

Grundlagen webbasierter Informationssysteme

Seminar im SS 2004

Web-Caching

Sebastian Adam

Gliederung

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Fokussierung

- Was muss ich als Entwickler über Web-Caching wissen?

Antwort

- Situation im Web
- Idee des Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien

14.07.2004

Web-Caching

Folie 2

Situation im Web

- ➔ Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Wirtschaftliche Bedeutung des Internet

- > 4,25 Mrd. Seiten, 22 Mio. Rechner, 34 Mio. dt. Nutzer
- Internet wächst weiter.
- Abwicklung betrieblicher Prozesse im Web nimmt zu.
- Für Privatpersonen werden Seiten zu persönlichen Services.
- Kaufbereitschaft von Endkunden steigt.
- Qualität, Verfügbarkeit und Schnelligkeit sind Erfolgsfaktoren.

14.07.2004

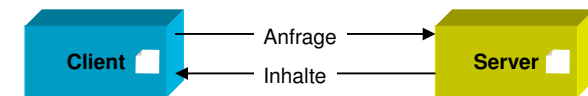
Web-Caching

Folie 3

Situation im Web

- ➔ Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Probleme des klassischen Web-Request



- Antwortzeit (Client)
 - Übermittlung des gesamten Seiteninhalts bei jedem Aufruf
- Skalierbarkeit (Server)
 - Parallele Beantwortung zahlreicher Requests
 - Dynamische Seitengenerierung (z.B. Warenkorb)
- Probleme nicht mit Hardware zu lösen, da
 - Ressourcen begrenzt oder zu teuer
 - Datenmenge wächst schneller als Kapazitäten

Allgemeines Problem

Hauptsächlich bei Top-Seiten

14.07.2004

Web-Caching

Folie 4

Situation im Web

- ➔ Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Beobachtungen

- Nur geringer Anteil des Netztransfers ist anfragespezifischer Inhalt.
- Großteil aller Anfragen auf relativ wenig Webseiten
- Folgerung:
 - Großteil (>60%) des Transfervolumens ist redundant.
 - Eliminierung unnötiger Datenübertragung kann Probleme reduzieren.

Web-Caching als Lösungsansatz

14.07.2004

Web-Caching

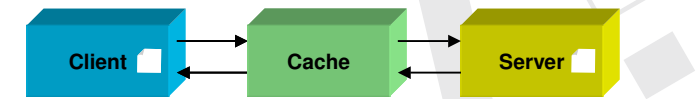
Folie 5

Web-Caching

- Situation im Web
- ➔ Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Idee und Ziele

- Speicherung von Kopien häufig angefragter Inhalte außerhalb des Servers möglichst nahe an den Clients
- Anfrage an Server nur noch für ungespeicherte Inhalte
- Dadurch
 - Kürzere Antwortzeiten
 - Reduzierung der Anfrage- und Rechenlast
 - Reduzierung des gesamten Netzwerktransfers



14.07.2004

Web-Caching

Folie 6

Web-Caching

- Situation im Web
- ➔ Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Anforderungen an einen Web-Cache

- Integration in die Client/Server-Kommunikation
- (Viele) Inhalte sind im Cache vorhanden.
- Inhalte sind stets aktuell (gültig).
- Beachtung von Sicherheitsaspekten

14.07.2004

Web-Caching

Folie 7

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- ➔ Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Platzierung eines Web-Caches

- Im Aufsuchweg zwischen Client und Server
- Ideal: Netzwerkknoten, die bei einem Request in jedem Fall durchlaufen werden
 - Client-Rechner
 - LAN-Internet-Gateway
 - Internet Service Provider
 - ...
- Proxy-Cache vs. Browser-Cache
- Wünschenswert: Cache für Client transparent

