



Education

Tiered File System without Tiers

Nick Kirsch
Director of Product Management, Isilon Systems

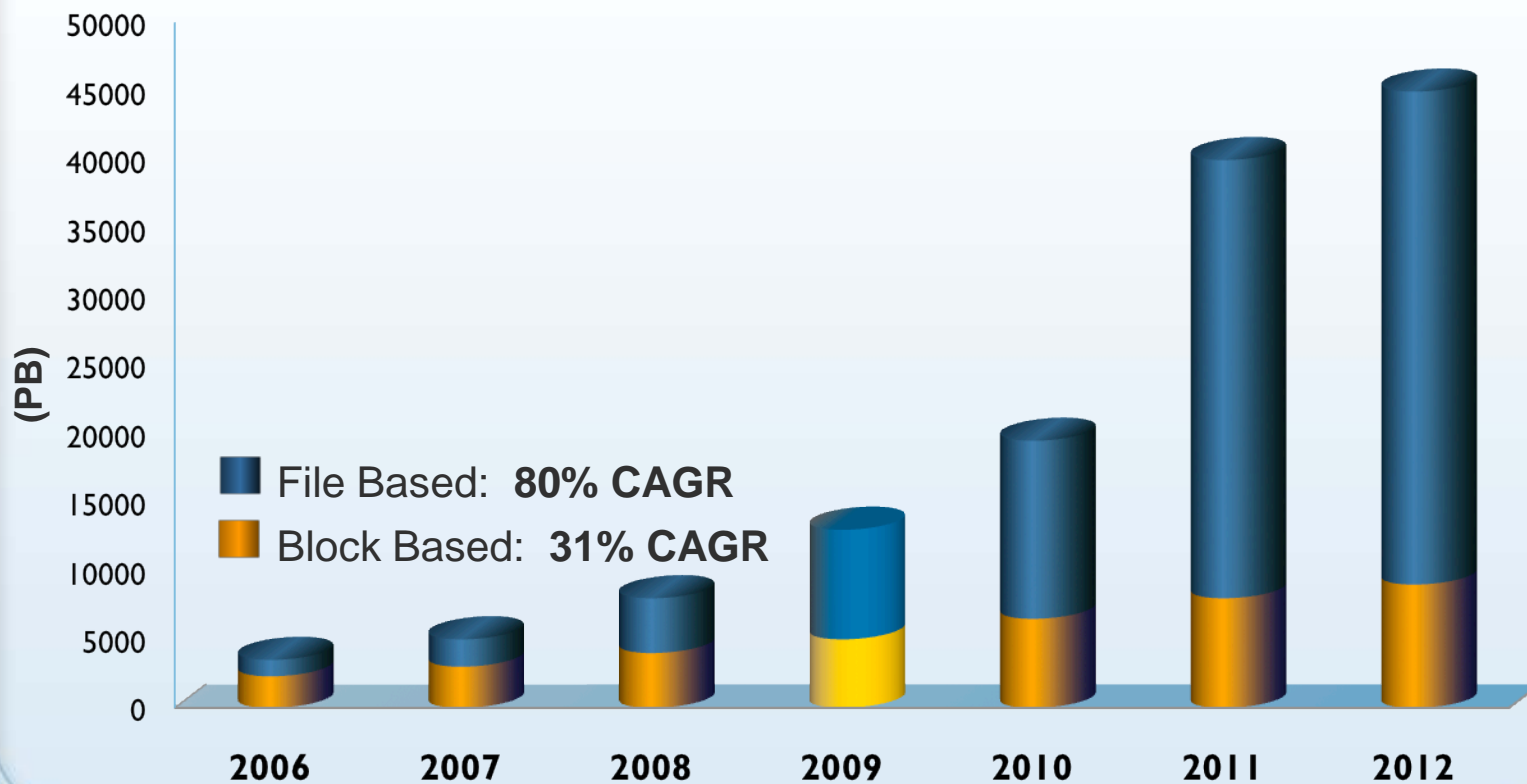
- The material contained in this tutorial is copyrighted by the SNIA.
 - Member companies and individual members may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced in their entirety without modification
 - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
 - This presentation is a project of the SNIA Education Committee.
 - Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
 - The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.
- NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.**

➤ Tiered File System - without Tiers

- ◆ File systems have evolved considerably, yet the vast majority of them are still a simple organizational layer on top of a block device. As the sheer amount of data scales in an organization, it becomes absolutely critical that file systems evolve as well. This presentation focuses on a modern file system which include native tiering capabilities, native per-file performance and protection capabilities, and advanced scalability.

