

# **Eine ökonomische Bewertung der Dienstauswahlverfahren in serviceorientierten Overlaynetzen**

**Michael Reinicke**

Universität Bayreuth

reinicke@uni-bayreuth.de

*Zusammenfassung: Einfache IT-Dienstleistungen werden im Konzept des On-Demand bzw. Grid Computing an externe Anbieter ausgelagert. Dieser bedarfsabhängige Bezug von Rechenleistung wird als Chance zur Beseitigung von Ineffizienzen und zur Kostenreduktion gesehen. Der vorliegende Beitrag beschäftigt sich insbesondere mit Verfahren zur Dienstauswahl, über die in den vorliegenden Konzepten solcher serviceorientierter Architekturen keine Aussagen bezüglich Anwendbarkeit und Performanz gefällt werden. Er vergleicht die existierenden Ansätze mit einem koordinatorfreien – auf ökonomischen Prinzipien beruhenden – Ansatz anhand ökonomischer Metriken. Mittels einer Simulation werden unterschiedliche Verfahren in verschiedenen Netzwerkszenarien untersucht und die bessere Anpassungsfähigkeit der koordinatorfreien Variante an Dynamik und Knotendichte des Netzwerks gezeigt.*

*Schlüsselworte: On-Demand Computing, Netzwerksimulation, Dienstauswahl, serviceorientierte Architektur, Grid Computing, Computational Economics*

## **1 On-demand Computing in der serviceorientierten Architektur (SOA)**

Die Unternehmen Sun Microsystems, IBM und HP, arbeiten seit einigen Jahren in ihren Softwarearchitekturen daran, Kunden IT-Dienstleistungen bei Bedarf („on-demand“) anzubieten. Das Geschäftsmodell des on-demand Computings (ODC), erlaubt es Unternehmen, Computerressourcen nur bei Bedarf zu nutzen und zu bezahlen. On-demand Computing ist eine Kombination zweier Vorteile. Erstens ist es ein Dienstzugriffsmodell, bei dem Unternehmen über die IT-Infrastruktur eines externen Dienstleisters Zugang zu zusätzlichen Ressourcen erhalten, um Bedarfsspitzen abzufangen. Zweitens wird mit Hilfe eines Zählwerterfassungsverfahrens ein „Pay as you go“-Bezahlmodell ermöglicht, das die Abrechnung nur derjenigen IT-Dienste und Ressourcen, die das Unternehmen auch tatsächlich

nutzt, erlaubt. Somit wird es Unternehmen einfacher gemacht, bedarfsgerecht zu kalkulieren [Sear03; Bloo03; Matt03].

Grid Computing ist eine Form der Virtualisierung, die dem On-demand Ansatz folgt. Grid Computing beschreibt die Bündelung einer großen Zahl von Rechner-systemen, deren aggregierte Prozessorleistung einen virtuellen Computer ergibt, der besonders rechenintensive Aufgaben durchführen kann. Dabei spielt es keine Rolle, an welchem physikalischen Ort diese Leistung erbracht wird. Solche Systeme werden derzeit genutzt zur Berechnung von z.B. Klimaveränderungen, zur Krebstherapie oder der Entschlüsselung des Genoms [Rech04]. Außerdem ermöglicht das ODC Unternehmen, brachliegende Kapazität zu nutzen. So lassen einige Unternehmen die Desktop-PCs ihrer Mitarbeiter nachts virtuell zusammenschalten, um aufwendige Berechnungen auszuführen. Grid Computing steht für einen Spezialfall des On-demand Computing.

### **1.1 On-demand Computing: Kostensenkung oder Risikoerhöhung?**

Aus betriebswirtschaftlicher Sicht sprechen in der Informations- und Kommunikationstechnologie (IuK-Technologie) besonders Kostenargumente für das selektive Fremdbeziehen von eng begrenzten, klar definierten Aufgaben. Für in Eigenregie durchgeführte Tätigkeiten muss das Unternehmen die notwendige Ausrüstung und das erforderliche Know-how besitzen bzw. erwerben und weiterentwickeln. Dies kann nur mit organisatorischen und humankapitalintensiven Anstrengungen erreicht werden, was aber besonders kleinen und mittelständischen Unternehmen (KMU), deren Kernkompetenz meist nicht im IT-Bereich liegt, schwer fällt. Auch mit den rasanten, technologischen Neuerungen Schritt zu halten, ist mit Aufwand versehen, der bequem externalisiert werden könnte. In besonderem Licht stehen besonders die periodisch anfallenden Kosten, die bei eigener Bereitstellung unabhängig von der Nutzung der Dienste anfallen. Hierunter fallen unter anderem Raum-, Personal- und Instandhaltungskosten, die einen immensen Fixkostenblock darstellen.

Bei Fremdvergabe von Aufträgen entsteht jedoch eine Unsicherheit über den entfernten Ablauf des Prozesses, bedingt durch einen inhärenten Verlust der Kontrolle, welcher dem Effekt der möglichen Kosteneinsparung gegenüber steht. Abgeleitet aus der Principal-Agent-Problematik kann ein nicht steuerbares bzw. nicht anreizkompatibles Verhalten des Fremdanbieters zu mangelnden Sicherheitsniveaus und Datenschutzproblemen führen und damit erwirtschaftete Marktpositionen ernstlich gefährden. Nicht selten wird bei den Anbietern an diesen, im Vorhinein nicht kontrollierbaren, Leistungseigenschaften – z.B. der dauerhaften Verfügbarkeit – gespart, um die Angebotspreise zu senken und damit attraktivere Verhandlungspositionen einzunehmen. Es schließen sich technische und ökonomische Problemfelder an, die dem Effekt der Kostensenkung massiv entgegen wirken können, so dass schließlich ein „Insourcing“ sinnvoller ist. Hier sind geeignete

te Mechanismen gefragt, die das Risiko einer Auslagerung kalkulierbar und damit den gesamten Prozess effizient machen.