14.07.2004

Web-Caching

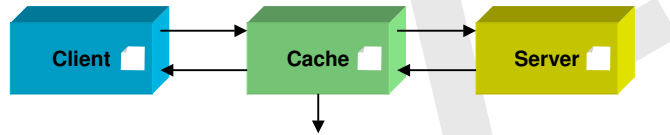
Folie 8

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Einlagerung von Inhalten

1. Vergleich von Datenbestand und Request
 - Gültiger Inhalt vorhanden → Schritt 5
2. Weiterleitung des Request an Server
3. Allokation von Speicherplatz (mittels TTL, LRU).
4. Einlagerung von URL und zugehörige Serverrückgabe als (key, value)-Paar
5. Auslieferung der Inhalte an Client



14.07.2004

Web-Caching

Folie 9

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Zielkonflikte

- Client-naher Cache
 - schnelle Auslieferung
 - Geringe Trefferquote, da nur kleiner Nutzerkreis
- Client-ferner Cache
 - langsame Auslieferung
 - Hohe Trefferquote, da großer Nutzerkreis

Idee

- Hierarchische Anordnung mehrerer Web-Caches
- Verteilung der Aufgaben
 - Client-nahe Caches: Reduzierung der Antwortzeit
 - Entfernte Caches: Reduzierung der Anfragelast

14.07.2004

Web-Caching

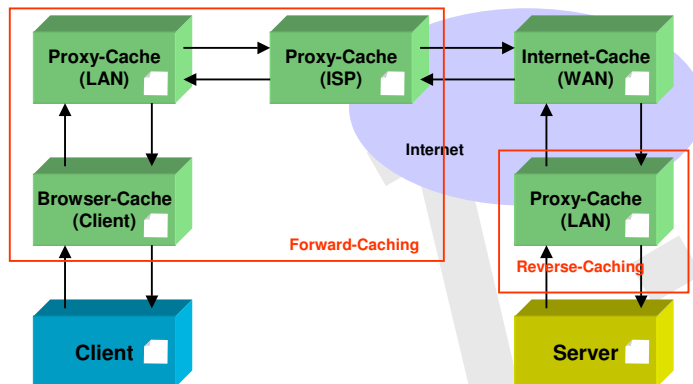
Folie 10

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Hierarchische Organisation

- Beantwortung durch niedrigsten Cache, der Inhalt besitzt
- Anschließende Speicherung in allen tieferen Caches



14.07.2004

Web-Caching

Folie 11

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Gewährleistung der Datenaktualität

- Vom Cache bereitgestellte Inhalte müssen zu jedem Zeitpunkt mit denen auf dem Zielsever übereinstimmen.
- Permanente Rückfrage zu teuer
 - Rückfrage nur bei Bedarf
 - Gültigkeitsfeststellung lokal im Cache
 - » Zusatzangaben und Regeln
- Zusatzangaben in HTTP-Response des Servers
 - Gültigkeitsfrist, Änderungsdatum, ...
- Regeln in HTTP-Response oder in Cache-Konfiguration

14.07.2004

Web-Caching

Folie 12

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Cache-relevante HTTP-Header

- Letzte Änderung (HTTP 1.0)
 - **Last-Modified: Son, 30 May 2004 11:32:19 GMT**
- Gültigkeitsfrist (HTTP 1.0)
 - **Expires: Mon, 31 May 2004 14:19:41 GMT**
- Erweiterte Cache-Kontrolle (HTTP 1.1)
 - **Cache-Control: <Parameter>**
- Beispiel-Parameter
 - **max-age = 3600** Ab Request ist Inhalt 1 Std. gültig
 - **private** Caching nur im Client erlaubt
 - **must-revalidate** HTTP-Regel muss befolgt werden
 - **no-store** Explizites Caching-Verbot

14.07.2004

Web-Caching

Folie 13

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Verwendung relevanter HTTP-Header

- Erstellung der Header-Informationen
 - Webserver-Konfiguration
 - Header-Kommandos in (Skript-)Sprachen
 - **<meta http-equiv>** in HTML
- Kein **Expires** gesetzt
 - Cache schätzt Frist über **Last-Modified**
- Kein **Last-Modified** gesetzt
 - i.d.R. kein Caching möglich

