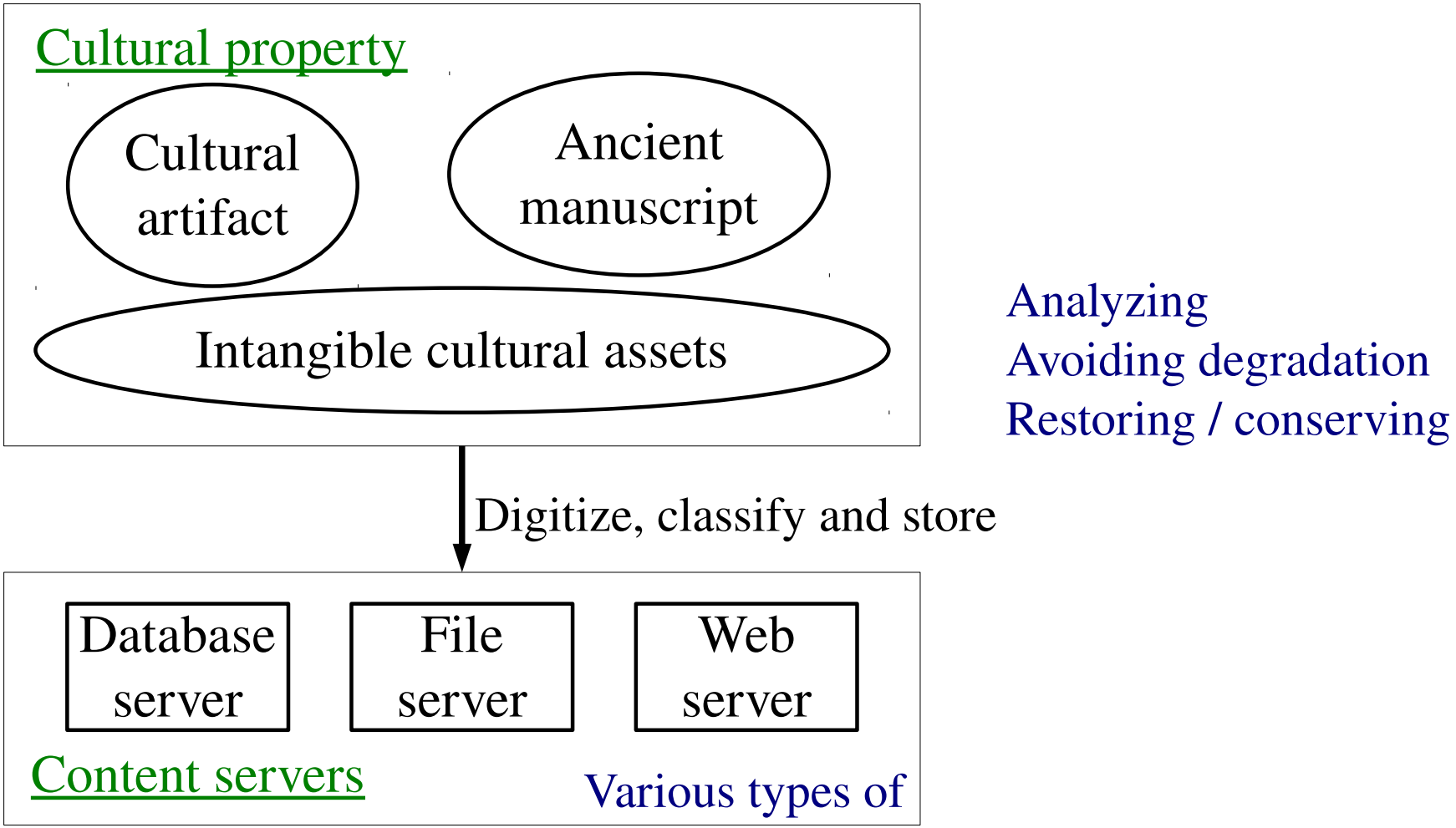
A Content Management Method by Integrating Distributed Content Servers

Masahito Shiba

Ryukoku University

Introduction

- Creating digital records and storing them are important techniques in research of cultural property.



