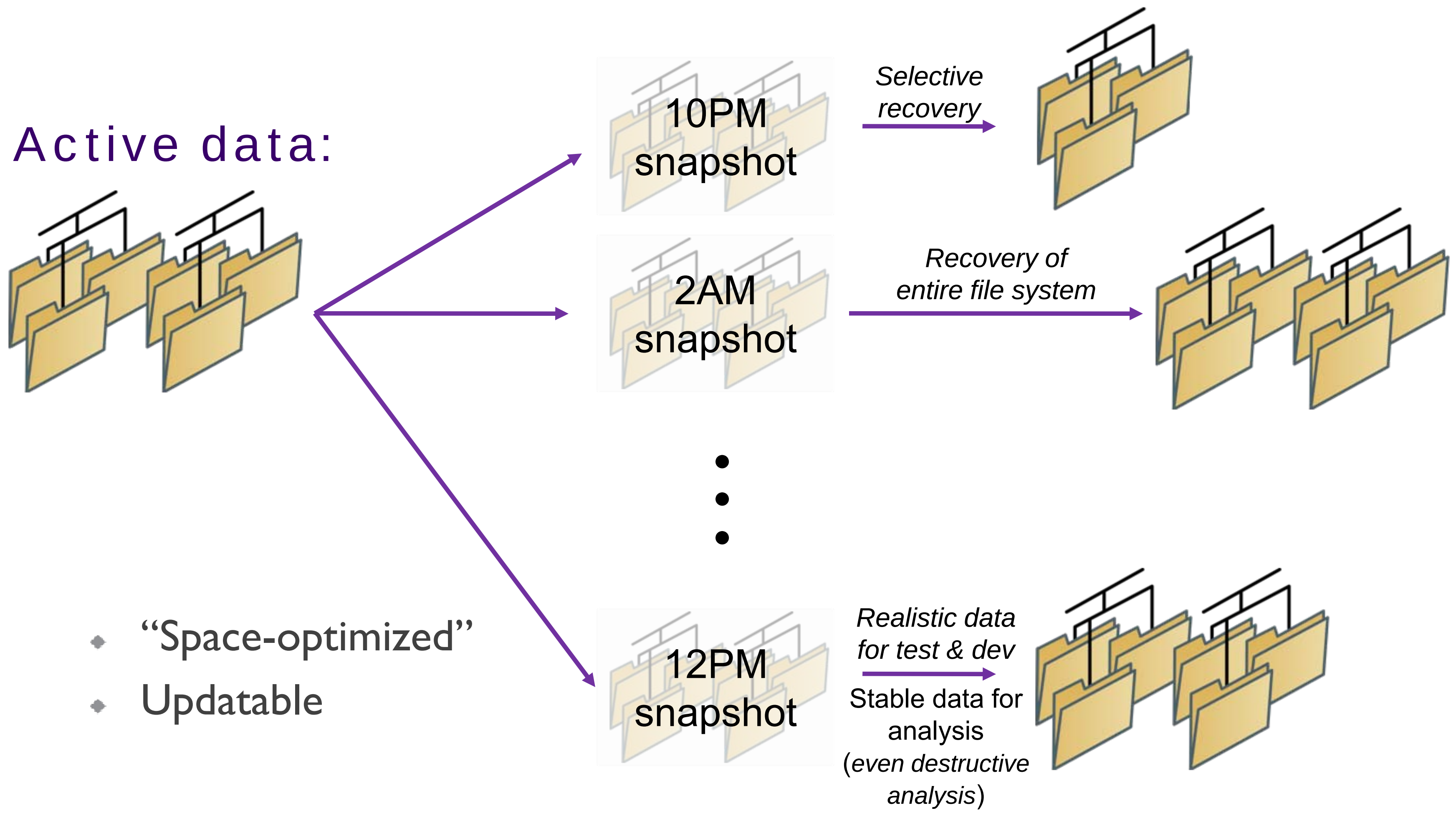


To:



- Enables
  - Fast recovery—even from disasters
  - Business process integration
  - "Scaling"

Why snapshots and clones ?

