
Wintersemester 2018/2019

Peer-to-Peer und Cloud Computing

Lösungsvorschläge zu Aufgabenblatt 1

1 Organic Computing und P2P- und Cloud-Computing (5 Punkte)

Weshalb beschäftigt sich das Forschungsfeld Organic Computing mit P2P- bzw. Cloud-Computing-Systemen?

- Was ist überhaupt Organic Computing? Definieren Sie den Begriff *in eigenen Worten!* (1 Punkt)

Lösung

- Motivation: immer mehr autonome Systeme mit Sensoren und Aktuatoren, selbstorganisiert, etc.
 - ⇒ steigende Komplexität
 - ⇒ konkrete mögliche Systemzustände im Vorfeld unbekannt
- OC ist ein Paradigma für den Entwurf zur Handhabung komplexer (Software-)Systeme, dabei werden oft naturinspirierte Verfahren benutzt.

- Warum sind P2P-Systeme aus Organic-Computing-Perspektive interessant? (2 Punkte)

Lösung

Wichtig ist *aus OC-Perspektive*, also: Warum würde ein OCler sich P2P-Systeme anschauen?

- Selbst-X-Eigenschaften eines P2P-Systems
 - * (Selbst-)Anpassung des P2P-Systems an die verfügbaren Ressourcen
 - * P2P-System als selbstorganisiertes System
- P2P-Systeme zeigen komplexes Verhalten
 - ⇒ könnten von OC-System optimiert werden

- P2P-Systeme sind manchmal Lösungen für Probleme des OC
⇒ manche OC-Systeme sind P2P-Systeme
- Welche Möglichkeiten könnte Organic Computing für den Cloud-Computing-Bereich eröffnen? (2 Punkte)

Lösung

- die hinter Cloud-Computing stehende Komplexität wächst stetig
OC zum Beherrschen dieser Komplexität (aus Betreibersicht)
⇒ „autonomere“ Cloud
- Selbstorganisation von Cloud-Centern
z. B. bei Ausfall von Hardware, für intelligenteres Load-Balancing, ...

2 Verständnisfragen zu P2P-Systemen (16 Punkte)

2.1 Bootstrapping-Problem (5 Punkte)

Wie finden frisch gestartete Peers einen Einstieg in ein P2P-Netzwerk? Beschreiben Sie das Problem dabei! Recherchieren und beschreiben Sie *drei* unterschiedliche Verfahren zu seiner Lösung!

Stellen Sie insbesondere mindestens ein Verfahren vor, das auch in *reinen (pure)* P2P-Netzwerken funktioniert.

Lösung

Bootstrapping-Problem: Frisch gestartete Peers benötigen die Adresse eines bereits im Netzwerk befindlichen Peers, damit dieser ihnen andere Peers vorstellen kann und der neue Peer sich eingliedern kann. Verschiedene Bootstrapping-Verfahren ermöglichen frisch gestarteten Peers auf unterschiedliche Weise an diese Information zu kommen.

Auswahl an Verfahren:

- zentraler Server, der eine Liste mit Adressen von Peers verwaltet (*Public-Hostlist-Server*)
- in die P2P-Software *hartkodierte Liste* von Adressen von Peers
- Durchtesten einer großen Anzahl von möglichen Peer-Adressen, z. B. einfach IP-Adressen auf dem Anwendungsport (*Bruteforcing*)
- *Cachen* der letzten Verbindungen mit dem Netzwerk, dann bei erneutem Verbinden zuerst Durchprobieren dieser Peers

- mittels dynamischem DNS: ein Peer biegt einen bestimmten (in die Software hartkodierte) DNS-Eintrag auf sich um – wenn dieser offline geht, wird das von anderen Peers bemerkt, die seinen Platz einnehmen

2.2 Strukturierte vs. unstrukturierte P2P-Systeme (7 Punkte)

- Stellen Sie die Vor- und Nachteile der beiden Arten von P2P-Systemen gegenüber. (2 Punkte)

Lösung

- Vorteile von strukturierten gegenüber unstrukturierten
 - * Suche gerichtet
⇒ effizienter (geringerer Nachrichtenaufwand)
 - * Exact-Match-Querys möglich
- Vorteile von unstrukturierten gegenüber strukturierten
 - * deutlich geringerer Managementaufwand bei Hinzufügen/Entfernen von Peers
⇒ deutlich effizienter bei stärkerer Fluktuation
- Welche der beiden Arten würden Sie für die folgenden Anwendungsszenarien nutzen? Warum? (5 Punkte)
 - File-/Content-Sharing
 - Information-Sharing
 - Bandwidth-Sharing
 - Storage-Sharing
 - Processing-Power-Sharing

Lösung

File/content sharing unstrukturiert

- meist stark schwankende Population
- i. d. R. reicht es dem Teilnehmer, *irgendein* Replikat der gesuchten Datei zu finden

Information sharing strukturiert

- zuverlässigere Suchmöglichkeiten, v. a. erwiesene Nicht-Existenz von Information
- weniger stark schwankende Population (Universitäten, Institute, ...)

Bandwidth sharing unstrukturiert

- stark schwankende Population (Verlassen des Netzwerks, wenn Download abgeschlossen)
- geringere Netzwerkauslastung (keine Neustrukturierungen notwendig)

Storage sharing – bei stark schwankender Population: unstrukturiert

- üblicher, da verteiltes Dateisystem oft dauerhaft gemounted: wenig schwankende Population ⇒ strukturiert
 - * Strategien für Replikation können besser umgesetzt werden
 - * es darf kein Datum verloren gehen, da meistens Fragmentation eingesetzt wird ⇒ sowieso schon Overhead für Leaver-Verwaltung

Processing power sharing unstrukturiert

- in der Regel stark schwankende Population
- Arbeitspakete müssen i. d. R. nicht auf ganz bestimmten Teilnehmern gerechnet werden

2.3 Zentralisierte P2P-Systeme und Client/Server-Architektur (4 Punkte)

- Welche gemeinsamen Stärken und Schwächen haben zentralisierte P2P-Systeme und die klassische Client/Server-Architektur? (2 Punkte)

Lösung

- gemeinsame Stärken
 - * kein kompliziertes Bootstrapping nötig (bei Client/Server sowieso nicht, bei zentralisiertem P2P-System ist es trivial)
 - * Zugangskontrolle leichter möglich
 - * schnelles Auffinden von angeforderten Ressourcen
- gemeinsame Schwächen
 - * skalieren nur bis zu bestimmter Menge von Ressourcen/Menge an Anfragen/... gut
 - * Single Point of Failure (zentrale Einheit/Server)
 - * Privatsphäre/Anonymität schwierig zu garantieren (zentralisierte Kommunikationswege)

* Kosten für Aufrechterhaltung des zentralen Elements

- Worin unterscheiden sich die beiden Ansätze? Welcher ist der wohl vordergründigste Unterschied? (2 Punkte)

Lösung

Ressourcen/Dienste bei P2P von *Teilnehmern* angeboten (zentrale Einheit nur für Koordination), bei C/S nur vom *Server*. Last wird bei P2P auf die Peers verteilt, bei C/S trägt sie der Server allein.

Viel Erfolg bei der Bearbeitung!