Introduction

- Digital archives, digital museums
  - Various types of contents
  - Various types of content servers
- Expertise of both of contents and systems are required
  - Some sites have lack of experts.
- Sometimes contents servers are not well designed
  - Running servers for each dataset
  - Efficient use of contents in different servers are prevented

Introduction

- To support management of content servers by techniques of system software
- The proposed system
  - integrates various types of content servers
  - supports dynamic addition/deletion of content servers
  - provides efficient methods to use distributed contents
- using P2P network techniques

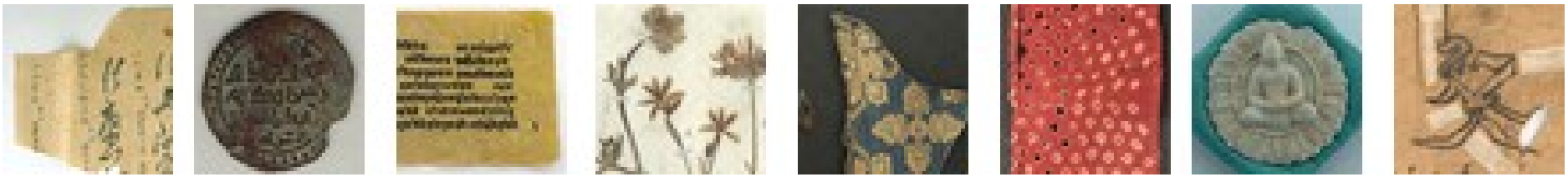
P2P technology

- P2P overlay network
  - to provide flexible communication function
- DHT (Distributed Hash Table)
  - to keep metadata of massive contents

- JXTA
  - A set of open protocols for P2P applications
  - There are some implementations for programming languages
    - a implementation for JAVA

Contents

- Contents of sites of digital archives have some features
  - Range of data size is very wide
  - Various types of data: text, image, movie
  - Contents are often added
  - Deleting or modifying does not often occurred