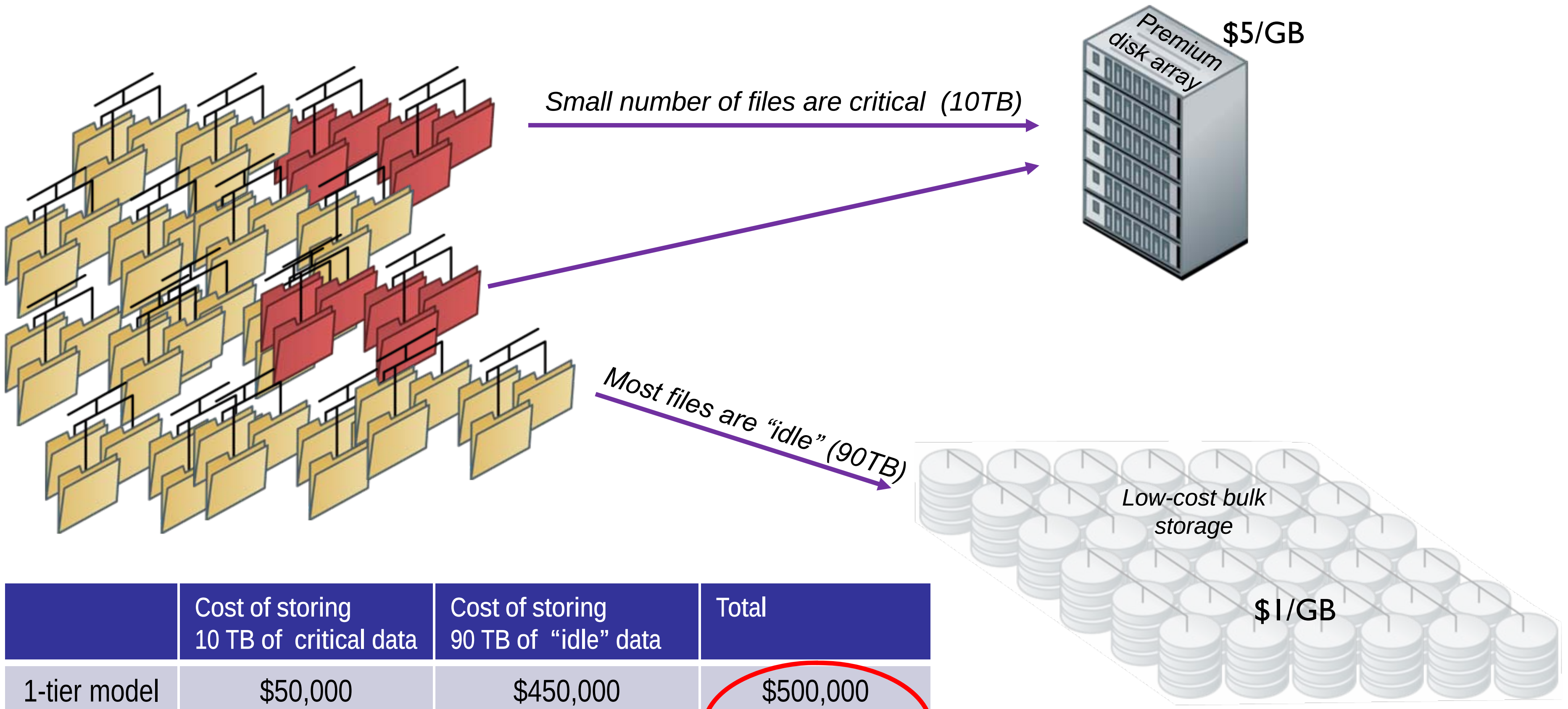


Active data:
10PM snapshot
2AM snapshot
12PM snapshot
Selective recovery
Recovery of entire file system
Realistic data for test & dev
Stable data for analysis (*even destructive analysis*)

- "Space-optimized"
- Updatable

Why file tiering ?

