Toward Eidetic Distributed File Systems

Xianzheng Dou, Jason Flinn, Peter M. Chen

UNIVERSITY OF MICHIGAN

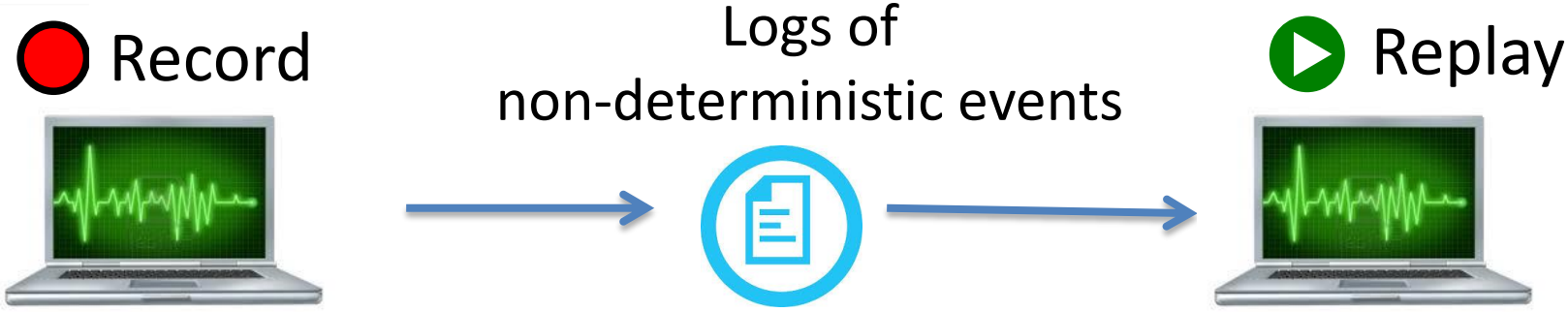
Rich file system features

- Modern file systems store more than just data
  - Versioning: retention of past state
  - Provenance-aware: connections between file data

- Problem:
  - High costs for providing these rich features
    - Past versions in finer granularity → Higher storage costs
  - To provide stronger support at reasonable overheads?
    - Recall any past user-level state
    - Provenance at the byte granularity

A fundamental redesign

- Responsibility of both FS and OS
- Eidetic systems
  - By pervasive deterministic record and replay



Eidetic distributed file systems

Substitute computation for data

Fundamental unit

- What is the fundamental unit of persistent storage?

Fundamental unit

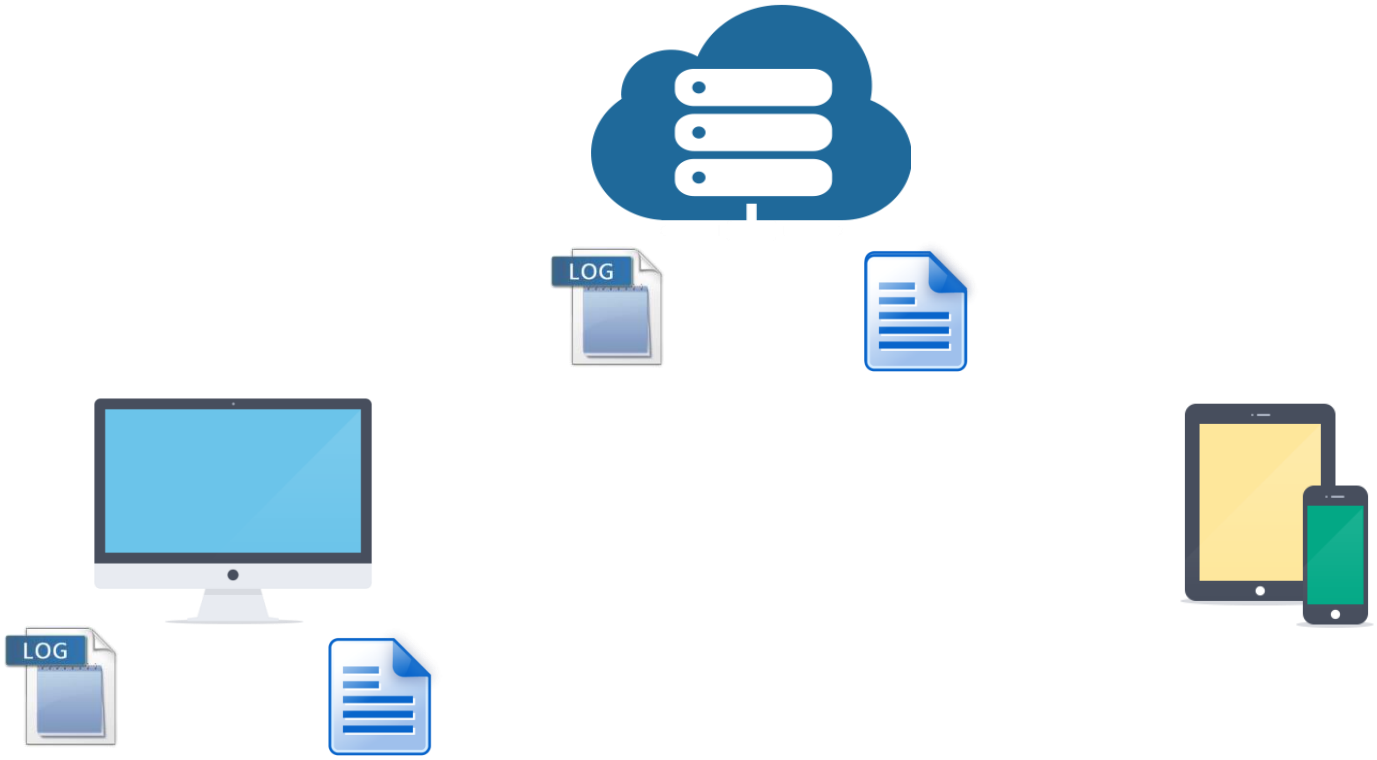
- What is the fundamental unit of persistent storage?