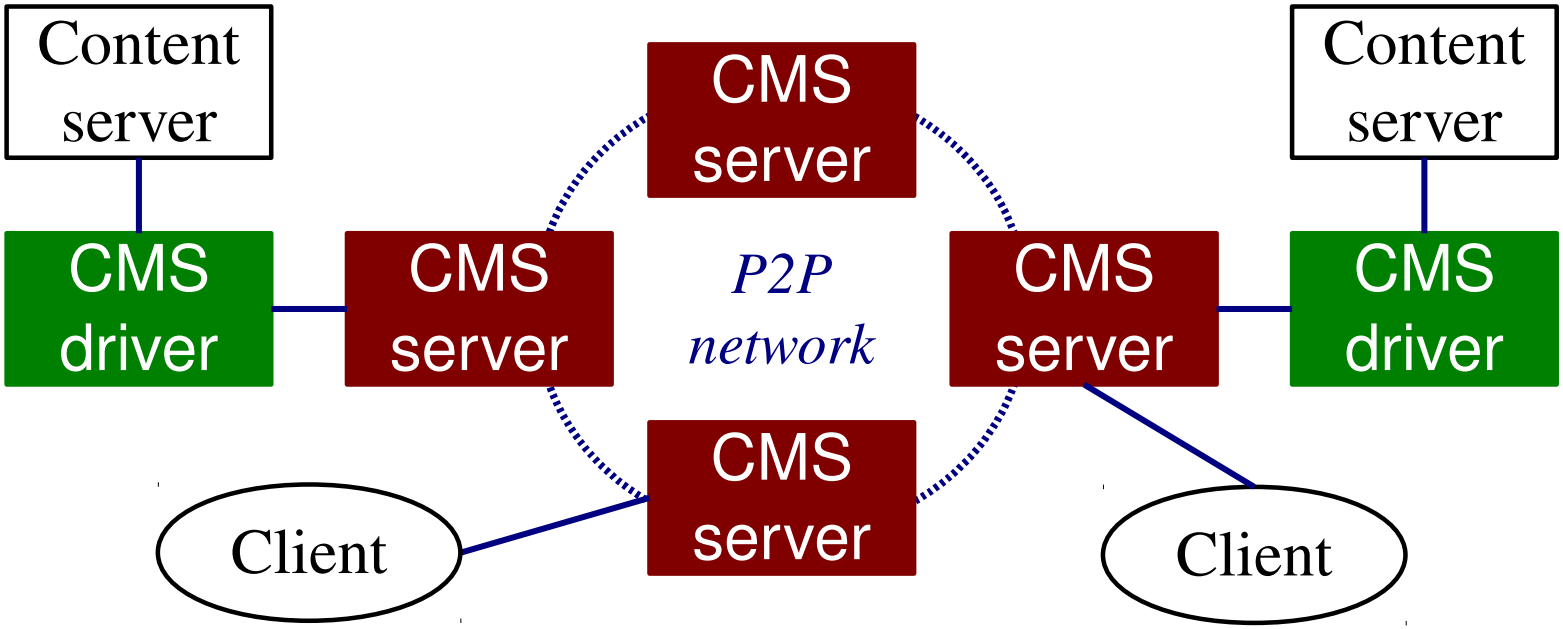


Related works

- There are some studies for such data
- Protocols for exchanging metadata of contents
  - may violate existing server layer
  - OAI-PMH
- CMS over a DHT
- Look-up records in a DHT with complex condition
  - regular expression

Overview

- *Content server* - existing servers
- *Client* - applications using contents in content servers
- *CMS server* - managing content servers and their contents
- *CMS driver* - handling content servers