100TB of data



Small number of files are critical (10TB)
Premium disk array
$5/GB
Most files are "idle" (90TB)
Low-cost bulk storage
$1/GB

	Cost of storing 10 TB of critical data	Cost of storing 90 TB of "idle" data	Total
1-tier model	$50,000	$450,000	$500,000
2-tier model	$50,000	$90,000	$140,000

Other reasons you might want to pay for a file system

- Certain features remain somewhat unique
  - Automatic file tiering
  - Snapshots and clones
  - Solid-state storage awareness
  - Remote file replication
- Integration with the larger IT operation
  - Backup
  - Database management systems
  - etc…
- Vendor support
  - Expertise
  - Policies and attitude

Coping with unpredictable growth

Either…

- *"I have no idea how much data I'm going to have a year from now"*

Or…

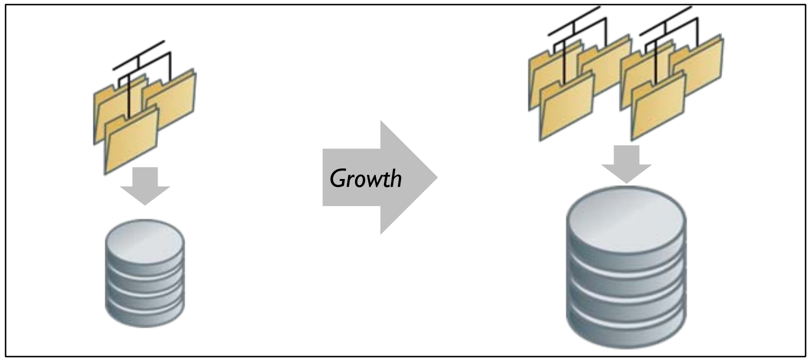
- *"Occasionally, I need an extra 100TB or so"*

- Two issues
  - Configuration: adding the storage to my data center
  - Provisioning: incorporating the capacity into my file systems

Requirement: file systems that can "grow" (take on more disk storage space)

