CSE 124 Demo: Scalable SuperProxy with Distributed File System

Tassapol Athiapinya - A50042456
Edwin Makiuchi - A07668278

SuperProxy

- Web-based HTTP proxy
- A user inputs target URL.
- The proxy returns cached/non-cached contents.
- Scalability depends on size of cache.

Solving Cache Limitations

- Hardware
  - Huge RAM
  - Flash drive
  - External hard drive
  - RAID
- Software
  - Regularly deletes data
  - Caches only some data

Solving Cache Limitations (2)

- File system across multiple machines
  - Google File System (like we talked in class)
- Our solution
  - Hadoop
    - MapReduce (for CPU scalability)
      - beyond our scope
    - Hadoop Distributed FS (for file system scalability)
      - What we use!
      - Very close to Google FS

HDFS Client Request

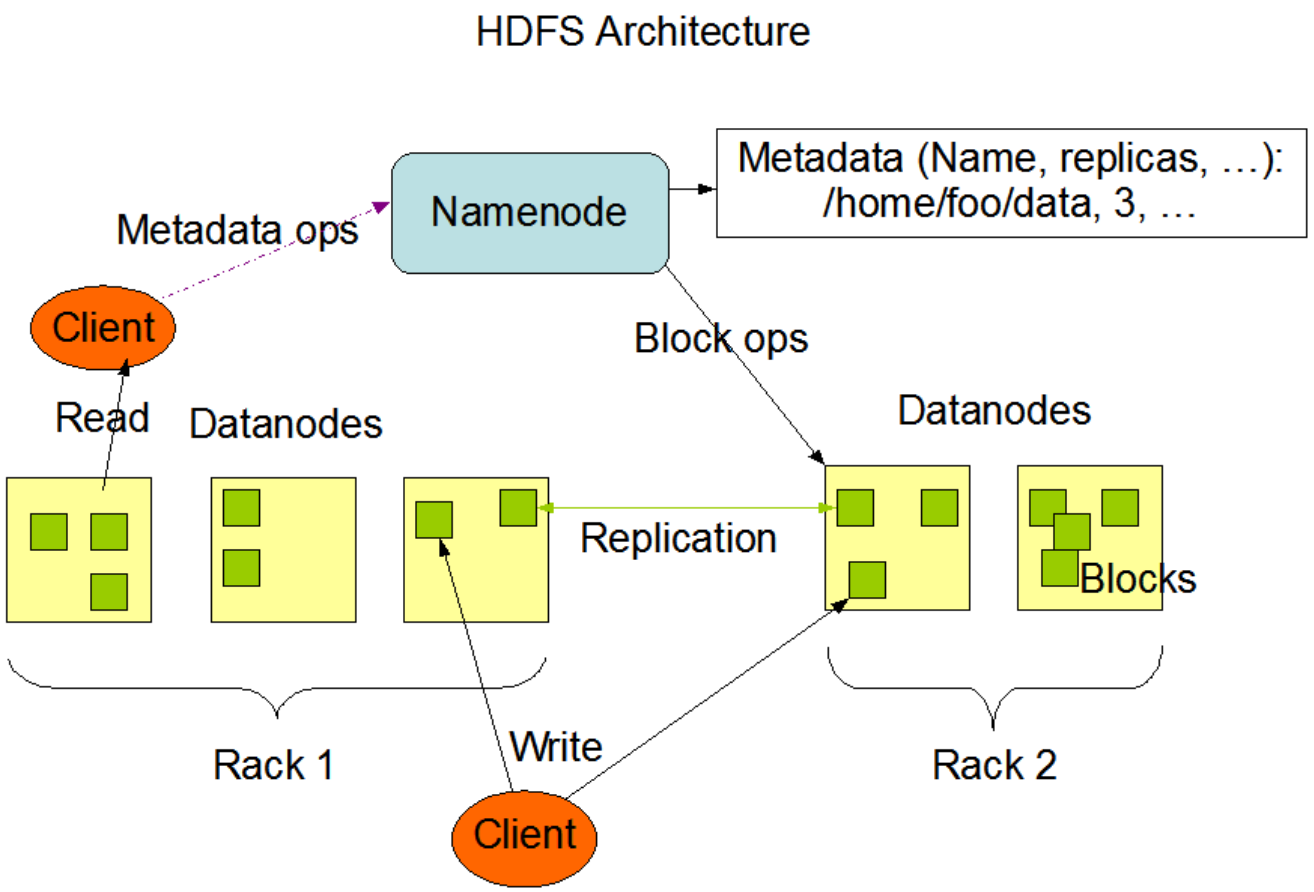


Image from http://hadoop.apache.org

HDFS File Chunks

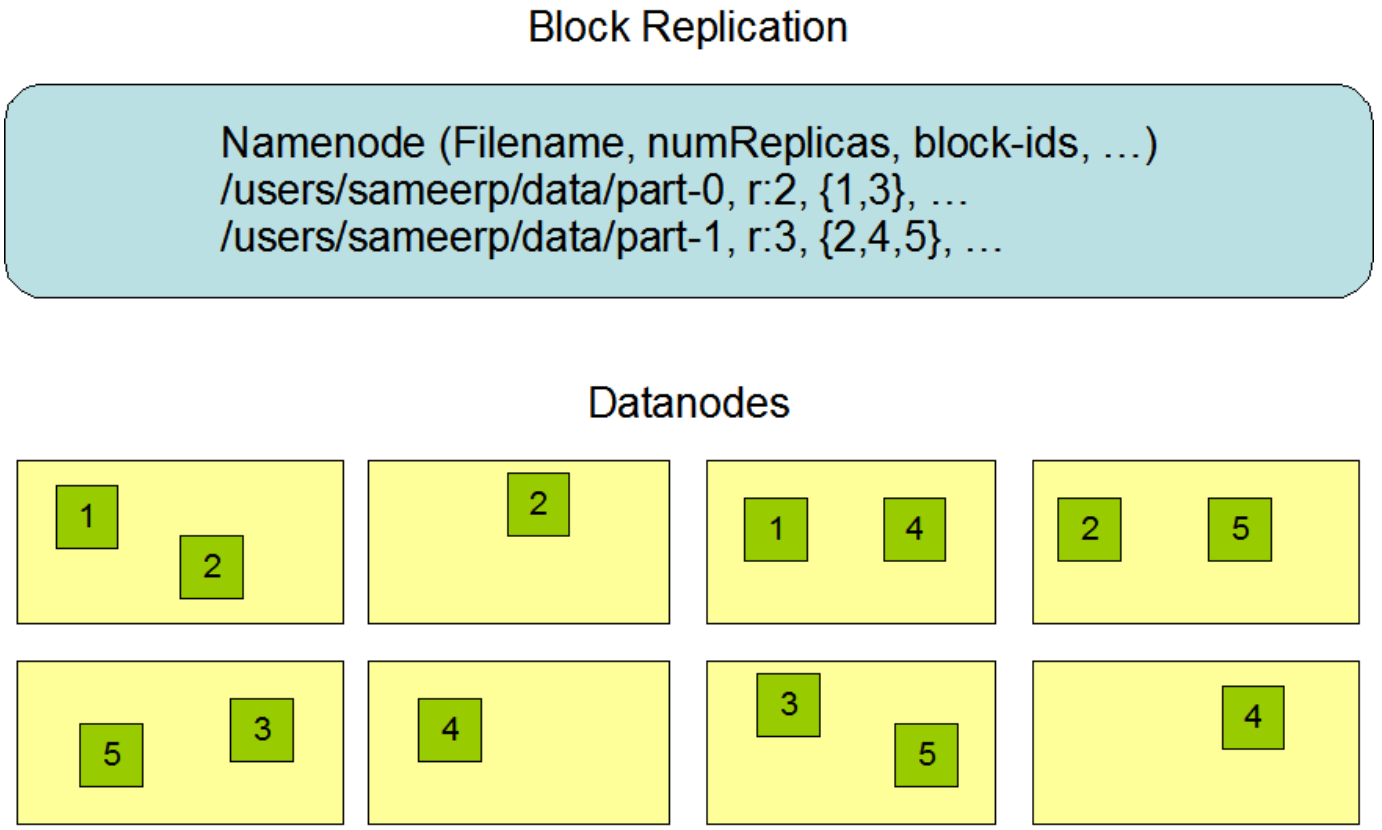
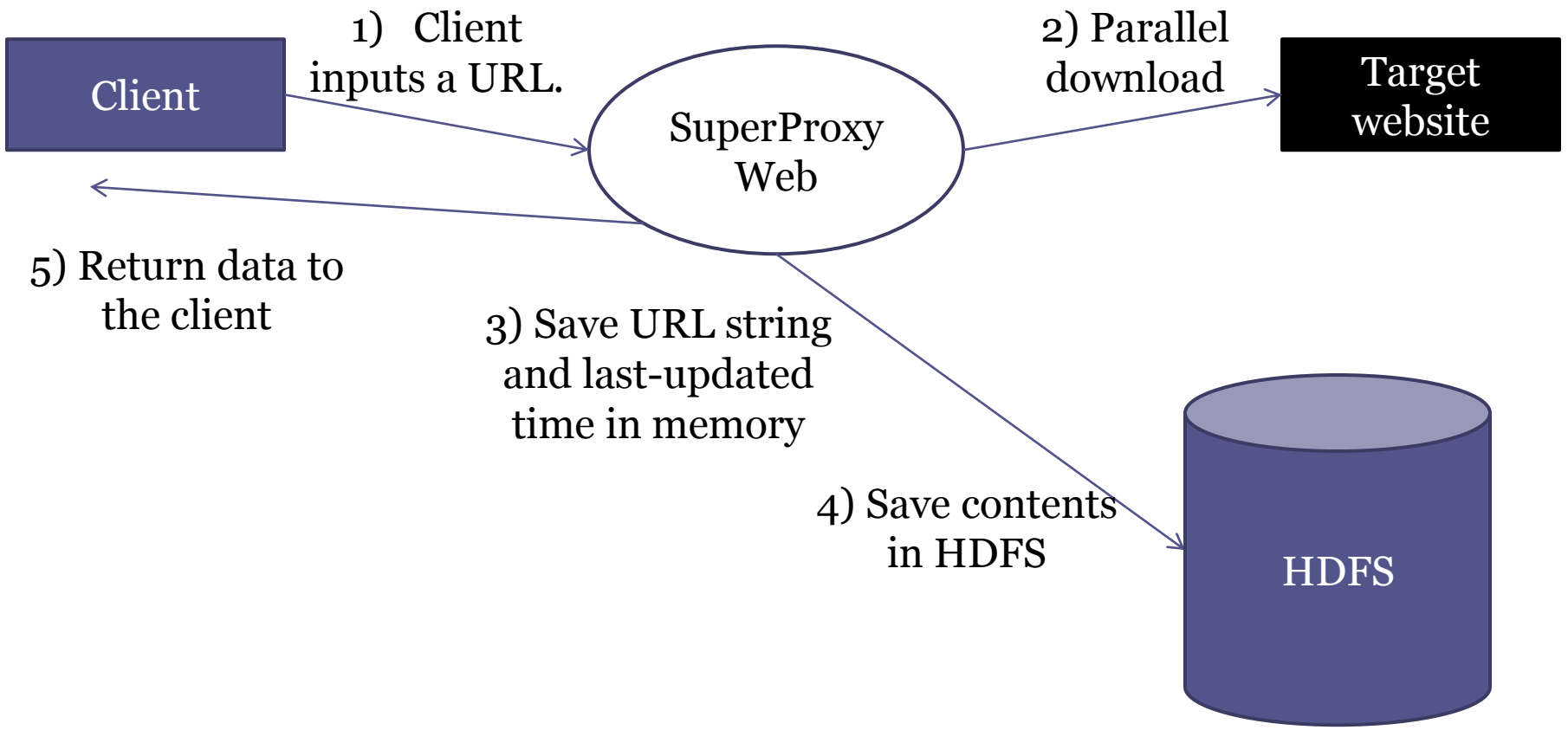


