

SciServer

Entwicklung einer Serversoftware zur Verwaltung von
wissenschaftlichen Daten

Jens Kapitza

November 2015

Gliederung

- 1 Einführung
- 2 Problemstellung und Lösungsansätze
- 3 Strukturen und Kommunikationsmuster im Lösungsansatz
- 4 Version 2.0
- 5 DEMO (interaktiv)

Thema

Wissenschaftliche Daten

- Sehr große Dateien
- Unbekannte Dateitypen
- Öffentlich, Lizenzen/Patente
- Sehr viele Dateien

Aktionen

- Anlegen
- Verwalten (Attribute, Schlagwörter)
- Verteilen
- Suchen

Gliederung

1 Einführung

2 Problemstellung und Lösungsansätze

- Einführung
- Cloud-Ansatz
- „private Cloud“-Ansatz
- Dokumenten Management System (DMS)
- Peer-to-Peer Ansatz

3 Strukturen und Kommunikationsmuster im Lösungsansatz

4 Version 2.0

5 DEMO (interaktiv)

Probleme

Betriebssysteme

- nicht einheitlich (Programme und Benutzerinterfacekonzepte)
- unterstützen nicht alles (Schlagwortbegrenzung, Dateigrößen)
- Protokolle weichen untereinander ab (Limitierungen HTTP)

Ziele

- Verwaltung und Suchen über Tags und Attribute
- Vorverarbeitung wie Prüfsummen, Filtervorgänge
- Benachrichtigungen
- Verteilen von Dateien

Anforderungen

- Plattform unabhängig
- Weboberfläche zum Suchen und Verwalten
- Einfach, einfach, einfach.

Java-Stack

- JSF, JMS, JPA, CDI
- JVM, Vorteil oder Nachteil?

Cloud-Ansatz

Möglichkeiten

- Dropbox
- Windows One Drive
- Google Drive, ...

Optionen

- Verteilung der Daten
- Zentrale Speicherung
- Weboberfläche und ggf. Client

Bedenken

- | | |
|---------------------|----------------------------|
| ■ Rechtlich? | ■ Informationen teilen? |
| ■ Datenspeicherung? | ■ Informationen verwalten? |
| ■ Unterstützung? | ■ Informationen suchen? |
| ■ Kosten? | ■ Informationszugriff? |

„private Cloud“-Ansatz

Möglichkeiten

- Owncloud
- Horde
- FTP, HTTP, NFS, ...

Optionen

- Verteilung der Daten (manuell)
- Zentrale Speicherung
- Weboberfläche meist keine Clients

Bedenken

- | | |
|---------------------|----------------------------|
| ■ Rechtlich? | ■ Informationen teilen? |
| ■ Datenspeicherung? | ■ Informationen verwalten? |
| ■ Unterstützung? | ■ Informationen suchen? |
| ■ Kosten? | ■ Informationszugriff? |

Dokumenten Management System (DMS)

Möglichkeiten

- openKM
- Alfresco One

Optionen

- Verteilung der Daten
- Zentrale Speicherung
- Weboberfläche und spezielle Clients als Addon

Bedenken

- | | |
|---------------------|----------------------------|
| ■ Rechtlich? | ■ Informationen teilen? |
| ■ Datenspeicherung? | ■ Informationen verwalten? |
| ■ Unterstützung? | ■ Informationen suchen? |
| ■ Kosten? | ■ Informationszugriff? |

Peer-to-Peer Ansatz

Möglichkeiten

- Napster, ...
- SciServer

Optionen

- Verteilung der Daten
- **Dezentrale Speicherung**
- Weboberfläche und Agenten

Bedenken

- | | |
|---------------------|----------------------------|
| ■ Rechtlich? | ■ Informationen teilen? |
| ■ Datenspeicherung? | ■ Informationen verwalten? |
| ■ Unterstützung? | ■ Informationen suchen? |
| ■ Kosten? | ■ Informationszugriff? |

Gliederung

1 Einführung

2 Problemstellung und Lösungsansätze

3 **Strukturen und Kommunikationsmuster im Lösungsansatz**

- Strukturen für Peers / Smart-Hosts / Proxys
- Agenten im Peer-to-Peer
- Systembefehle und Arbeiten mit Dateien