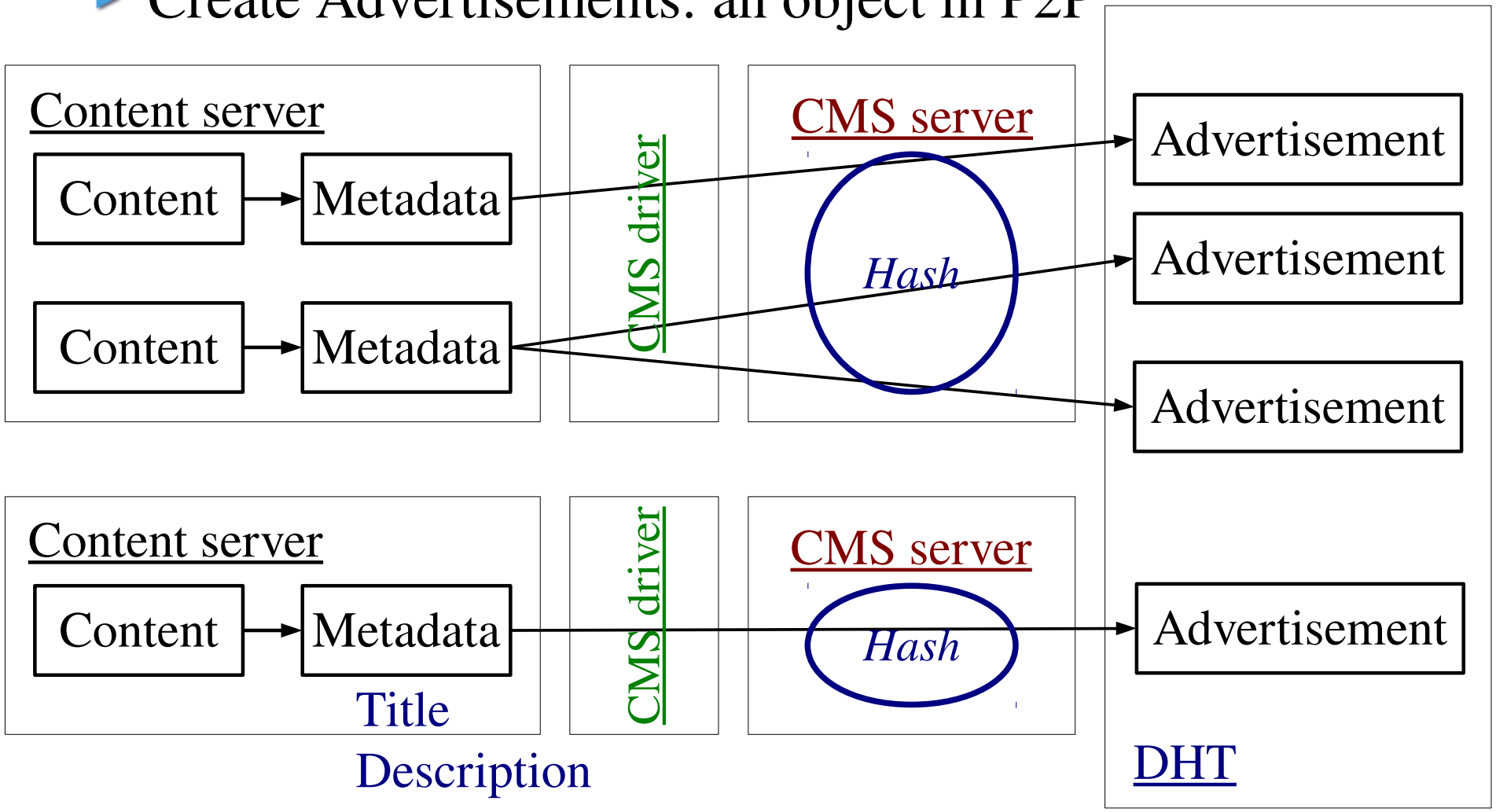


System structure

- CMS servers
  - communicate with each other
  - build a P2P overlay network and a DHT
  - provide functions of looking up and getting contents
- CMS drivers
  - Each CMS driver is bound to a content server.
  - communicates with corresponding content server
  - controls the content server according to requests from the CMS servers

Keeping content data in a DHT

- CMS servers register metadata into the DHT
  - Get content data using CMS drivers
  - Create Advertisements: an object in P2P



Look-up contents

- Client
  - connects to one of CMS servers
  - send a request of look-up
- CMS server
  - looks up advertisements that match the request by communicating with the other CMS servers
  - sends content data created from the matched advertisements to the client.
    - metadata including the location of the content

Add a content server

- Add new content servers to the system dynamically
- CMS server
  - creates a CMS driver instance for the content server
  - binds the driver to the content server.
- CMS driver
  - gets content data from the content server
  - sends them to the CMS server.
- CMS server
  - receives the content data and creates advertisements.
  - registers these advertisements into the DHT.

Creating CMS drivers

- CMS drivers access content servers
  - database server
  - file server
  - web server
- Add a new type of content server by creating a CMS driver

- The system administrators create CMS drivers as their needs
  - CMS drivers are reused easily with small fix
  - Creating a CMS driver is similar to creating a client of the content server