14.07.2004

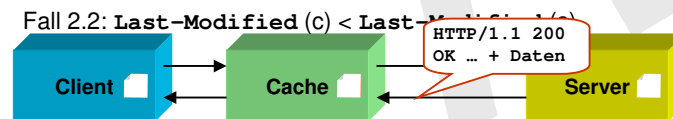
Web-Caching

Folie 14

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Gültigkeitsprüfung (Grundprinzip)



14.07.2004

Web-Caching

Folie 15

Grundlegende Funktionsweise

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Sicherheitsrestriktionen

- Verschlüsselte Inhalte (SSL)
 - werden nicht in Caches abgelegt
- Authentifizierte Inhalte
 - werden i.d.R. nicht gespeichert
 - Sicheres Caching dennoch möglich
 - » **Cache-Control: public, no-cache**
 - » Ablage im Cache
 - » Weiterleitung des Client-Request zum Server
 - » Auslieferung nur bei erfolgreicher Authentifizierung
- Problem: Sicheres Caching ist langsam
 - Unkritische Inhalte auslagern

14.07.2004

Web-Caching

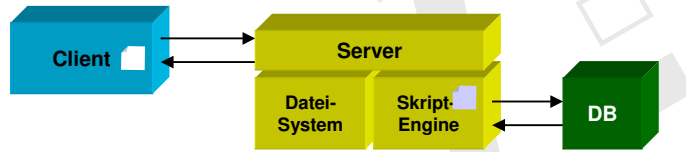
Folie 16

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Dynamische Webseiten

- HTML-Code wird bei Aufruf durch Software generiert
 - Nicht in Datei gespeichert
- Nötig für interaktive und personalisierte Angebote
 - Warenkorb, Kunden-Login, etc.
- Auch bei kleineren Projekten zunehmende Verbreitung



14.07.2004

Web-Caching

Folie 17

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Probleme und Herausforderungen

- Ausführung eines Programms für jeden Seitenaufruf
 - Zusatzbelastung: Rechenaufwand
- Erhöhte Anfragelast an Server, da
 - viele Inhalte verschlüsselt
 - » kein Caching möglich
 - personalisierte Seiten nur für einen Nutzer von Bedeutung
 - » Caching in Proxys ineffizient
 - kurzlebige Teilinhalte (z.B. Aktienkurse)
 - » Erneuerung in kurzen Zeitabständen
- Reichen bisherige Absätze aus?

14.07.2004

Web-Caching

Folie 18

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Herkömmliches Proxy-Caching

- Gut geeignet für
 - statische Webseiten
 - langlebige, öffentliche Inhalte
 - » z.B. Produktkataloge
- Schlecht geeignet für
 - kurzlebige, öffentliche Inhalte
 - » z.B. Auktionsstände, Aktienkurse
- Nicht geeignet für
 - persönliche Inhalte
 - » Benutzer-Kennungen wie Cookies werden im Cache-Schlüssel nicht berücksichtigt

Neue Ansätze im dynamischen Umfeld notwendig!

14.07.2004

Web-Caching

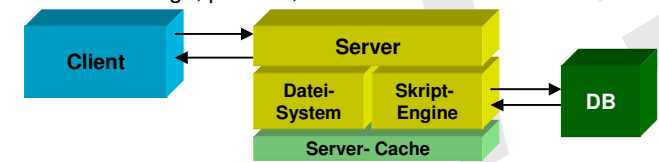
Folie 19

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Serverseitiges Caching

- Idee
 - Minimierung der Rechenlast durch Caching von (Zwischen-)Ergebnissen der Seitengenerierung
- 4 Granularitätsstufen
 - DB Query, Execution Result, Fragment, Page
- Aktualisierung
 - on-change, periodic, on-demand



14.07.2004

Web-Caching

Folie 20

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Edge Side Includes – ESI (Idee)

- Serverseitiges Caching
 - Reduzierung der Rechenlast
 - Caching im Sitzungskontext
 - Keine Reduzierung der Transferlast
- Idee: Serverseitige Ansätze auch in externen Caches
 - Zerlegung von Seiten in Fragmente
 - Getriggerte Ersetzungsstrategie
 - Unterstützung von personalisierten Inhalten