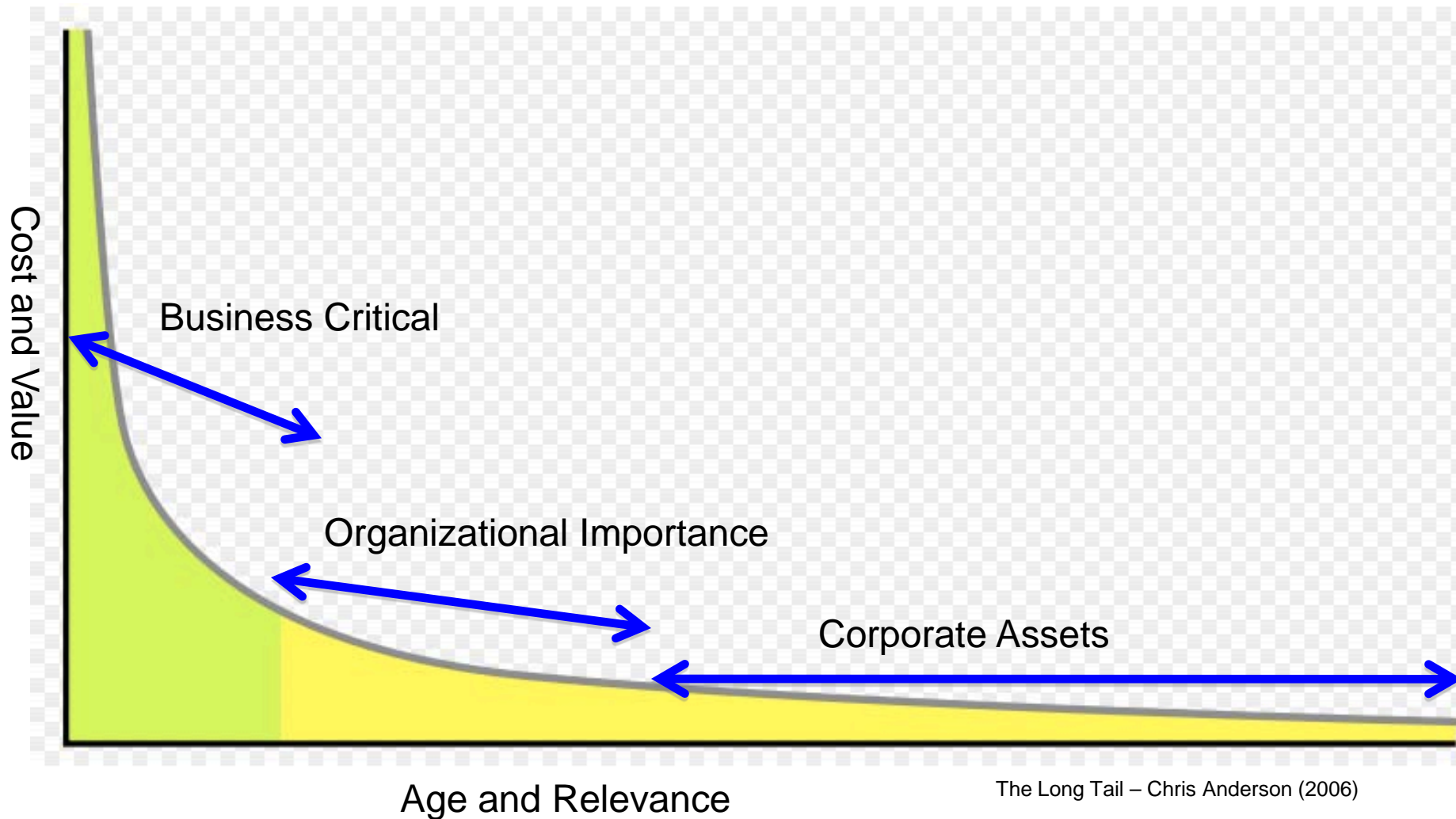
You Face a Deluge of Data Growth

Worldwide File And Block Disk Storage Systems, 2005-2012*



* Source: IDC

Is Most of That Data Valuable?



