Comparing P2P Systems

Anthony D. Joseph
John Kubiatowicz
CS294-4

Why so many systems?

- Many different types of target users
- Many different types of environments
- Many design choices
- Many hazards
- Many data types
- Many ….

Networks

- Chord
- CAN
- Tapestry
- Pastry
- Kademlia
- Viceroy
- Bamboo
- …

- Similar interfaces
  - DHT, DOLR
- Different design goals
  - Locality, Topology
  - Fault-tolerance

Systems We've Read About

- Freenet
- Publis
- SFS
- Bayou
- FARSITE
- Logistical Networking
- Pangaea
- Pastiche
- Gia
- OceanStore
- PAST
- Squirrel
- CFS
- Ivy
- PeerDB
- PIER

Systems 1

- Freenet
  - Anon, cens. resistant storage
  - Objects ref'd by SHA-1 hash over content (GUID-CHK)
  - Objs named by GUID-Signed Subspace Key pointing to CHKs
  - Steepest Hill Climbing query routing with TTL
  - Space allocated by popularity
  - Power-law node degrees
  - Tolerates up to 30% failure
- Publis
  - FT, anon, censorship resistant storage
  - Tamper evident, src anon, updatable, deniable
  - Persistent, extensible
  - Splits enc key into k shares
  - Retrieve k shares for content
  - Static mapping of share locations to servers
  - Indirection-based (file) update mechanism vulnerable to server compromise

Systems 2

- SFS
  - Auth, secure, encrypted client-server storage and access control
  - ACL-based auth of individuals, groups, and groups of groups
  - Caching for speed and availability
- Bayou
  - Replicated P2P DB
    - Atomic operations
    - Whole DB replication
  - Operation-based updates
  - Tentative local commits enforced by primary global commit
    - Apps ctl data view
  - Gossip-based info propagation
  - Merge procedures for per-write conflict resolution

Systems 3

- FARSITE
  - P2P storage
  - Max size ~10^5
  - Large-scale read-only sharing, small-scale read/write-sharing
    - Complex lease mechanism
  - Assumes user auth infra
  - Byzantine ring formed for each namespace
  - Reliability and availability through whole file replication
- Logistical Networking
  - Network storage layer
  - IBP: unreliable, transient byte-arrays on depots
  - Aggregation into exNodes
    - Can implement arbitrary reliability mechanisms
    - Analog to Unix inodes

Systems 4

- Pangaea
  - Server-based replication
  - Assumes trusted servers
  - Two-levels of servers:
    - Gold
      - Fully connected clique
      - Strong maintenance
    - Bronze
      - Limited connectivity
  - Last writer wins conflict resolution
- Pastiche
  - P2P data replication for whole machine backup
  - Built on Pastry
  - Enc storage of immutable chunked data
  - Network distance or coverage based buddy choices

Systems 5

- Gia
  - Modified Gnutella protocol
  - Argues against DHTs for this search type
    - Transient P2P clients
    - Keyword-based searches
    - Searching for hay instead of needles
  - Capacity-based topology adaptation
  - Flow-ctrl for queries
- OceanStore
  - Wide-area CS/P2P replicated, robust, secure, auth data storage
  - Built on Tapestry, Bamboo
  - Byzantine update commit
  - Per-write conflict resolution
  - Erasure coding based replication (robustness) with block caching (performance)

Systems 6

- PAST
  - P2P archival storage model
    - No updates
    - Whole-file storage
  - Tries to balance per-node storage load (assumes ≤ 100x diff)
  - Replica and file diversion to maintain k copies
- Squirrel
  - Decentralized P2P web caching
  - Homestore model: stores content at home and client nodes
  - Directory model: use recent clients

Systems 7

- CFS
  - P2P file storage
    - Lease-based
    - Read-only for clients
    - Publishers can update
    - No explicit delete
  - Built on Chord
  - Storage load-balancing
  - Provably efficient and robust
  - Built on DHASH xface
    - File split into blocks
    - k replication
- Ivy
  - R/W P2P file storage
  - Log-based, built on DHASH
  - Snapshot and view-based approach
  - User control over consistency/serialization

Systems 8

- PeerDB
  - On path to a P2P DB
  - No global schema
  - Incomplete replication
  - Dynamic reconfiguration
  - Requires small subset of persistent servers
- PIER
  - P2P DB
    - Built on CAN and others
  - Relaxed consistency
  - Scalable with namespace model
  - Std schemas
  - Several join schemas

Evaluation Metrics

- Commit model (e.g., primary, group, all)
- Information propagation model (e.g., flood, epidemic, multicast)
- Topology
- Search model (e.g., targeted, flood, epidemic, multicast)
- - Expressiveness
- - Information placement/autonomy
- Scaleability
- Target user?
- Reliability / robustness (i.e., data that is eventually available)
- Availabililty (i.e., data that is always available)
- Quality of service
- Anonymity/privacy
- Censorship-resistance
- Publisher/Server deniability
- File integrity
- File authenticity

Metrics from class

- Maintainability / Manageability
- Topology
  - Roles: client, supernode, server
- Defense against selfish/ malicious behaviors
  - Denial of svc resilience
- Scope of knowledge
- Needle vs Hay
  - False negatives
- Static resilience vs MTTR
- Performance under churn
- Emergent behaviors
- Non-data services
  - GRID computing
- Trust model (physical vs virtual)
  - Authentication
  - Authorization
  - Admission control
  - Integrity
- Node heterogeneity
  - Function, capabilities, ownership, dynamic election/configuration
- Indirection between obj lookup and routing
- Application semantics used in routing
- Data type / structured data