Generell sind on-demand Architekturen also wirtschaftlichen Problemen ausgesetzt. Diese Probleme können jedoch meist auf das Nichtvorhandensein von marktlichen Mechanismen zurückgeführt werden [Eggs01, S. 249]. Zur Analyse der noch einzuführenden, wirtschaftlichen Eigenschaften, wird davon ausgegangen, dass vollständig funktionierende Märkte nur dann auf Dauer Erfolg versprechend sind, wenn sich alle betriebswirtschaftlichen Transaktionsphasen auf den zu untersuchenden Markt abbilden lassen. Als Transaktion als solche wird die Vereinbarung und Regelung über den Tausch aufgefasst. Aufgrund asymmetrischer Informationsverteilung fallen dabei jedoch Probleme (und Kosten) in den einzelnen Phasen an [Pic<sup>+</sup>96, S. 41], die auf klassischen Märkten durch (einen Markt voraussetzende) Institutionen gelöst werden können. Diese Institutionen sind im ODC jedoch nicht vorhanden und daher bedarf es zu einer Outsourcing Entscheidung einer genaueren Analyse der Kosten.

## **1.2 Unterstützung der Transaktionsphasen durch Standardisierung**

Bei on-demand Verträgen ist die Betrachtung der Transaktionskosten von essentieller Bedeutung. Die externen Transaktionskosten repräsentieren die Kosten der Marktnutzung. Die Höhe dieser Kosten hängt in besonderem Maße davon ab, wie spezifisch die zu erbringende Leistung ist und wie oft diese vom gleichen Anbieter erbracht wird. Denn über einen langen Nutzungszeitraum amortisieren sich dann die Kosten der Marktnutzung. Eine geringe Spezifität der auszulagernden Aktivität erlaubt dabei die preisgünstige Externalisierung. Bei hoher Spezifität dagegen ist es schwierig, die Leistung zu beschreiben und zu bewerten. Sind die Leistungen hoher Spezifität jedoch bereits im Vorfeld weitestgehend standardisiert, kann dem Effekt geeignet entgegengewirkt werden. Die serviceorientierte Architektur (SOA) versucht hier, diese Spezifität durch eine Standardisierung zu reduzieren und ist im Wesentlichen eine Sammlung von Diensten innerhalb eines Netzwerks. Diese Dienste haben die Fähigkeit, miteinander zu kommunizieren und Daten auszutauschen. Die Nachrichteninhalte können entweder einfache Daten zur direkten Nutzung durch den Anwender oder Verwaltungsdaten zur Koordination zweier oder mehrerer Dienste sein. Serviceorientierte Architekturen sind jedoch keine neue Erfindung, die erste SOA wurde etwa mit der Nutzung von DCOM oder Object Request Brokern (ORBs) auf der CORBA Spezifikation realisiert [SOA04].

Seit den letzten Jahren tauchen im Zusammenhang mit der SOA immer wieder die Begriffe *Web Services* und *SOAP* auf. Diese Begriffe dienen der Spezifizierung der Dienste und Protokolle in der SOA und zielen damit auf eine Verringerung der Spezifität und damit auch der asymmetrischen Informationsverteilung in der ersten Transaktionsphase, der Anbahnungsphase, ab: Für die Dienstsuche beschreibt

die WSDL (Web Service Description Language) [W3C01] in einem XML-Format den Dienst mit bestimmten Kriterien. Zusammen mit UDDI (Universal Description, Discovery and Integration), den gelben Seiten der Web Services, kann ein Informationsdefizit aufgebrochen werden und ein Dienst aufgefunden werden [Webo04].

SOAP [W3C04] wiederum spezifiziert die Nachrichten, die zwischen den Applikationen auf Seiten der Nachfrager und Anbieter ausgetauscht werden. So können die Hürden der Kommunikation verschiedener Anwendungen, die womöglich auf verschiedenen Betriebssystemen laufen, erklommen werden. Hier werden demnach die Phasen Verhandlung und Abwicklung unterstützt.

Seit Beginn des Jahres 2004 verfolgen die Firmen IBM, HP und Sun zusammen mit der Globus Foundation das Ziel, Web Services und Grid Computing durch eine Erweiterung der Web Services Spezifikationen um Web Service Notifications und ein Web Service-Resource Framework (WSRF) zusammenzufassen [Sabb04; Kril04]. WS-Notification ist eine Spezifikation zum so genannten Einleiten eines Web Service Ereignis; WS-Resource Lifetime erlaubt Betreibern von Web Services das Setzen einer Zeitspanne in der die Definition einer Ressource gültig ist; WS-Resource Properties definieren dagegen, wie solche Daten abgefragt und geändert werden können. WS-Notification und das WSRF erlauben eine standardisierte Infrastruktur für Geschäftsanwendungen und Grid Ressourcen. Diese Spezifikationen schafft ein „publish and subscribe messaging model“ und damit die Fähigkeit, Ressourcen zustandsabhängig zu modellieren [Kril04].

Die SOA ist also ein Ansatz, über standardisierte Nachrichtenprotokolle und -spezifikationen, Transaktionskosten zu senken und leistet damit einen Beitrag, Unsicherheit in den ersten Transaktionsphasen zu reduzieren. Dennoch bleiben Fragen unbeantwortet. Etwa, wie (inhaltlich gleiche) Dienste aus einer Liste aufgefunderer Dienste ausgewählt werden sollen. Darüber werden in der SOA keine Aussagen gefällt. Mit diesem Problem wird sich der Beitrag im Folgenden beschäftigen. Für die weitere Argumentation wird insbesondere die These aufgestellt, dass die Wahl des Dienstes einen Einfluss auf das Funktionieren bzw. die Leistungsfähigkeit der SOA hat. Bevor jedoch auf die Dienstausswahl als solche eingegangen wird, soll zunächst erläutert werden, wie die Leistungsfähigkeit einer SOA beschrieben und gemessen werden kann.