What are the challenges?

➤ CIO/Director of IT

- ◆ Budget – Storage Cost VS Value of Data
- ◆ Business Compliance and Security Requirements

➤ Storage Administrator

- ◆ Policies for Managing Data
- ◆ Policies for Managing User Interaction
- ◆ Storage System Management

➤ End-User

- ◆ Forced to Manage Data
- ◆ Forced to Understand Storage Landscape

- Scalability with Flat Operating Costs
- Average Cost Tracks Cost/Commodity Curve
- Automated and Granular Data Classification
- Automated and Granular Data Movement
- Transparency to Application and End-User
- Simple, Easy-To-Manage, Scales “Ininitely”

➤ Turtle Strategy – Slow Growth

- ◆ Data-growth is significantly below Moore's Law
- ◆ Provision Additional Storage When Necessary
- ◆ Migrate All Data to New Storage
- ◆ Little Impact on User Data Management



Storage Array



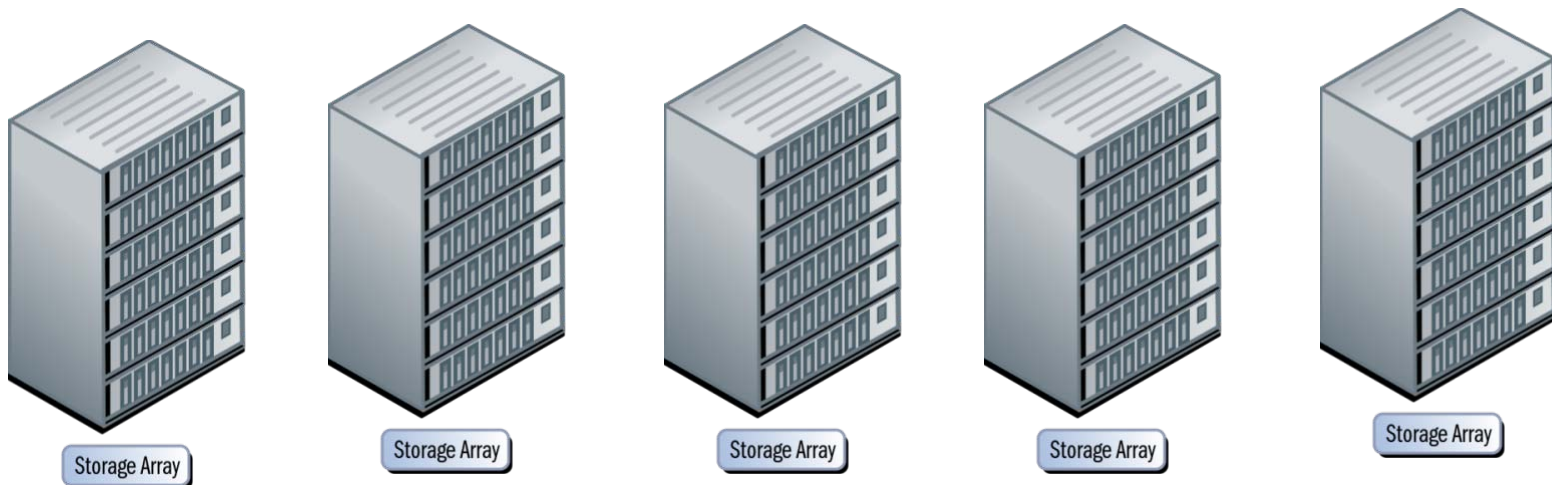
Storage Array



Storage Array

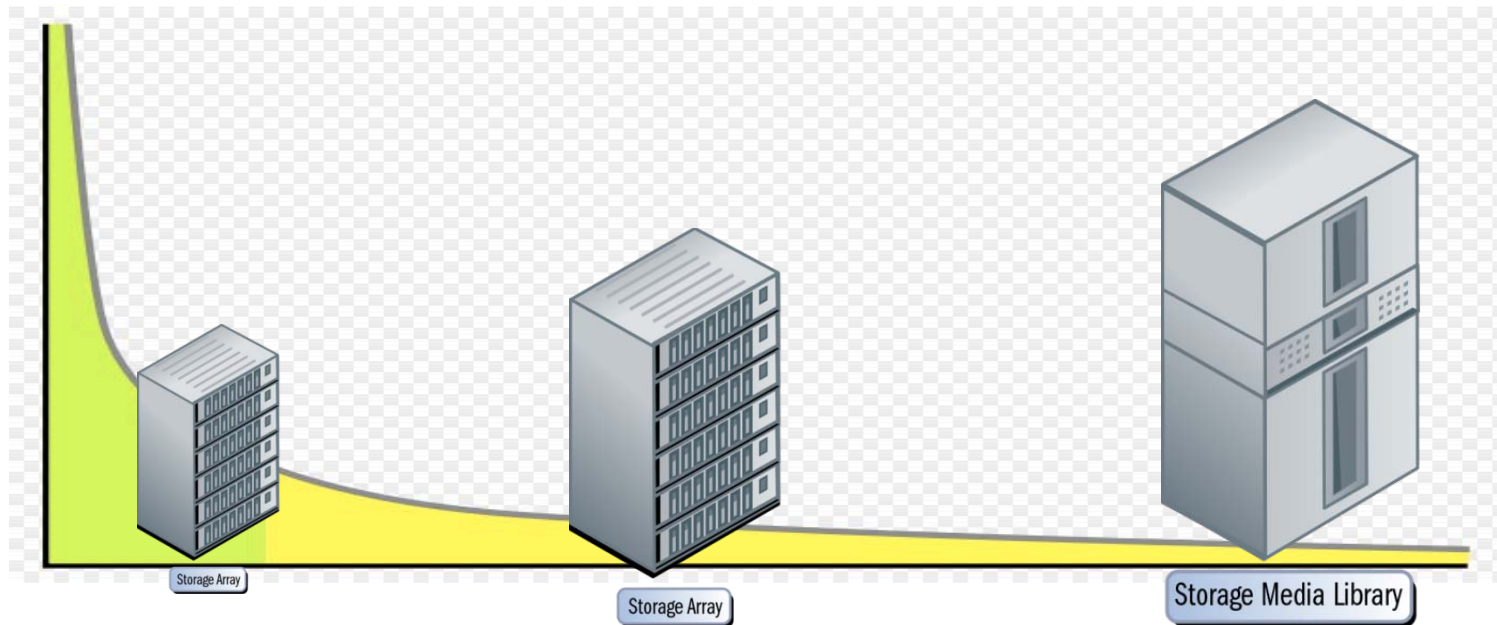
➤ Pod Strategy – Silos of Storage

- ◆ Data-growth matches Moore's Law
- ◆ Provision Additional Storage On Regular Cycle
- ◆ New Data to New Storage
- ◆ Users Manage Data Locations



► Tiering/Hierarchical Storage

- ◆ Tiering means establishing a hierarchy of storage systems based on service requirements such as performance, business continuity, security, protection, retention, compliance, and cost.



Tiered File System without Tiers

© 2010 Storage Networking Industry Association. All Rights Reserved.

Today's Tiering Implementations

➤ “Stubbing”

- ◆ Copy/Delete Scheme
- ◆ Leaves References

➤ Pros

- ◆ Helps End-User

➤ Cons

- ◆ Technical Challenges
- ◆ Increased Management
- ◆ Silos of Storage
- ◆ Affects Backup, DR, etc.

