pStore: a Semantic-Aware File Store

Christos Karamanolis
Magnus Karlsson
Chunqiang Tang*
Zhichen Xu

*U. of Rochester
Motivation

storage systems are extensions to human memory
   but there is a huge gap between the two
   – human brains are smart
   – the storage systems are dumb

this is because we associate “meanings” (semantics) to the things we remember, but the storage systems today don’t!
Motivation

when we search for something

– we may remember this something has certain property
– we may have abstract notion of the object (e.g. tiger has stripes)
– we may do so according to the relationships of this object to other objects, e.g., this paper was written by a student of …

– …

when we memorize things, we may

– remember only the differences of an object to another object
– recognize that many objects belong to the same “category”
– …
we discover meanings of things incrementally; we realize usage of things gradually
Today’s contents are full of semantics

example: a digital movie studio
  • hundreds of scenes
  • millions of data objects

a variety of semantic information
  – every version of an object is annotated with changes done
  – info. about versions and dependencies among files is important when, e.g., rendering a scene
  – an artist may search for material that other people have edited/produced
  – the view of what data are stored in the system may be different depending on the application and user
Vision

capture and use **semantic information** for …

- fast searching and retrieval of data
  - combine with exiting techniques for searching (indexing, semantic vectors, content-specific tools)
  - improve completeness and accuracy of searching

- efficient data storage
  - data compression based on semantic relations of data

- improved performance
  - data hoarding, data placement and replication/caching
  - efficient rich media distribution and streaming

- highly available data sharing
  - balance consistency & availability according to semantics
  - sophisticated access control and security
Challenges

- What are the common semantic relations of interest?
- How to capture semantic information?
- How to handle dynamic evolution of semantics?
- What are the basic tools and APIs users/applications require?
use a generic data model to capture semantic metadata
  – handle dynamic evolution/elaboration of schema
  – more generic than the existing semantic file systems
    • use RDF (semantic web) for semi-structured data
    • basic relations: similarity (e.g., versions), dependencies, associative, context

extract the commonality of data management applications in a generic framework
  – the data model, basic mechanisms and APIs make building customized solutions easy
    • event model, customized namespaces, security, searching, archiving