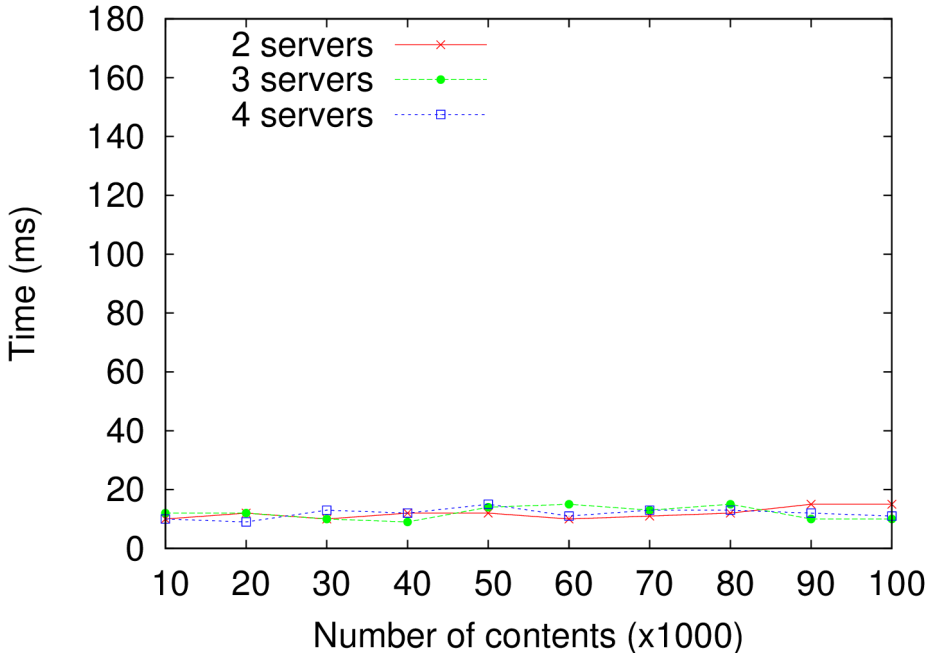
Look-up contents

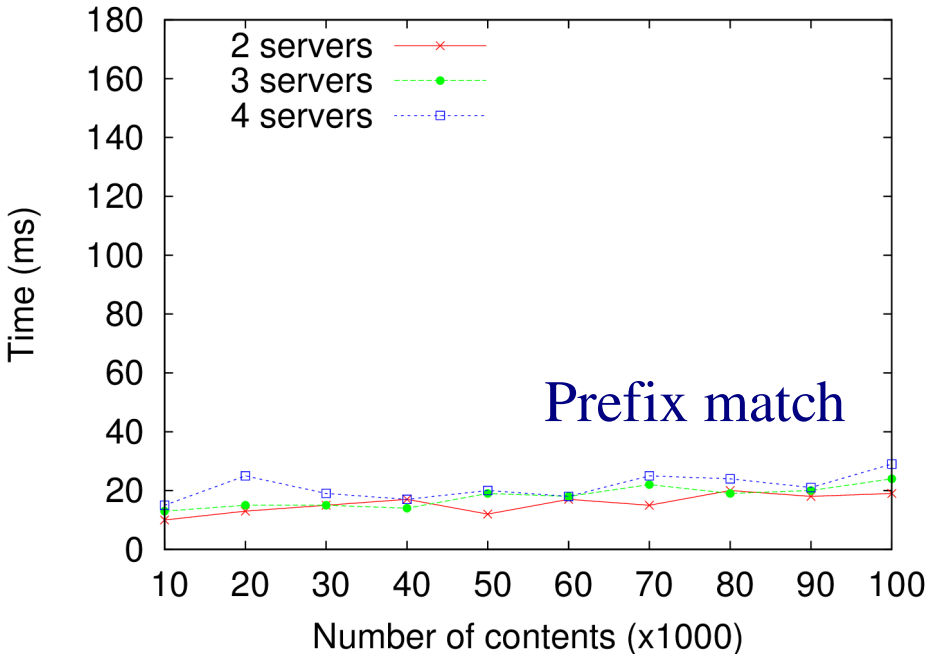
- The system manages many contents
  - Handling contents in two or more contents servers
  - Scalability is important
- Execution time when a client looked up contents.
  - Sending a request to a CMS server and getting a result
  - Contents were character strings eighteen characters or less
  - Each content server had from 10,000 to 100,000 contents