➤ Software-Layer

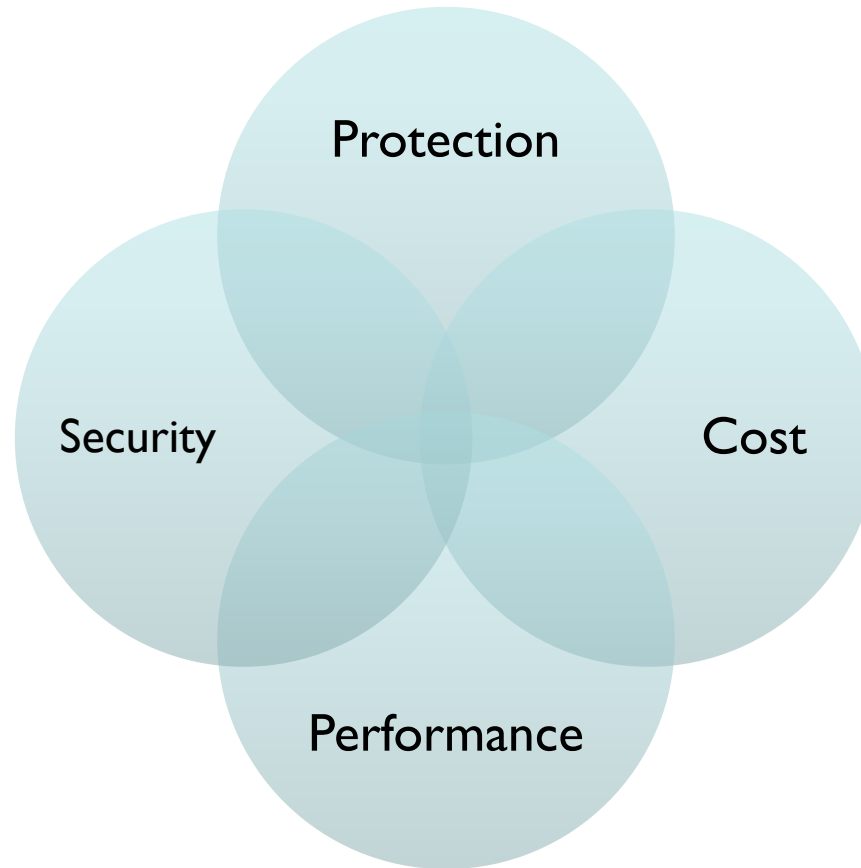
- ◆ Client Software
- ◆ Asset Management

➤ Pros

- ◆ Helps Administrator

➤ Cons

- ◆ Unfamiliar for Users
- ◆ 3rd Party Dependency
- ◆ Silos of Storage
- ◆ Affects Backup, DR, etc.



- An Abstraction on Top of Storage Media
- Organizes Data into Files, Databases, Indexes
- Network Attached System (NAS)
 - ◆ File System is Understood by the Storage System
 - ◆ Clients and Storage System Share a Higher-Level Protocol
- Storage Area Networking (SAN)
 - ◆ File System is Understood by the Client (s)
 - ◆ Clients and Storage System Share a Lower-Level Protocol

- Tiering Multiple File Systems Introduces Complexity
- Tiering Requires Different Storage Media
- Effective Scaling Requires Networking

- File Systems are Media and Network Agnostic
 - ◆ File Systems interact with Volumes and LUNs
 - ◆ File Systems are ignorant of the underlying media
 - ◆ File Systems are built for single systems

- Hardware Abstraction is No Longer Necessary
 - ◆ It complicates scaling, management, and increases costs

➤ What Can You Do with a Single File System?

- ◆ That spans multiple hardware tiers of storage ...
- ◆ That can scale seamlessly from GBs to EBs ...
- ◆ That allows old hardware to be removed ...
- ◆ That allows new hardware to be added ...
- ◆ That performs protection and security at a file-level ...
- ◆ That can optimize for specific hardware and media ...
- ◆ That can optimize for specific types of files over time ...

➤ You can implement Tiered Storage ...