14.07.2004

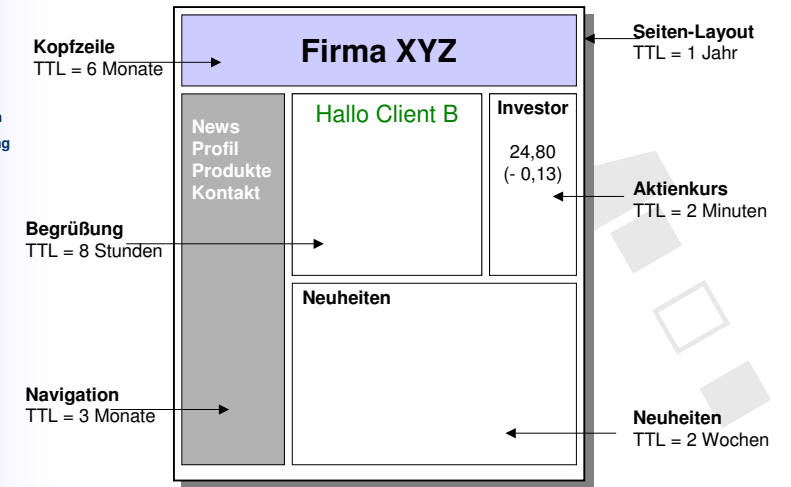
Web-Caching

Folie 21

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Edge Side Includes – ESI (Prinzip)



14.07.2004

Web-Caching

Folie 22

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Edge Side Includes – ESI (Sprachkonstrukte)

- Fragmentierung des HTML-Codes mittels XML
- Tags zur Fehlerbehandlung, Fallunterscheidung und Einbeziehung externer Variablen (z.B. Cookies) vorhanden
- Code-Beispiel

```
<html><head><title>ESI Beispiel</title></head><body>
<esi:try>
  <esi:attempt>
    <esi:include src=„www.beispiel.de/kopf.htm“
      onerror=„continue“/>
  </esi:attempt>
  <esi:except>
    Alternativer HTML-Code
  </esi:except>
</esi:try>
Statischer HTML-Code oder weitere ESI-Anweisungen
</body></html>
```

14.07.2004

Web-Caching

Folie 23

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Edge Side Includes – ESI (Invalidierung)

- Server kann explizit Inhalte in Caches entfernen.
- Austausch von XML-Nachrichten über HTTP
 - Invalidation-Request von Server an Caches
 - Invalidation-Response von Caches an Server

```
POST /x-invalidate HTTP/1.0      HTTP/1.1 200 OK
[...]                          [...]
<INVALIDATION>                 <INVALIDATIONRESULT>
  <OBJECT>                       <OBJECTRESULT>
    <BASICSELECTOR               <BASICSELECTOR
      URI=„/cache.htm“/>         URI=„/cache.htm“/>
    <ACTION/>                    <RESULT ID=„1“
  </OBJECT>                      STATUS=„SUCCESS“
</INVALIDATION>                NUMINV=„1“/>
                                </OBJECTRESULT>
                                </INVALIDATIONRESULT>
```

14.07.2004

Web-Caching

Folie 24

Ansätze im dynamischen Umfeld

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- ➔ Ansätze im dynamischen Umfeld
- Neuere Strategien
- Zusammenfassung

Edge Side Includes – ESI (Bewertung)

- W3C-standardisiert
- JESI – ESI für JavaServerPages
 - Automatische Erstellung der Tags und HTTP-Header
 - Unterstützt partielle Ausführung von JSP
 - Automatische Invalidierungsnachrichten
- Zentralisierte Inhaltskontrolle
- Spezielle Anwendungs- und Cache-Server nötig
- Dynamische Layouts nicht unterstützt
- Mangelnde Transparenz für Entwickler

14.07.2004

Web-Caching

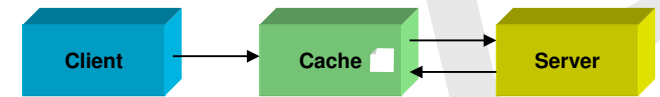
Folie 25

Neuere Strategien

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- ➔ Neuere Strategien
- Zusammenfassung