### 1.3 Technische und ökonomische Bewertungsmetriken

Zur qualitativen Bewertung der Leistungsfähigkeit einer serviceorientierten Architektur tragen mehrere technische und ökonomische Merkmale bei. Die ökonomischen Größen sind meist aus den technischen Merkmalen ableitbar.

Zu den technischen Kriterien gehören z.B.

- die Allokationsrate, die die Anzahl der Kontrakte im Verhältnis zu den gestarteten Dienstanfragen misst;

- die Dauer zwischen dem Starten der Dienstanfrage bis zur erfolgreichen Zuweisung, die durch die Response Time (REST) beschrieben wird und
- die Anzahl der notwendigen Kontrollnachrichten.

Diese Metriken sind, wie bereits angedeutet, im Rahmen der Betrachtung in ökonomische Metriken überführt worden; eine hohe Allokationsrate führt etwa zu einer höheren Anzahl von Kontrakten und damit zu einer Verbesserung der Gesamtwohlfahrt, da die Bedürfnisse der Akteure häufiger befriedigt werden können. Die Allokationseffizienz und die Wartezeit bis auf den Dienst zugegriffen werden kann, kann mit den Kosten des Ausfalles des Netzes bzw. den Risikokosten für ein nicht zeitgerechtes Übertragen des Dienstes assoziiert werden. Eine hohe Wartezeit führt dazu, dass das Risiko in Kauf genommen werden muss, den Dienst nicht am Wunschverfügbarkeitstermin zu erhalten. Niedrigere Werte der Response Time (REST) bzw. der Kommunikationskosten deuten also einen Nutzenzuwachs an.

Überdies erlaubt das Heranziehen dieser Kriterien, die Fähigkeit auf exogene Schocks (Ausfall von Links, Knoten und Diensten; Nachfragespitzen) zu reagieren zu erfassen: Ein System erholt sich schneller und lässt den Schluss zu, höhere Nutzengewinne zu ermöglichen, als ein System, das leicht gestört werden kann und damit aus den Grenzen zu heben ist. Können ausreichende Werte der ökonomischen Messgrößen garantiert werden, so lässt das auch den Schluss zu, dass das System in der Leistungsfähigkeit verlässlicher ist und damit Kosten für eine Versicherung des Ausfalls, sowohl auf Seiten der Nachfrager als auch auf Seiten der Anbieter reduziert werden können.

## 2 Dienstausswahlverfahren zur effektiven Selektion

Große und komplexe Computernetze, die verteilte Dienste verschiedener Art bei Bedarf zur Verfügung stellen, werden seit Ende des letzten Jahrhunderts in der serviceorientierten Architektur zusammengefasst. Im Rahmen dieses neuen Ansatzes soll eine große Anzahl von Rechnern lose gekoppelt werden, um eine verteilte Informationssuche, eine parallele Verarbeitung von Aufgaben oder eine persistente Datenspeicherung zu erlauben. Content Distribution Netzwerke [KoDa99; RaAg99; Akam04] oder Peer-to-Peer Netze [Clar99; FoKe97] sind solche Softwarearchitekturen, die Dienste bereitstellen und über Computernetzwerke verfügbar machen. Die Ressourcen sind über einfache Kommunikationsinfrastrukturen – wie dem Internet – verbunden. Diese Netze werden z.B. zum Bereitstellen von Multicast-Diensten für Gruppen [RaAg99], zum Speichern extrem großer Datensätze [Clar99] oder zum Ausführen von Anwendungen, die Rechenleistung in der Größenordnung von Gigaflops benötigen [FoKe97] eingesetzt. Populäre Anwendungen, die nun auch verstärkt im Interesse der Industrie stehen und das Potential für grundlegende Innovationen bergen, sind etwa das Grid Computing für eine

verteilte, parallele Verarbeitung, File Sharing bzw. Persistent Storage Systeme oder Instant Messaging.

Eine wichtige Fragestellung für den Erfolg dieser Anwendungen ist, wie nutzbare Dienste und Informationen aus einer Fülle von Daten im Netz gefunden und extrahiert werden können [CATN03b; Mil<sup>+</sup>02; Snell03]. Effektive Dienstfindungs- und Dienstauswahlmethoden sind daher notwendigerweise erforderlich. Bei gegebener Komplexität und Dynamik einer Grid Infrastruktur, ist die Skalierfähigkeit und die Verwaltung einer hohen Anzahl von heterogenen Ressourcen die überragende Herausforderung für die Nachhaltigkeit. Insbesondere für das Management dieser Netze müssen automatisierte, vorzugsweise selbstorganisierende Lösungskonzepte zu den folgenden Auswahlproblemen bereitstehen:

- Das initiale Verteilen bzw. Neuverteilen von Diensten und Ressourcen im Netzwerk: Aufgrund der Annahme, dass eine spezifische Nachfrage für einen Dienst bzw. Web Service existiert, stellt sich für den Anbieter die Frage, von wo der Dienst physikalisch bereitgestellt werden sollte. Der Ort des Angebots beeinflusst den Ertrag des Dienstes für den Eigentümer maßgeblich und zudem die subjektive Empfindung des Nachfragers (z.B. durch Wartezeiten). Auch die Performanz des Netzes hängt von der Platzierung der Dienste ab. Eine Veränderung in der Struktur der Nachfragen könnte beispielsweise zu einem notwendigen Neuverteilen der Ressourcen führen. Existierende Ansätze sind etwa Xweb Multicast dampening [Ard<sup>+</sup>01], der OPTORSIM Replica Optimizer [Bel<sup>+</sup>03] oder DYNAMO [Sil<sup>+</sup>03].
- Das Auffinden von verfügbaren Diensten stellt ein Problem dar. Bei Existenz mehrerer redundanter Dienste, die jeweils alleine die Nachfrage befriedigen könnten, stellt sich die Frage, welche dieser Dienste gerade verfügbar sind. Genau diese Instanzen in einer angemessenen Zeitspanne zu finden ist eine anspruchsvolle Aufgabe (Dienstfindung). Unstrukturierte Verfahren zur Dienstfindung werden bereits in Globus, Gnutella oder Datagrid eingesetzt, garantieren aber das Auffinden nicht. Die aus diesem Grunde entwickelten strukturierten Verfahren stellen ein viel versprechendes Novum dar, da eine Dienstfindung in einer vorhersagbaren Zeit garantiert werden kann. Mit Hilfe von verteilten Datenbanken und so genannten Distributed Hash Tables (DHT) sorgen die Netzknoten gemeinsam für ein effizientes Auffinden [Rat<sup>+</sup>01; Bal<sup>+</sup>03].
- Die Auswahl eines passenden Handelspartners: Aus einer Liste von aufgefundenen Dienstinstanzen muss der Nachfrager bzw. der Resource Broker einen passenden Transaktionspartner wählen, der beide Teilnehmer gleichermaßen befriedigt. Dieser Prozess liegt im Fokus des vorliegenden Beitrages. Die gängigen Grid Systeme bedienen sich meist zentraler Resource Broker zur Auswahl des Dienstes. Verfügbare Ansätze – wie etwa Nimrod/G oder OptorGrid [Bel<sup>+</sup>03] – sind meist mit der Dienstfindung verknüpft.

Im Folgenden werden die verschiedenen Verfahren der Dienstausswahl anhand der Zentralität des Auswahlmechanismus und der Reihungsmethoden zur Allokation kategorisiert.

## 2.1 Zentrale Dienstfindung und -auswahl

Eine typische Form der Realisierung der Dienstausswahl ist die der Verwendung eines zentralen Resource Brokers. Dieser sammelt die ihm zugetragenen Informationen über Nachfrage und Angebot vorhandener Dienste. Die Liste der möglichen Allokationspartner wird zentral angelegt und nach definierten Bewertungskriterien gereiht. Die Akteure (Nachfrager und Anbieter) aktualisieren diese Information in bestimmten zeitlichen Abständen mittels Nachrichten, die ihren derzeitigen Anfragestatus und ihre Verfügbarkeit beinhalten, um den Broker auf einem möglichst aktuellen Informationsniveau zu halten. Denn nur mit aktuellen Daten kann der Broker zw. Koordinator die Zuordnung fehlerfrei herstellen. Den einzelnen Instanzen wird nach der durch den Broker durchgeführten Zuweisung – die nach einer definierten Anzahl von erhaltenen Daten erfolgt bzw. nach zeitlichen festgelegten Intervallen – nur die Allokationsinformation mitgeteilt. Der Nachfrager tritt dann in den direkten Kontakt mit dem Anbieter, um die temporäre Kooperation zu initiieren. Bild 1a stellt diesen Verlauf grafisch dar.

Charakteristisches Beispiel ist Condor-G [Bel<sup>+</sup>03] [Fre<sup>+</sup>02]. Condor-G nutzt einen Resource Broker. Jede Nachricht über eine Nachfrage oder ein Angebot wird mit einem so genannten ClassAd versehen. Verfügbare Ressourcen werden gemäß Nutzerpräferenzen, Zuweisungskosten oder Anfangszeitpunkten geordnet. Die ClassAds bestehen unter anderem aus einem speziellen „Requirement and Rank“-Etikett. Das Rank-Kriterium wird genutzt zur Reihung der Angebote. Die verfügbaren Ressourcen und Nutzeranfragen werden zudem in einer bestimmten statischen Frequenz nach ihrem aktuellen Status abgefragt. Nach Ablauf einer bestimmten Zeitspanne entscheidet der Broker, wer welchen Dienst in Anspruch nehmen darf. Zunächst werden hierzu verfügbare Ressourcen nach Nutzerpräferenzen, Allokationskosten und erwarteten Start- und Bearbeitungszeiten eingestuft. Nach einem Anpassen der Anforderungen werden die Rangfolgen für jedes mögliche Paar gebildet, aufgrund derer über eine Heuristik die Zuweisung erfolgt [Lit<sup>+</sup>88; Liu<sup>+</sup>02]. Der Broker sendet die Allokationsdaten sodann an die Peers. Diese wiederum wickeln die Transaktion dann ohne Zutun des Brokers ab. Weitere, ähnliche Verfahren finden sich unter [Cha<sup>+</sup>98; FoKe97; RaAg99].

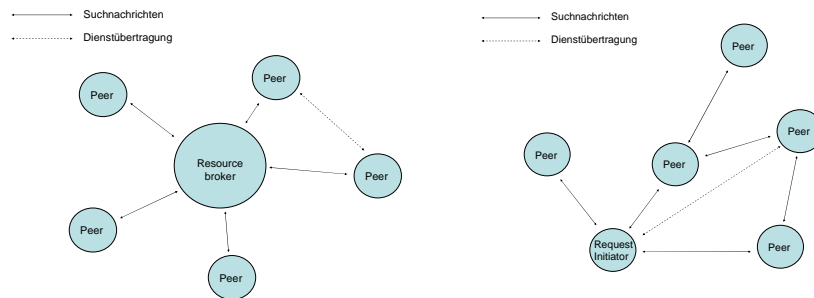


Bild 1a/b. Ablauf zentraler und dezentraler Dienstfindung und -selektion

Wird der zentrale Broker in Netzen eingesetzt, die nicht-statischer Natur sind, deren Akteure also nicht konstant erreichbar und verfügbar sind, kommt es zwangsläufig zu Allokationen zwischen Handelspartnern, die nicht mehr verfügbar sind, bzw. zu theoretisch besseren Allokationsmöglichkeiten durch das Hinzukommen von neuen Akteuren, sei es durch Ausfälle bzw. Aktivieren von neuen Knoten und Links oder bewusste Nutzereingriffe. Diese suboptimalen Zuweisungen nennt man Fehlallokationen, da sie unter Ausschluss globaler Informationen den Optimalitätsanspruch nicht gewährleisten können. Um Fehlallokationen effektiv zu vermeiden, ist es notwendig, dass die Ressourceninhaber ihre Verfügbarkeit ständig überprüfen und die beim Resource Broker liegenden Informationen aktualisieren. Die Endnutzer speichern gesendete Anfragen in Warteschlangen und aktualisieren diese Liste kontinuierlich.