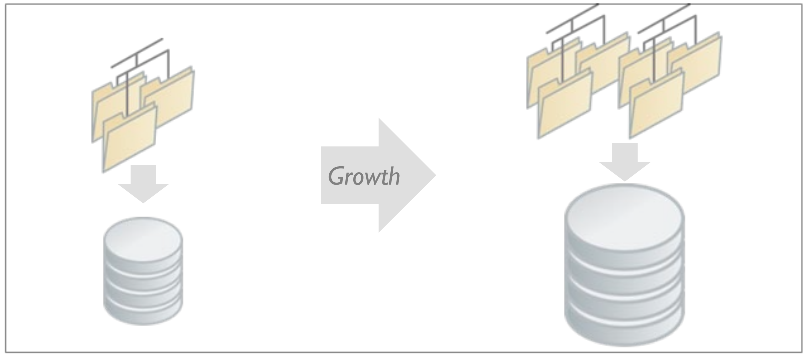
Coping with unpredictable growth

- The old way…
  - "Disks" that get bigger

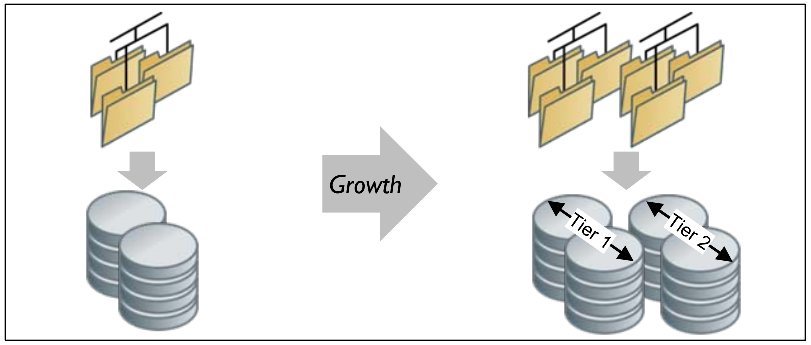


Coping with unpredictable growth

- The old way…
  - "Disks" that get bigger

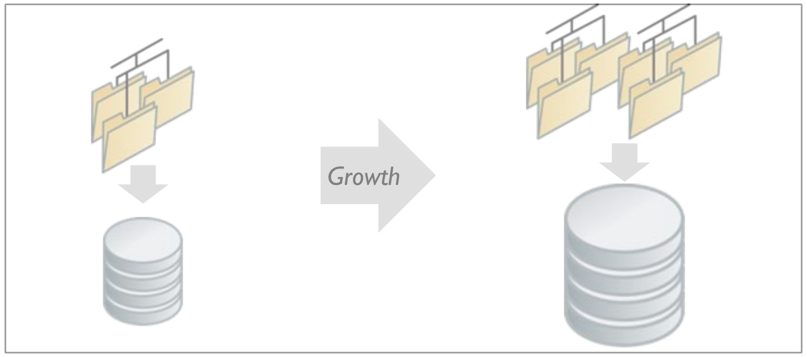


- The current way…
  - Storage pool-aware file systems

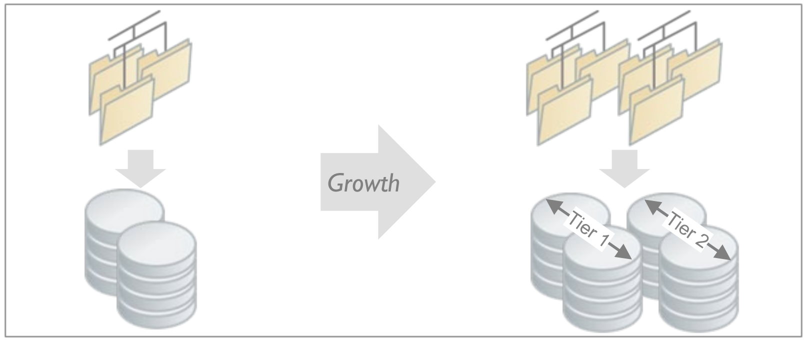


Coping with unpredictable growth

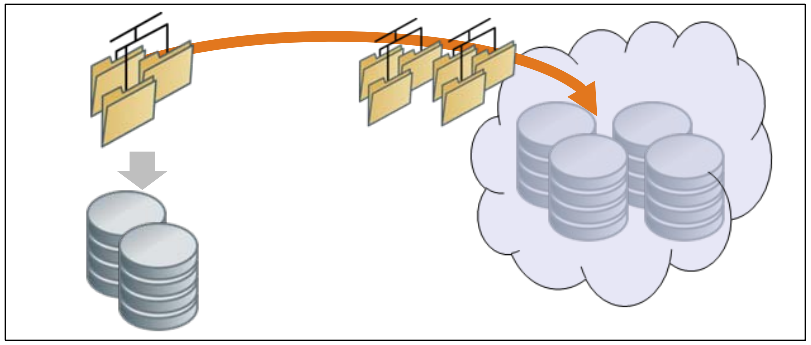
- The old way…
  - "Disks" that get bigger



- The current way…
  - Storage pool-aware file systems



- The emerging way:
  - Temporary overflow to "the cloud"



"I know how much data I have to store and it's *a lot*"

- Storing 100 million files is complicated
  - Searching, inserting, allocating space,…
  - …and the big one… "fscking" (pronounced 'fisking')

"I know how much data I have to store and it's *a lot*"

- Storing 100 million files is complicated
  - Searching, inserting, allocating space,…
  - …and the big hitter… "fscking" (pronounced 'fisking')

- Storing a petabyte of data is also complicated
  - At (an optimistic) 100 MB/s, it would take 115 days to copy (*backup as we know it is not an option*)

  - A disk will fail roughly every 20 days
    (*systems have to be designed to work with broken components*)

"I know how much data I have to store and it's *a lot*"

- File system solutions
  - Current: divide and conquer: organize files in multiple file systems (*by business application, file type, etc.…*)
  - Emerging: super-scale file systems (*there actually are file systems designed for petabytes*)
- Storage "farm" solutions
  - Current: enterprise-class disk arrays
  - Emerging: reliable systems built from commodity components
  - Even more emerging: "the cloud" as a second tier