Image from http://hadoop.apache.org

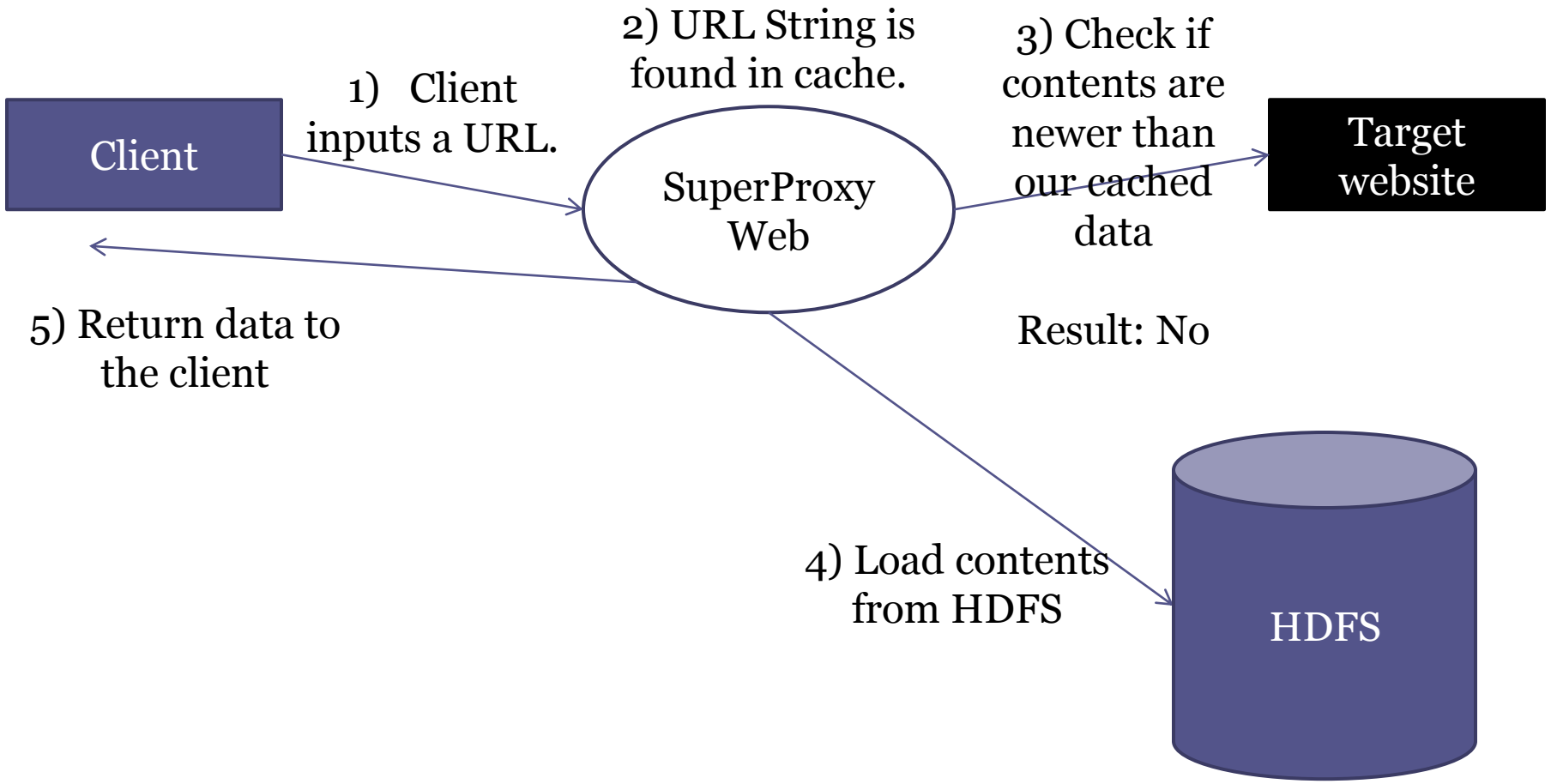
Our SuperProxy

- Tomcat
- HTTP Servlet
- URL rewriting
  - Rewrite html links and image links to request data from our server
  - From:
    http://cseweb.ucsd.edu/classes/fa10/cse124/ucsd_logo.gif
  - To:
    http://216.24.193.224:8080/cse124demo/CheckCache?url=http://cseweb.ucsd.edu/classes/fa10/cse124/ucsd_logo.gif