Für optimale Ergebnisse ist demnach eine globale Sicht auf das System erforderlich, was jedoch problematisch ist [Li<sup>+</sup>02]. Im Folgenden sind die Erfordernisse für ein idealisiertes Zuweisen aufgeführt:

- Erstes Erfordernis der zentralen Kontrolle ist, dass der Zustand des Netzes sich zwischen dem Beginn und dem Ende des Berechnungs- bzw. Allokationsprozesses nicht ändern darf. Netzwerke sind jedoch sehr dynamisch und ändern schnell ihren Zustand; Angebot und Nachfrage verschieben sich sehr häufig und Ausfälle [Best95] treten bei gleichzeitigem Hinzukommen neuer Dienste [Ami<sup>+</sup>98] regelmäßig auf. Dynamische Netze benötigen hingegen eine fortlaufende Aktualisierung des Koordinationsmechanismus, die die ständigen Umweltveränderungen reflektiert.
- Eine zweite Eigenschaft ist, dass der Koordinator über globales Wissen in Bezug auf den Zustand des Netzwerkes verfügen muss. Dies wird meistens durch bestimmte Zeitintervalle zwischen dem ersten Empfang der Statusnachrichten der entfernten Ressourcen bzw. Nachfragen und dem Empfang der letzten Nachricht bestimmt. Bei einem großen Durchmesser des Netzes ergeben sich hohe Latenzzeiten für die Nachrichten, die bei Eintreffen evtl. nicht mehr die aktuellen Daten beinhalten.



- Drittens stellt der Koordinator als solches ein Problem dar. Da Angebots- und Nachfragenachrichten durch das Netz zum zentralen Koordinator geleitet werden müssen, wird dem System unnötiger Datenverkehr aufgebürdet. Solange die Kontrolldaten von der Größe her nur klein im Vergleich zu den Anwendungsdaten sind, ist das weniger problematisch, doch dann, wenn das zentrale Prinzip auf weitere Anwendungsgebiete angewandt wird, werden erhebliche Kapazitäten in Anspruch genommen, die zu Lasten des Systems gehen [Ard<sup>+</sup>03].

## 2.2 Auswahl der Dienste beim Konsumenten (dezentrale Dienstausswahl)

Eine Alternative zum vorgestellten, zentralen Resource Broker versprechen dezentrale Mechanismen zu sein, welche die Entscheidung auf den anfragenden Client verschieben. Typische Realisierungsformen finden sich in Peer-to-Peer Netzen, für die Gnutella [AdHu00] ein charakteristisches Beispiel ist. Gnutella nutzt das Fluten des Netzwerkes zur Dienstfindung: Nachdem ein Client eine Anfrage initiiert hat, werden Nachrichten an die Nachbarn verteilt. Auch diese fragen ihre Nachbarn an. Ein Time-to-live Parameter (TTL) begrenzt dieses Ausbreiten von Suchnachrichten und damit den Suchraum unter der optimistischen Annahme, dass der Dienst gefunden wird [Mi<sup>+</sup>02]. Über alle aufgefundenen, auf die Anfrage passenden Dienste wird der Client mittels einer Nachricht, dem so genannten QueryHit informiert. Diese Nachrichten werden außerdem in den Zwischenknoten gespeichert um zukünftige Anfragen zu beschleunigen. Bild 1b zeigt den Verlauf.

Suchverfahren, wie etwa Chord, Pastry oder CAN (Content Addressable Network), nutzen eine Distributed Hash Table (DHT) um dieses Fluten zu umgehen. Die Suche wird in diesen Verfahren über alle Knoten verteilt und ein Auffinden kann garantiert werden [Rat<sup>+</sup>01].

Diese unterschiedlichen Suchprozesse führen jeweils dazu, dass eine Liste der verfügbaren Dienste beim Client generiert wird. Der Client muss nun mit seinem eigenen, lokalen Wissen einen Dienst aus der Liste auswählen. Diese Auswahl trifft er unter Unwissenheit von Netzwerktopologie, etwaigen Flaschenhalssituationen und Knappheitssituationen. Aus diesem Grunde ist die Selektion des Dienstes von besonderer Wichtigkeit, da sie über den Erfolg der Transaktion entscheidet. Zum Zwecke der Entscheidung wird die Wahl mit Hilfe einer gereihten Liste durchgeführt.

## 2.3 Ordnungsmethoden

Die Reihung der gefundenen Dienste hat eine signifikante Auswirkung auf die Güte des Matchings. Sowohl der Resource Broker in der zentralen Variante, als auch der Nachfrager im dezentralen Szenario generieren bei der Suche eine Liste. Jeder Treffer muss anhand eines Verfahrens bewertet werden und in eine Reihe einge-

ordnet werden, die zur Weiterverwendung genutzt wird. Wie auch bei der Nutzung von Suchmaschinen im Internet haben diese Listen einen entscheidenden Einfluss darauf, welcher Link bzw. Dienst gewählt wird. Als Analogon dienen Suchmaschinen im World Wide Web, deren Leistung darin besteht, dem Nutzer auf den ersten Plätzen Inhalte zu präsentieren, die seiner zuvor eingegebenen Suche entsprechen und das Informationsdefizit beheben. Für die Reihung existieren unterschiedliche Methoden, die im Folgenden kurz erläutert werden.