Fundamental unit of persistent storage: ***Logs of non-determinism***

Files are only considered as caches

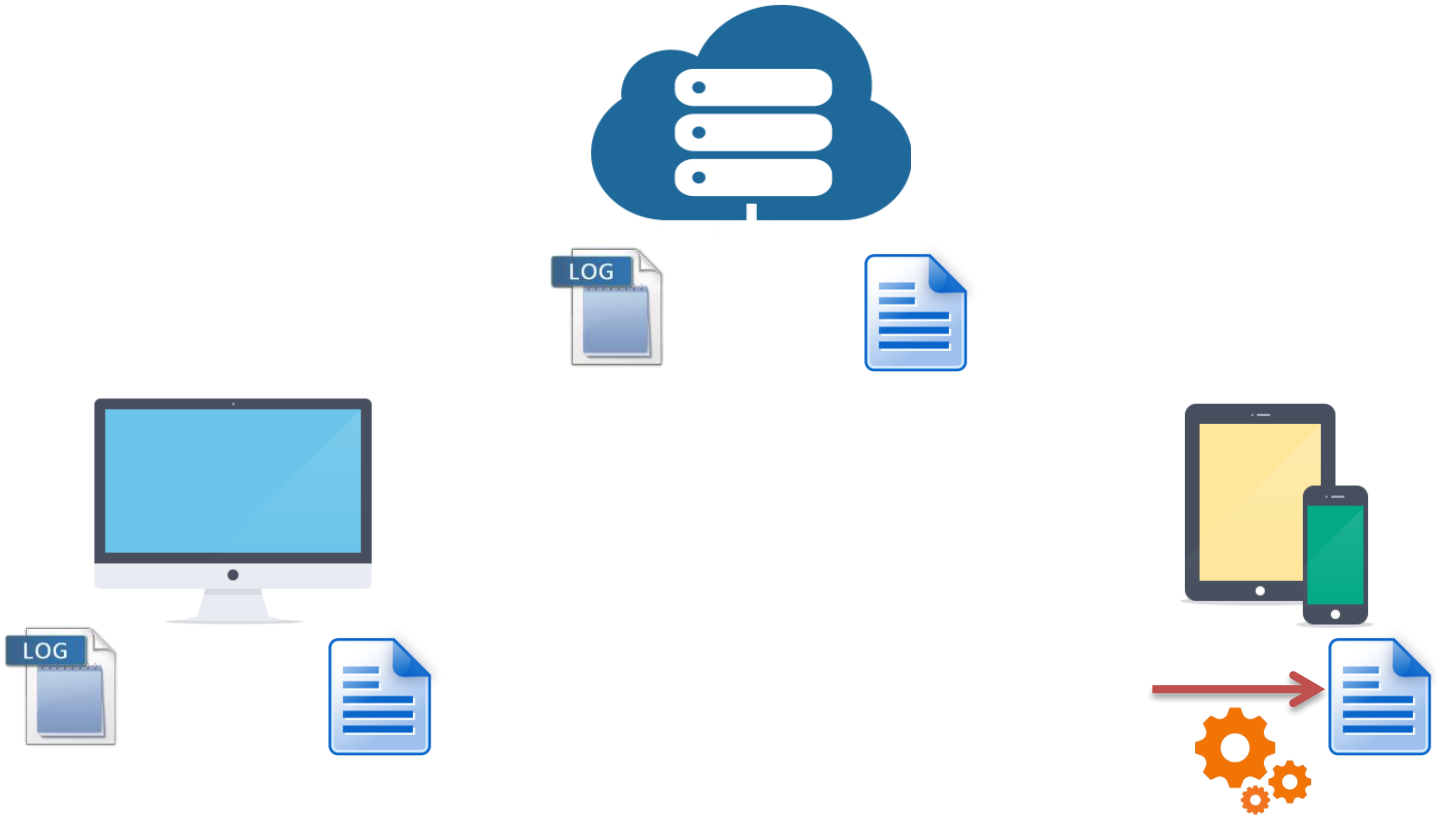
Available transfer methods

- How should a client read a particular version?



Available transfer methods

- How should a client read a particular version?



Available transfer methods

- How should a client read a particular version?

Computation costs
Communication costs
User waiting time
Energy consumption

Conclusions

- Stronger support of versioning and provenance
  - at a lower cost compared to traditional designs
- Reduce communication costs
  - By substituting computation for data

Thank you!

A clean-sheet design of FS

- Eidetic systems prototype
  - Graft eidetic support onto an existing FS
  - Consider only local storage
- An eidetic distributed file system
  - A small number of personal devices + cloud servers
- New design choices
  - Fundamental unit of persistent storage
  - File transfer

File persistency

- When is a file considered persistent on the server?

As long as logs generating the data are persistent

Updating server cache

- Should the server cache the file version?

Probability of future access
Costs for regeneration

Feasibility

- Eidetic system overheads
  - Record 4 years of workstation data on a 4TB hard disk
  - Under 8% performance overhead on most benchmarks
- Applications (log size vs. diff size)
  - Logs are smaller
    - image/audio editing, latex, make, slides editing
  - Diffs are smaller: text editing
- File sharing
  - Most files are not shared

Available transfer methods

- How should a client read a particular version?

By value
By replay on the client
By replay on the server
From peers