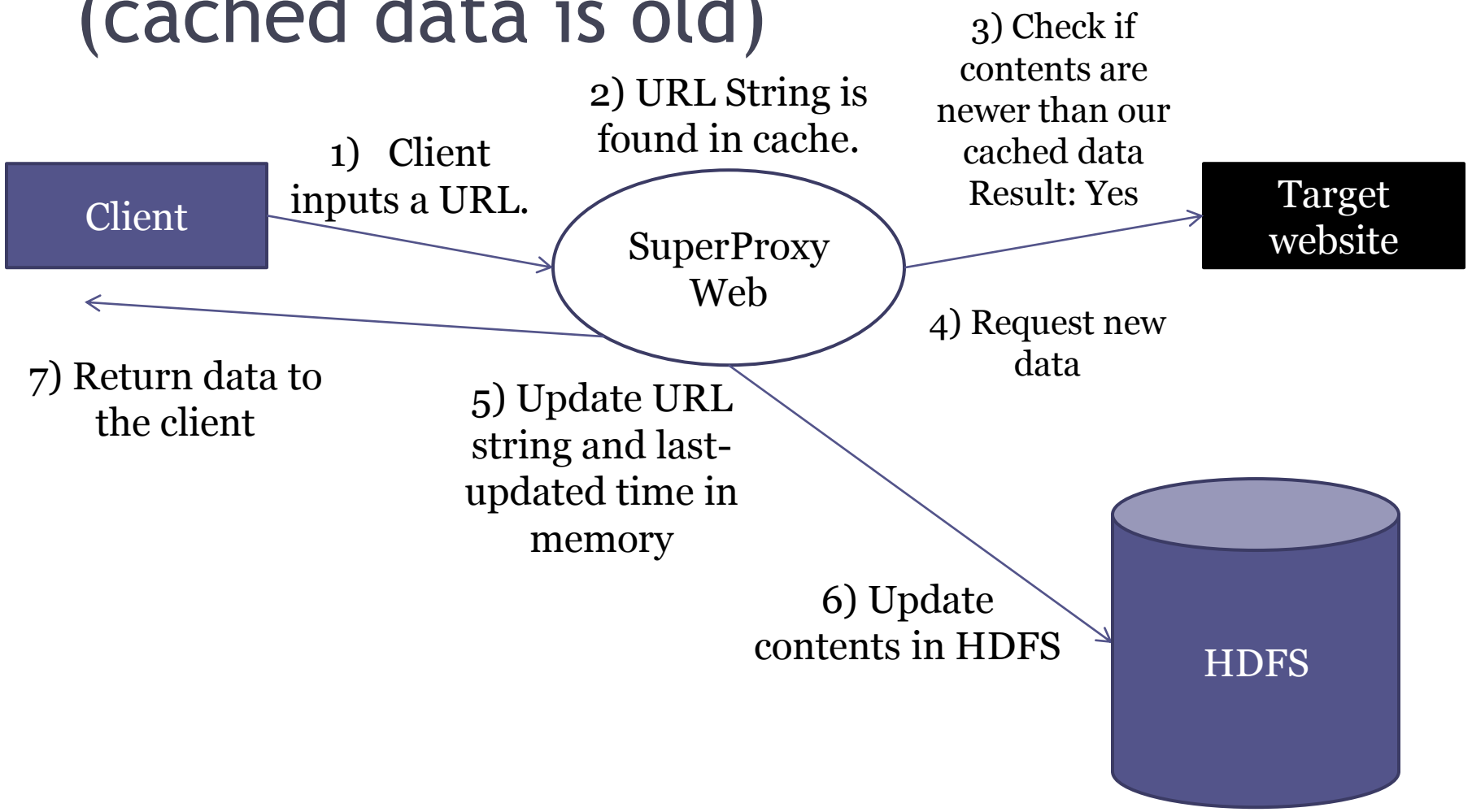
Our SuperProxy Workflow (non-cached)



Our SuperProxy Workflow (cached data is latest)



Our SuperProxy Workflow (cached data is old)



Demo

- Live demo at

http://216.24.193.224:8080/cse124demo/