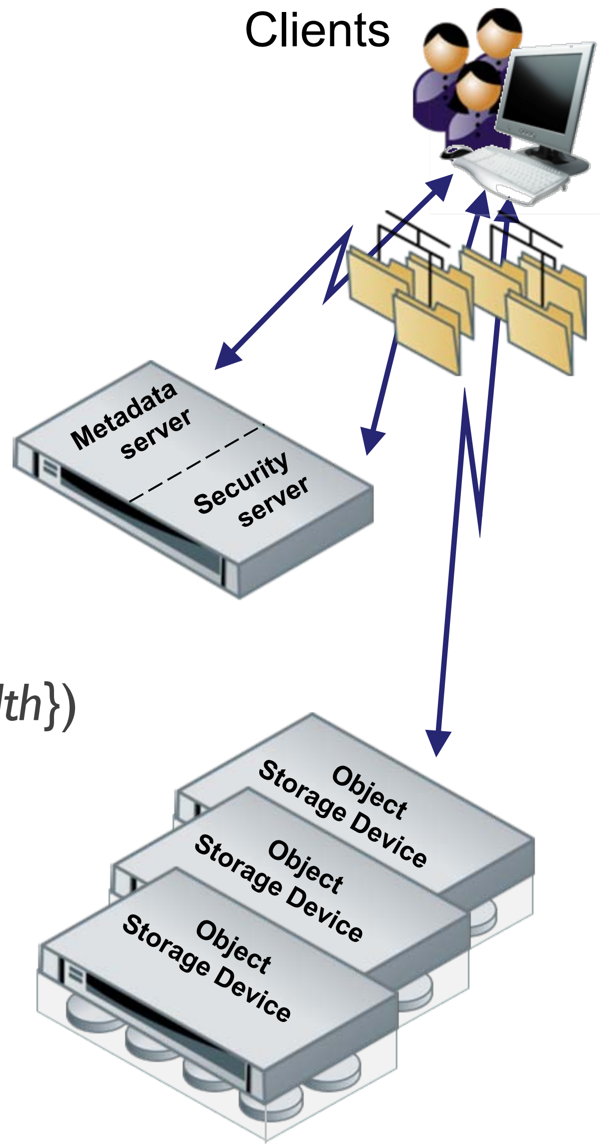
Check out SNIA Tutorials:

Object-Based File Systems: an overview

The entire cloud track

“I know how much data I have to store and it’s *a lot*”

- Object-based file systems
  - Based on object storage devices (OSDs)
  - File = one or more <u>*groups of objects*</u>
  - Metadata Servers manage objects
  - Clients communicate directly with OSDs
- Why they work:
  - Scaling:
    (*more capacity* = *more* {*processing, cache, bandwidth*})
  - Performance
    (*secure direct client⇔OSD data transfer*)
  - Robustness
  - (*layers of fault protection*)



"I know how much data I have to store and it's *a lot*"

- What's the downside ?
  - There has to be a downside
- There is:
  - Maturity of the technology
  - Fewer suppliers
  - Some products require custom integration
  - Limited integration with data center processes (backup, database,…)
- But…
  - The pressure to keep more data online is building
  - Standardization is proceeding

So…watch this space

Who can afford a roomful of IT wizards ?

- It's not as simple as just managing file systems
  - Storage provisioning
    - How many of what kind of "disks" do I need ?
    - When I need more (or less) storage, how disruptive is that ?
  - Application integration
    - Can data be backed up in a reasonable time ?
    - Do the right applications (and only they) have access to data ?
    - Do my structured databases get the service levels they need ?
  - End-to-end tuning
    - How do I find the root causes of application response time problems ?
    - Is hardware (particularly expensive hardware) being used effectively ?

Who can afford a roomful of IT wizards ?

Current techniques

- Storage provisioning
  - "Volume" managers
  - Disk storage virtualization products
  - Flawed premise: storage provisioning is a separate problem from file management
    (*NAS improves the situation somewhat*)
- Application integration
  - May be a reason to consider premium file systems
- End-to-end tuning
  - Operating system tools + human intuition

Who can afford a roomful of IT wizards ?