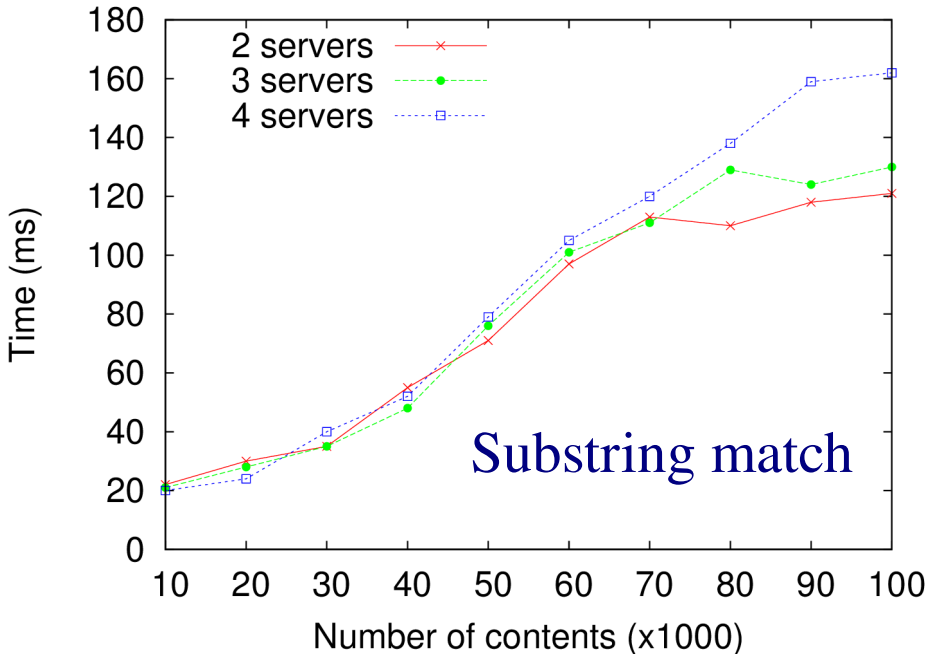
Environment
- Pentium4 2.4 GHz, 1000 Mbps Ethernet
- Linux 2.6.27, Java SE 6, JXTA 2.5

Look-up contents

- Execution times increased slightly in the exact match and the prefix match.
- The number of content servers did not affect the execution times very much.