### **2.3.1 Ordinale Reihung**

Die Auswahl eines Diensteanbieters aus einer Liste von Kandidaten wird in den meisten Anwendungen durch eine Reihung nach technischen Merkmalen durchgeführt, z.B. anhand des Kriteriums „Antwortzeit“. Die möglichen Dienste werden nach Antwortzeit aufsteigend sortiert und der Nutzer wählt den Dienst an erster Stelle aus. Ist dieser Dienst jedoch nicht mehr verfügbar, so bedient er sich des nächsten Dienstes in der Liste, solange bis ein Kontrakt zustande kommt. Ist kein Kontrakt zu Stande gekommen, die Liste aber abgearbeitet, so muss eine neue Anfrage gestartet werden.

Diese Reihung nach technischen Kriterien hat zum Nachteil, dass die Unterschiede zwischen den Plätzen der Liste nicht durch Beträge dargestellt werden, die den absoluten Nutzenverlust von first-best auf second-best darstellen. Es kann daher nicht gesagt werden, wie deutlich eine Verschlechterung bei Ausfall eines bevorzugten Dienstes ist. Eine Lösung verspricht daher die Verwendung ökonomischer Prinzipien, die im Folgenden dargestellt werden.

### **2.3.2 Ökonomische Reihung durch interne Preise**

Für die Nutzung ökonomischer Prinzipien müssen das Netz und die Güter jedoch wertorientiert betrachtet werden, was dazu führt, dass ein Preis für alle beziehbaren Netzressourcen existieren muss. Das erlaubt die Berechnung eines Nutzens auf Seiten des Nachfragers, der entsprechend den Angebotspreis mit einem potentiellen Nutzen bei Vertragserfüllung vergleichen kann und daraus seinen Nutzenzuwachs berechnen kann.

In Projekten, wie etwa GridBus oder Nimrod/G, wird die Ergebnisliste über die Bewertung von Nutzenzuwächsen sortiert. Damit können die Unterschiede in der Reihenfolge eindeutig in Nutzendifferenzen dargestellt werden. Der Nimrod/G Resource Broker ist ein Planungs- und Steuerprogramm, das für Dienstauffindung, -auswahl und -verteilung zuständig ist. Während erste Versionen noch ausschließlich fristenbasiert alloziiert haben, entscheiden aktuelle Versionen mit Hilfe ökonomischer Prinzipien: Die implementierte GRACE Infrastruktur innerhalb des Brokers erlaubt es, um Ressourcen zu handeln – ähnlich wie in einem Markt. Die Entscheidung basiert damit auf Kosten und Preisen, aber weiterhin auch auf Zugangsnormen und Fristen [Buyy02].

Diese ökonomischen Prinzipien werden z.B. von katallaktischen Informationssystemen unterstützt. Das Implementieren dieser Informationssysteme nutzt Methoden sowohl aus der Agententechnologie, als auch der Ökonomie, so genannte agentenbasierte „computational economics“ [Tesf97]. Autonome Softwareagenten verhandeln untereinander, um ihren Nutzen zu maximieren und passen ihre Verhandlungsstrategie unter Verwendung von maschinellen Lernverfahren stetig an (Evolutionäre Algorithmen, numerische Optimierung, z.B. Nelder/Mead’s Simplexmethode [Pre<sup>+</sup>02], hybride Methoden, z.B. das VID Modell von Brenner [Bren96]). Die Signalwirkung von Preisen führt zu einer konstanten Anpassung des Gesamtsystems und verbreitet Veränderungen in der Knappheit von Ressourcen innerhalb des Systems. Die sich ergebenden Muster sind vergleichbar mit denen, die in Marktexperimenten mit menschlichen Akteuren beobachtet werden [KaRo95; Prui81; Smit62]. Eine erfolgreiche Anwendung der Katallaxie in verteilten Allokationsverfahren verspricht den Vorteil einer flexiblen Struktur und eines inhärenten, parallelen Verarbeitens im Vergleich zu einem zentralen, auktionatorbasierten Ansatz. Ein Beispiel zur Simulation einer Wertschöpfungskette ist das Multiagentensystem AVALANCHE [Eyma00].

Von der Verwendung ökonomischer, rationaler Prinzipien in Dienstausswahlverfahren wurde bisher jedoch meist abgesehen. Open Agoric Systems waren ein früher Versuch, marktliche Prinzipien in Computer Systeme zu implementieren. Aufgrund mangelnder Notwendigkeit für reale Anwendungen und fehlender Skalierbarkeit schenkte ihnen die Wissenschaft kaum Beachtung [MiDr88]. Die Tauschbörse Mojo Nation [Cave00; Mojo03], die die Ressourcen, wie etwa Speicherplatz, nur über einen Markt anbot, hatte Akzeptanzprobleme einerseits wegen des Fehlens von erprobten Bezahlverfahren für Micropayments und andererseits wegen des Feststellens von Sabotageakten der Nutzer, die sich durch Betrachten Ihres virtuellen Kontos angespornt fühlten, das System auszustechen. Zusammenfassend lässt sich sagen, das Dienstausswahlverfahren zentralisiert oder dezentral durchgeführt werden können, während sich der Auswahlmechanismus technischer oder rationaler, ökonomischer Prinzipien zum Reihung der Ergebnisse bedienen kann. Bildet man diese Ergebnisse auf eine 2x2 Matrix, so ergibt sich Bild 2.