Bisher

- Cache enthält Inhalte, die in der Vergangenheit von Clients angefordert wurden.
- Ersetzung ungültiger Einträge erst bei erneutem Request
 - Nutzer stoßen Aktualisierungsprozess an.
 - Bei Clientzugriff oft kein aktueller Inhalt vorhanden
 - » Hit-Rate gering (< 50%)



14.07.2004

Web-Caching

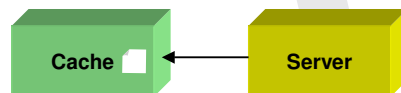
Folie 26

Neuere Strategien

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- ➔ Neuere Strategien
- Zusammenfassung

Push-Caching

- Cache enthält Inhalte, die von Servern repliziert wurden
- Ersetzung ungültiger Inhalte durch Server (vgl. ESI)
 - Zentral koordinierte Invalidierung und Replikation
 - Bei Clientzugriff stets aktueller Inhalt vorhanden
- Meist in kommerziellen Content Delivery Networks



14.07.2004

Web-Caching

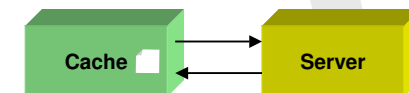
Folie 27

Neuere Strategien

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- ➔ Neuere Strategien
- Zusammenfassung

Aktives Web-Caching

- Cache enthält Inhalte, die in der Vergangenheit von Clients angefordert wurden.
- Ersetzung ungültiger Einträge selbständig in Leerlaufzeiten
 - Cache berechnet ideale Einlagerungszeitpunkte
 - Bei Clientzugriff stets aktueller Inhalt vorhanden
 - » Hit-Rate von ~70% möglich



14.07.2004

Web-Caching

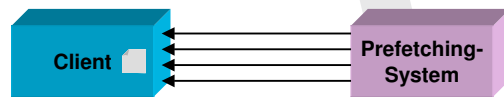
Folie 28

Neuere Strategien

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- ➔ Neuere Strategien
- Zusammenfassung

Prefetching

- Automatische Übertragung von erfahrungsgemäß nachfolgend angeforderten Inhalten
- Prefetching-System lernt Interessen der Nutzer und stellt Querverbindungen zwischen Inhalten her.
- Kein Caching-Verfahren, Kombination aber sinnvoll



14.07.2004

Web-Caching

Folie 29

Zusammenfassung

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- ➔ Zusammenfassung

Mehr Web-Caching ...

- Hier: Perspektive für Entwickler und Content Provider
 - Grundlegendes Verständnis
 - Entwicklungsansätze
- Web-Caching umfasst aber noch mehr
 - Interne Implementierung
 - Cache-Cluster und -kooperation
 - Multimedia-Caching
 - ...
- Caching von Webservices

14.07.2004

Web-Caching

Folie 30

Zusammenfassung

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- ➔ Zusammenfassung

Web-Caching kurz gefasst

- Nutzer werden aus nahen Zwischenspeichern bedient
- Ausgereifte Verfahren für statische Webseiten
- Dynamische Webseiten stellen neue Herausforderungen
- ESI und serverseitiges Caching
- Neuere Einlagerungsstrategien

14.07.2004

Web-Caching

Folie 31

Zusammenfassung

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- ➔ Zusammenfassung

Ausblick

- Web-Caching muss weiter optimiert werden
 - Datenmenge, Erwartungen, Skalierbarkeit, sichere Inhalte
- Neuere Strategien vielversprechend
 - Standardisierung?
- Caching für Webservices gewinnt an Bedeutung
 - Erweiterung von WSDL und SOAP nötig

14.07.2004

Web-Caching

Folie 32

- Situation im Web
- Web-Caching
- Grundlegende Funktionsweise
- Ansätze im dynamischen Umfeld
- Neuere Strategien
- ➔ Zusammenfassung



Fragen ?

Sebastian Adam

s_adam@informatik.uni-kl.de