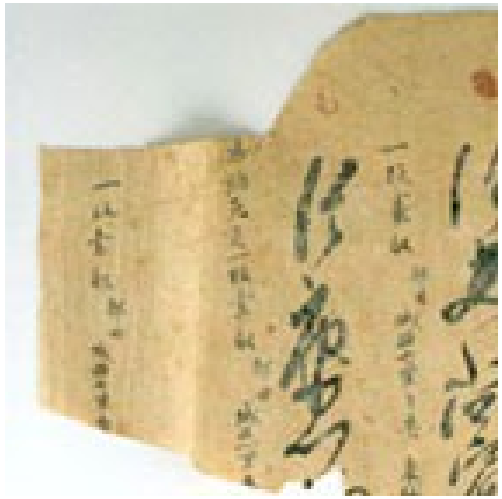






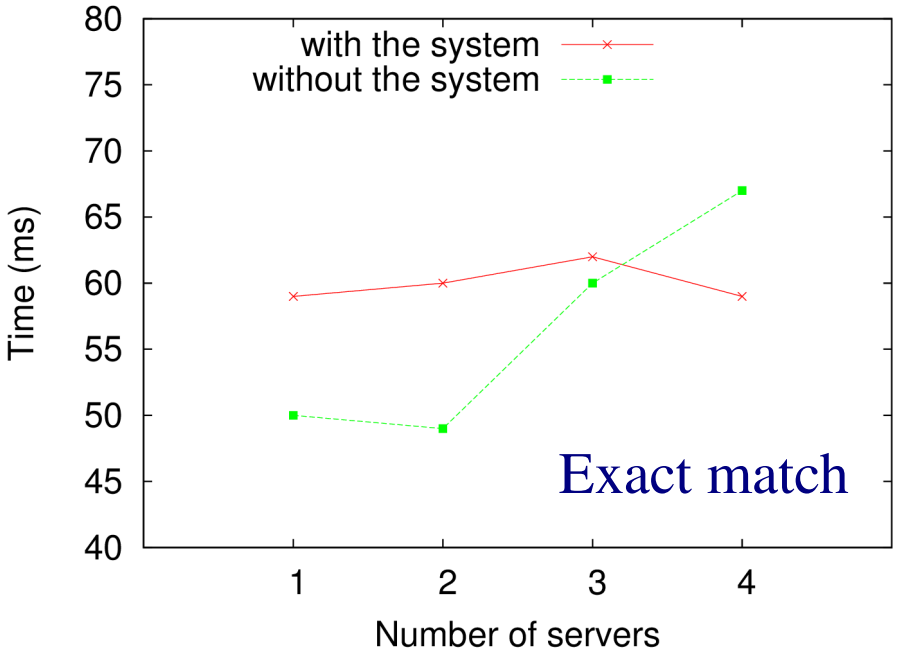
Retrieving contents

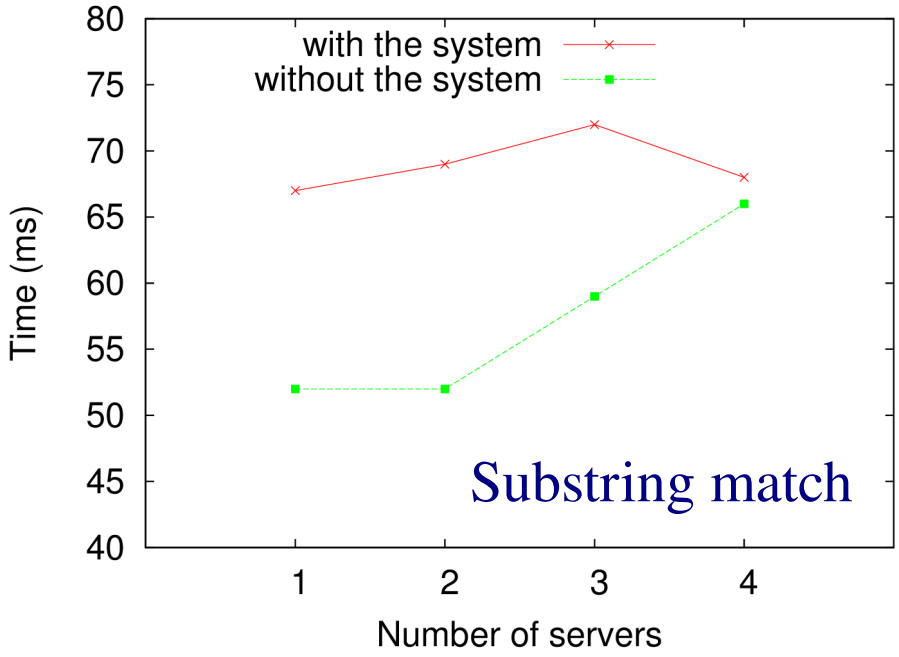
- Overhead of the system
  - using a P2P network
  - handling contents through the system
- Retrieving contents with/without the system
  - Contents were captured images of ancient manuscripts
  - about 10 KB
    - small data for measuring the overhead



Retrieving contents

- *Without*: The client sent queries directly to content servers and got a content matched to them.
- *With*: The client sent a query to a CMS server and got a content matched to it.





The increase of the execution time is a little when the client uses the system.

Conclusion

- A P2P based content management system
  - manages content servers and their contents
  - integrates various types of content servers
  - provides efficient methods to use contents in different servers