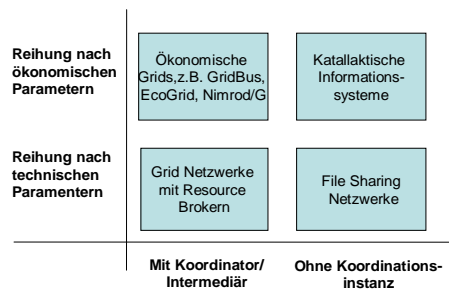


Bild 2. Dienstausswahlverfahren: Stand der Wissenschaft

## 3 Simulation und Evaluation

Die vorgestellten Dienstausswahlverfahren sollen anhand einer Simulation einander gegenübergestellt und für die Einsatzzeichnung in unterschiedlichen Szenarien technisch und ökonomisch anhand der in Kapitel 1.3 vorgestellten Metriken bewertet werden. Zur Vereinfachung wurde auf den Vergleich zwischen den verschiedenen Ordnungsmethoden verzichtet. Es wurden demnach nur die oberen beiden Verfahren aus Bild 2 zur Simulation herangezogen, die Dienstausswahl mit und ohne Koordinator. Eine einfache Simulationsumgebung für dieses Vorhaben wurde erstmals im Projekt CatNet [CATN03a] entwickelt und seitdem weiterentwickelt und erweitert. Dieses Kapitel soll die formulierte These bzgl. des Einflusses der Dienstausswahl auf die Leistungsfähigkeit der SOA untermauern und einen Vergleich der Verfahren ziehen.

### 3.1 Netzwerkattribute und Hypothesen

Zum Vergleich wurden die zu untersuchenden Verfahren in ein Netzwerk, bestehend aus 106 Netzknoten eingebettet. Auf den 106 Knoten waren 75 Clients (Nachfrager) an den Aussenkanten verteilt, den so genannten Edges, auf allen Knoten konnten dagegen Dienste liegen (Bild 3a). Das Netz sollte in verschiedenen Simulationsläufen unterschiedliche Eigenschaften bezüglich der physikalischen Konzentration und der stetigen Verfügbarkeit der Dienste haben, um damit relativ einfach auf existierende Netze abstrahiert werden zu können. Dazu wurden jeweils fünf Stufen der Konzentration und der Verfügbarkeit (bzw. Ausfallwahrscheinlichkeit) der Dienste in den Simulator implementiert. Die Dienste wurden bei Initialisierung nach dem Zufallsprinzip in fünf verschiedenen Konzentrationen verteilt. Folglich wurden für 2 Verfahren jeweils 25, also insgesamt 50 Simulationsläufe durchgeführt (Bild 3b).

Die Dichte bzw. Konzentration der Dienste kann in Netzen sehr unterschiedlich sein. In so genannten Content Distribution Netzen (CDN) zur Verteilung von Web Content liegen die Dienste physikalisch sehr konzentriert, da die Webseiten von nur wenigen Servern millionenfach an die Clients verteilt werden. Man spricht davon, dass die Dienste hier sehr dicht konzentriert sind. Für die Simulation wird dieses Szenario mit Dichte 0 repräsentiert. Dagegen sind Dienste in Ubiquitous Computing Szenarien meist weit verteilt und an einem physikalischen Ort liegt nur ein Dienst physikalisch. Dichte 1 bis 4 stellen die Übergänge zur Veranschaulichung des Verhaltens der in Kapitel 1.3 vorgestellten Metriken dar.

Sind ein Netz und seine Knoten über den Beobachtungszeitraum statisch in der Verfügbarkeit der Dienste und Links, so spricht man von statischen Netzen. Diese Netze liegen vor, wenn die Teilnehmer Ihre Verfügbarkeit durch organisatorische Maßnahmen garantieren können, z.B. Unternehmensnetzwerke. Diese werden in der Simulation mit der Dynamikstufe 0 repräsentiert. Auf der anderen Seite gibt es Netze, deren Dienstressourcen nur kurze Zeit verfügbar sind, und neue Dienste

hinzukommen. In Peer-to-Peer File Sharing Anwendungen kann dieses Verhalten etwa beobachtet werden. Die Stufen dazwischen sollen einen Übergang dieser beider Extreme herstellen.

Innerhalb eines Experimentes senden die 75 Clients insgesamt 2000 Nachfragen, die Dauer der exklusiven Dienstbeanspruchung, also des Vertragsverhältnisses, betrug konstant 50ms, danach war der Dienst wieder frei verfügbar. Die Abstände der Anfragen eines einzelnen Clients wurden so gesteuert, dass das System ständig unter Last steht (Ergebnisse der Experimente zur Bestimmung der Eingangswerte finden sich unter [CATN03a]). Die verschiedenen Eingangswerte für die Dichte und Dynamik sind in Tabelle 1a und b dargestellt.

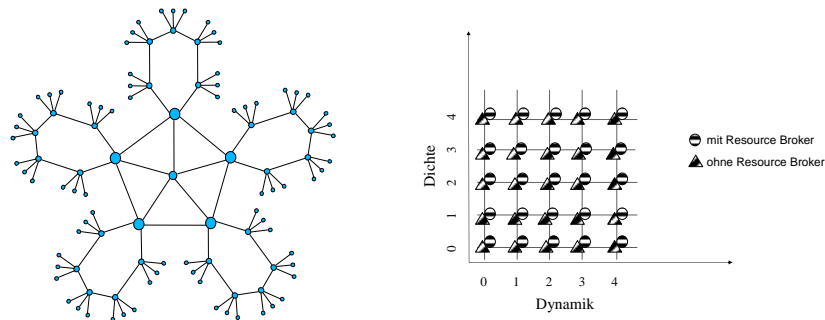


Bild 3a/b. Netztopologie und Experimentraum mit den Dimensionen Dichte und Dynamik

Dichte	Verteilung der 300 Dienste auf die vorhandenen Knoten	Dynamik	Wahrscheinlichkeit des Dienstausfalls in % (Messung alle 200ms)
0	6 Knoten mit je 50 Diensten	0	0
1	15 Knoten mit je 20 Diensten	1	15
2	25 Knoten mit je 12 Diensten	2	30
3	50 Knoten mit je 6 Diensten	3	45
4	75 Knoten mit je 4 Diensten	4	60