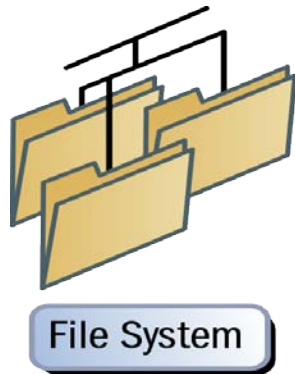
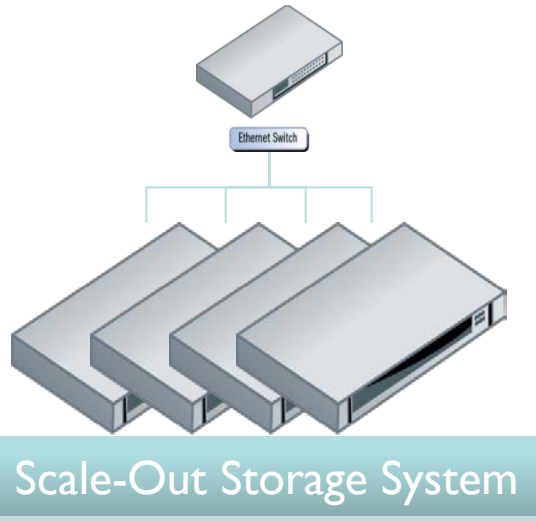
➤ Critical Design Principles

- ◆ The File System Must Be Aware of All Media
- ◆ The File System Must Be Scalable
- ◆ The File System Must Be Easy To Manage

➤ Consequences

- ◆ File System Provides Data Protection
- ◆ File System Addresses Different Storage Media
- ◆ File System Accesses Networked Components
- ◆ File System Allows for Easy Addition and Removal
- ◆ The File System Must Provide Granular Policies

Implement Tiering with Scale-Out



- Hardware Building Blocks – “Nodes”
- Single File System and Volume
- High Performance Network Interconnect
- Integrated Multiple Tiers of Storage
- Easy-to-Use, Cost-Effective
- SAN or NAS

➤ Different Hardware Nodes

- ◆ Globally distributed DRAM cache
- ◆ Blazing Fast Tier 0/1 – SSD or SAS nodes
- ◆ Mid-Range Tier 2 – performance SATA
- ◆ Cost-Optimized Tier 3 – dense SATA
- ◆ Performance Nodes – All CPU, DRAM, no disk
- ◆ Meta-Data Acceleration

➤ Per-File/LUN Granularity

➤ Single File System, Single Volume, Single Namespace

- Executive Staff Files (and Exchange Mail Store)
 - ◆ Optimize for Random Access and Meta-Data Lookups
 - ◆ Data Protection: +3 (survive three simultaneous failures)
 - ◆ Use Tier-I Storage Nodes (SSD or SAS)
 - ◆ Automatically Migrate to Tier-2 (SATA) after 3 days
 - ◆ Automatically Migrate to Tier-3 (dense SATA) after 7 days
 - ◆ Store Snapshots on Tier-3 (dense SATA) immediately
- Optimizes for **Immediate** Performance
- Minimizes High-End Performance Cost
- Continual Protection At All Times
- Cost Reduce “Old” Data Very Quickly

- VMware Images (both LUNs and files)
 - ◆ Optimize for Random Access
 - ◆ Default Protection: 2x (survive two simultaneous failures)
 - ◆ Use Tier-I Storage Nodes (SAS)
 - ◆ Store Snapshots on Tier-2 (mid-range SATA)

- Optimizes for **Application** Performance
- Different Protection Scheme for Performance
- Cost Reduce as Value Declines

- Archive Policy – For Everything Else
 - ◆ Default Protection: +2 (survive two simultaneous failures)
 - ◆ Use Tier-3 Storage Nodes (dense SATA)
 - ◆ Move Data to Tier-2 If Activity Increases

- Optimize Cost Across the File System
- No Manual Intervention
- No User Visibility

Scale-Out is Perfect for Tiering

- Scalability with Flat Operating Costs
 - ◆ Single File System, Single Point of Management
- Average Cost Tracks Cost/Commodity Curve
 - ◆ Automatically Drive Data to the Most Cost Effective Tier
 - ◆ Constantly Minimize the Use of the Performance Tier
- Automated/Granular Data Classification/Movement
 - ◆ Policy-Based, Per-File/LUN Classification
 - ◆ Protection, Performance, Security, and Cost
- Transparency to Application and End-User
 - ◆ Single Name Space, No Name Space Tricks, No Stubs
- Simple, Easy-To-Manage, Scales “Infinately”

- Please send any questions or comments on this presentation to SNIA: trackfilemgmt@snia.org

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

- SNIA Education Committee

Nicholas Kirsch