Hierarchical File Systems are Dead

Margo Seltzer, Nicholas Murphy
May 18, 2009
HotOS 2009
A Moment of Silence

HERE LIE SEVERAL GENERATIONS OF FILE SYSTEMS 1965-2009
Audience Participation
Is this your grouping?
How about this?
Perhaps this one?
Why different answers?

• Each object had different attributes
  – Shape
  – Color
  – Size
• How you sort depends on how you want to use them?
  – I need things to represent buttons
  – I need something that will go with a Harvard shield
  – I need things that will fit in this picture
Outline

• A geek’s view of namespaces: hierarchies.
• Real people’s view of namespaces: search.
• What should we do about it?
Files are Objects

- So, where do you store these objects
  - Red/Triangle/Small
  - Square/Medium/Blue
  - Large/Circle/Yellow
- Kind of depends, doesn’t it?
Fundamental Problem

- Hierarchical namespaces:
  - make you designate a most important attribute
  - entangle storage and naming
  - make you know where something is to find it

- So why do we use them?
  - We need to organize the physical world.
  - We can’t juggle too many things at once.
  - They bring order to chaos.
How did we get here?

• The virtual world models the real world.
  – Filing cabinets
  – Dewey decimal system
  – *Can only arrange physical objects one way*

• Access to data performed by people.
  – People can only handle a limited number of things.
  – File systems were small
  – *Just wasn’t a big problem*
When it was a big problem

- Databases to the rescue!
- Highly structured data.
- Large volumes, accessed efficiently.
- But …
  - Provide controlled access to data
  - Limited interfaces
  - No “hands-on” manipulation
Poor man’s data management

- Lighter weight than a DBMS
- Easier to use than a raw device
- What is it?
Poor man’s data management

• Lighter weight than a DBMS
• Easier to use than a raw device
• What is it?
• The File System!
  – Unstructured
  – Directly accessible
• Served quite reliably for three decades …
Outline

- A geek’s view of namespaces: hierarchies.
- Real people’s view of namespaces: search.
- What should we do about it?
The Web

• Tried to impose a hierarchical namespace.
• When was the last time you typed a URL longer than: www.someplace.com?
• Why not?
The Web

- Tried to impose a hierarchical namespace.
- When was the last time you typed a URL longer than: www.someplace.com?
- Why not?
  - Auto-complete
  - Bookmarks
  - Search
Data Quantity + Web = Search

• The web is too big to organize.
• Most people’s computers are now too big to organize too:
  – MacOS Spotlight
  – Windows Desktop Search
  – Google Desktop.
• Normal users don’t know where things live.
• Most of us don’t know either.
The Brave New World

• What do we need from file systems:
  – Backward compatibility
  – Separate naming from access
  – Unstructured data access
  – Direct access to data
Outline

• A geek’s view of namespaces: hierarchies.
• Real people’s view of namespaces: search.
• What should we do about it?
A Modest Proposal

- Let’s rearchitect the world!
- Get rid of the hierarchy as a structuring mechanism.
- Build a system with search as its primary access method.
- Implement the hierarchy on top of that.
A Proposed Architecture

Stable Storage
A Proposed Architecture

Extent Allocator

Stable Storage
A Proposed Architecture

Object Index and Metadata

Extent Allocator

Stable Storage
A Proposed Architecture

Type-specific Indexes
Object Index and Metadata
Extent Allocator

Stable Storage
Invitation

• We’re trying to build one of these.
• We invite you to build one too.
• Why not just build indexes on top of POSIX?
  – Might be right -- it’s what we’re doing now.
  – But POSIX is limiting in some ways.
  – We think the world will be simpler and cleaner and
    will perhaps pave the way for more interesting
    things to come.
A Moment of Silence

HERE LIE SEVERAL GENERATIONS OF FILE SYSTEMS 1965-2009
Thank You!

Margo Seltzer, Nicholas Murphy
margo@eecs.harvard.edu
May 18, 2009
HotOS 2009
New APIs

• Naming
  – Fake POSIX
  – Tagged lookup:
    • FULLTEXT/HotOS + FULLTEXT/2009 + IMAGE/

• Access
  – Pretty traditional (read, write)
  – But how about:
    • Insert bytes into the middle of an object
    • Remove bytes from the middle of an object
    • Free space from the middle of a file (without moving things around).