Education SNIA

Emerging techniques

- Integration of the "stack"
  - Software file systems that manage their own storage
- Task elimination
  - File storage systems that automatically
    - Balance load across resources
    - "Self-heal" when something fails
    - Automate client-side tasks (e.g., mount management)

What's all this about "solid state storage"

- The price of the flash memory used in cell phones, digital cameras, MP3 players etc. has dropped to the point where it's realistic to consider using it as high-performance storage in the data center
- Vendors have (wisely) chosen a disk emulation model ("SSD")
  - Works with existing infrastructure and application software
- Pluses
  - Very high performance (especially random retrieval)
  - "No moving parts" reliability
- Minuses
  - Device "wearout"
  - Relative cost is still breathtaking

Check out SNIA Tutorials:

The entire SSD track

What's all this about "solid state storage"

- Good uses for SSDs
  - Retrieve-mostly "information banks"
    - e.g., stable records of recent transactions
  - High-value, small-bulk data
    - e.g., currency trading
  - Extremely latency-critical applications
    - e.g., currency trading
- Not-so-good uses for SSDs
  - Update-intensive applications
    - e.g., logs, current transactions,…
  - Large, sequentially-written files
    - A/V streams, genomic databases,…

SSDs and file systems

Check out SNIA Tutorial:

'Storage tiering and the impact of flash on file systems'

- New requirements
  - Support for "tiered" storage
    - Small, high-performance, expensive
    - Large, middling-performance, low-cost
  - Effective exploitation of tiered storage
    - Automated migration among tiers
    - SSDs as NVRAM
- File systems and SSDs aren't really integrated yet
  - Much complication could be eliminated
  - File systems need to accommodate SSD peculiarities
- Recommendations
  - Deploy for specific reasons
  - Look for file systems with some degree of "SSD awareness"
  - Have a plan to cope with device wearout

"I need to store files on the Internet"

- The most rapidly changing facet of enterprise data storage
- Value proposition
  - "Pay as you go" pricing
  - Fast reaction to overflow situations
  - Minimize real estate, hardware, power & cooling, administrative staff,…
- "Standards"
  - De facto: Amazon S3
  - Emerging: SNIA's Cloud Data Management Interface
- Elephants in the room
  - A few big players
  - A plethora of smaller players that want to be big
  - Application-specific storage services (e.g., email archiving, backup,…)

Check out SNIA Tutorial:

CDMI

"I need to store files on the Internet"

- Ask yourself what you need:
  - Storage as a service (general & application-specific)
  - Storage + computing as a service
  - Storage + application development environment as a service
  - Integration of on-premise storage and cloud service

- Ask candidate providers:
  - How do I access data stored in your cloud ?
  - What security, availability, and performance guarantees ?
    - In particular, where can my data be stored ?
  - What's your liability if you lose my data ?
  - If I don't like your service, how do I get my data back ?
  - How do I avoid needless charges ?

"The cloud" is developing rapidly in "learn as we go" mode

It will almost certainly change the nature of data centers significantly in the next half decade

Closing thoughts

- For 'run-of-the-mill' applications
  - Terabytes, not petabytes; megafiles, not gigafiles
  - Go with "free"—they're actually pretty good
- Extraordinary circumstances demand extraordinary capabilities
  - Scale (capacity, files, clients, performance)
  - Predictability
  - Requirements (application integration, clones, storage tiers, etc.)
- File systems
  - Are continually evolving and being enhanced
  - What you pay for today may be "free" tomorrow
  - Are increasingly becoming "appliance-like"
- You should really be looking at "the cloud"
  - If you reject it, reject it for good reason

Q & A
(time permitting)

SNIA Tutorials references

- Please check also the following tutorials

Check out SNIA Tutorial:

The File Systems Evolution

Check out SNIA Tutorial:

Object-Based File Systems: an overview

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA
  - trackfilemgmt@snia.org
    (File Systems & File Management)

Many thanks to the following individuals for their contributions to this tutorial.

- SNIA Education Committee

Philippe Nicolas	Kerstor
Christian Bandulet	Oracle
Tushar Tambay	Symantec
Jonathan Goldick	LSI