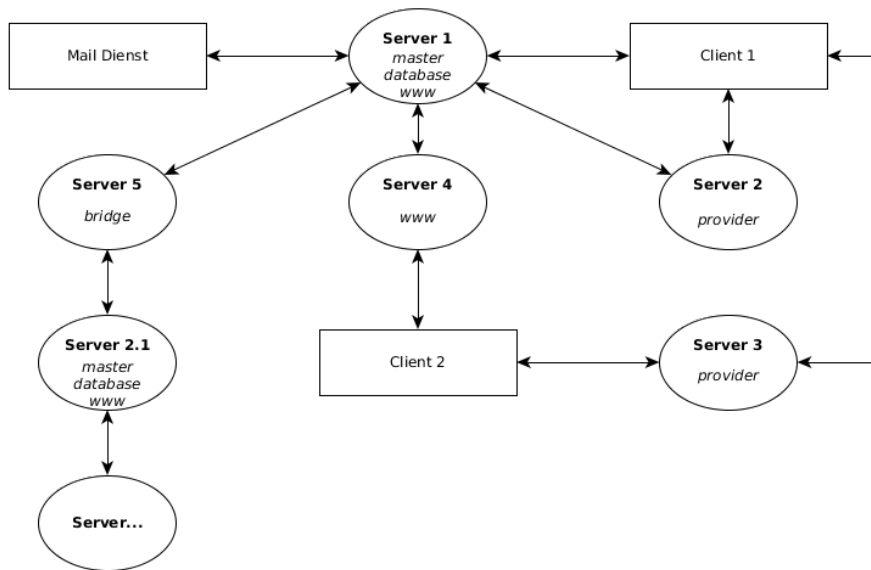
4 Version 2.0

5 DEMO (interaktiv)

Strukturen

- Client-Server** Weboberfläche nutzt HTML und HTTP. Diese bindet Browser an.
Mailclient wird durch die „database“ und den Benachrichtigungen durch IMAP und SMTP angebunden.
- Peer-to-Peer** einfach: Jeder redet mit jedem und ist jedem anderem gleich gestellt.
Aber: einzelne Peers erledigen als Agenten bestimmte Aufgaben.
Das System kann einfach wachsen, indem weitere Anwendungen gestartet werden.
- Publish-Subscribe** Kommunikation wird an alle angemeldeten Peers verteilt. vgl. Bus-System, Observer-Pattern, Vermittler

Agenten im Peer-to-Peer



Arbeiten mit Dateien

Java-Stack

Files, Path Zugriff auf die Informationen einer Datei

WatchService Überwachen der Änderungen in einem Ordner
Aber: nicht wer die Änderung gemacht hat (Prozess),
sondern nur, dass Änderungen passiert sind.

Prozesse Plattform unabhängig, Aber kaum Informationen über
Prozesse 'generalisiert' möglich.

Netzwerk, JMS Nutzung von Javas interner Serialisierung.
Bibliotheken haben Memory-Leaks und Abstürze.

TODOs

- Prozesse** Möglichkeit bieten, auf Informationen der gestarteten Prozesse sowie deren Fortschritt zu verfolgen.
- Netzwerk** Kommunikation anpassen auf SSH.
Einen eigenen Server bereitstellen (mit Benutzer/Peer Verwaltung) unabhängig vom Betriebssystem.
- Web** Migration auf einen kleineren Webserver.
Ein Applicationserver bietet mehr als die Anwendung braucht, jedoch ein einfacher Webcontainer zu wenig.

TODOs

WatchService Versuch herauszufinden welcher Prozess die Änderungen gemacht hat.
Bessere Benachrichtigungen.

Mail-Client Einen eigenen Agenten für Benachrichtigungen und Interaktion mit dem System schreiben.

Client-Anwendung Viele Funktionen können nicht ohne Desktop-Anwendung ausgeführt werden, die Sicherheitsbeschränkungen in HTML können je nach Browser nur schwer umgangen werden. Das ist nicht praktikabel! Daher ist eine Desktop-Anwendung besser für einige Funktionen geeignet.

DEMO

- Starten des Master, Database, Provider und WWW
- Suchen einer Datei
- Taggen einer Datei
- Benachrichtigungen (für E-Mail, Internet nötig)
- Ausführung eines Filters/Befehls

Fragen?

*Wer nicht danke sagen kann,
wird irgendwann vergeblich bitten.*
– Fred Ammon, <http://www.aphorismen.de>