Tabelle 1a und b. Simulationseingangswerte für die Dichte und die Dynamik

### 3.2 Der Netzwerksimulator J-Sim/TCL

Der Unterbau der Simulationsumgebung wird von J-Sim bereitgestellt. J-Sim ist eine objektorientierte, komponentenbasierte Umgebung zur diskreten Simulation

von frei definierbaren Netzwerktopologien [JSIM03]. Die Netzwerkeigenschaften (Protokolle, Bandbreiten, Verlustraten etc.) können dabei frei bestimmt werden und tragen damit zu einer möglichst wirklichkeitsgetreuen Simulationsmodellierung bei. Die Simulation findet auf TCP/IP bzw. UDP/IP Paketebene statt und lässt vielfältige Freiheitsgrade in der Konfiguration zu. J-Sim kann derartig konfiguriert werden, dass ein spezifisches Anwendungsschichtnetz simuliert wird. Zudem können verschiedene Softwareagenten auf die Knoten aufgesetzt werden, die autonom handeln, und Ihren Nutzen zu maximieren versuchen. In der im Rahmen dieses Beitrages durchgeführten Simulation wurden spezielle Agenten für Kunden, Speicherplatzressourcen und Dienste programmiert und auf die entsprechenden Knoten gesetzt. Die Konfiguration des Netzwerkes erfolgt über Tcl/Tk-Skripte, die dynamisch generiert werden.

### 3.3 Simulationsergebnisse

In diesem Abschnitt finden sich die aktuellen Ergebnisse der Simulation. Aus Platzgründen wird ausschließlich auf die Metrik der Allokationsrate (RAE) eingegangen.

#### 3.3.1 Allokationseffizienz ohne Resource Broker

Bild 4a zeigt mit Hilfe eines Oberflächendiagramms den grafischen Verlauf der Allokationseffizienz über die Szenarien hinweg. Es ist zunächst erkennbar, dass die besten Werte erzielt werden für Netze, die eine geringe Ausfallwahrscheinlichkeit (Dynamik 0) aufweisen. Hier werden Werte von knapp 90% erreicht. Bei ansteigender Dynamik der Dienste fällt die Allokationsrate jedoch schnell ab. Bei einer Ausfallwahrscheinlichkeit von 60% (Dynamik 4) können bei Dichtestufe 0 nur noch knapp die Hälfte der Anfragen befriedigt werden. Auffällig ist, dass bei einer breiteren Verteilung der Dienste, also dem Ansteigen der Dichtestufe, diese Rate wiederum steigend ist. In Dichtestufe 4 werden noch 73% der Anfragen erfolgreich zugewiesen und die Transaktion abgewickelt. Somit beträgt das Defizit bei Erhöhung der Ausfallwahrscheinlichkeit unter 20%. Eine breitere Verteilung der Dienste fängt also die steigende Ausfallwahrscheinlichkeit ab; offensichtlich kann der Ansatz unter Abwesenheit eines Resource Brokers besser mit physikalisch verteilten Dienststandorten umgehen.

#### 3.3.2 Allokationseffizienz mit Resource Broker

Bild 4b demonstriert den Verlauf der Allokationsrate bei Einsatz eines Resource Brokers. Während in Dichte 0 und Dynamik 0 über 97 % der Anfragen befriedigt werden, erkennt man, dass bei Anstieg der Verteilung ein leichter Rückgang erkennbar ist (bis auf 87% bei Dichte 0). Dieses Ergebnis liegt auf der Hand, da in statischen Netzen eine vollständige Sicht über das Netz erreicht werden kann und damit die Zuweisung recht nah am Optimum liegt. Erhöht sich nun die Verteilung der Dienste, so ist dies mit einer Komplexitätssteigerung zu vergleichen: Der

Resource Broker hat Probleme mit Diensten die an vielen, verschieden entfernten Orten liegen. Erhöht man zusätzlich die Ausfallwahrscheinlichkeit, so ist das Defizit weit deutlicher erkennbar. In Dynamikstufe 4 werden bei Konzentration der Dienste nur 43% der Anfragen befriedigt, in Dichte 4 und Dynamik 4 nur noch 33% der Anfragen.

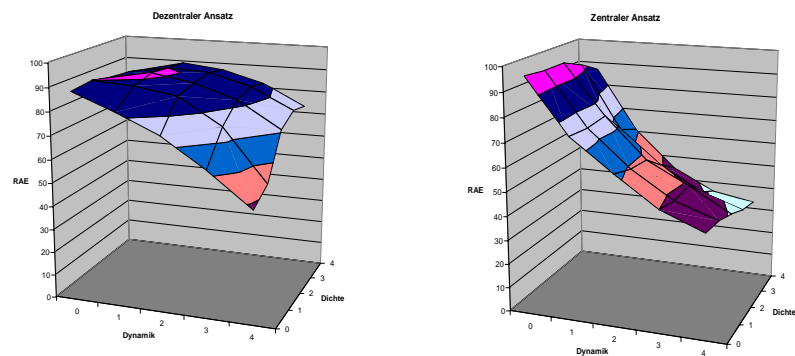


Bild 4a und b. Gemessene Allokationseffizienz ohne und mit Resource Broker

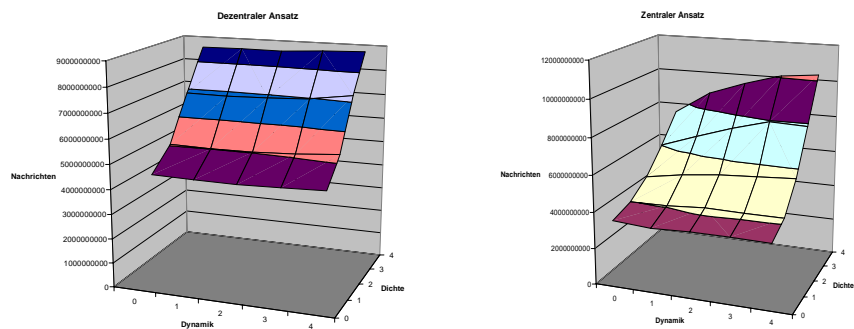


Bild 5a und b. Anzahl der Kontrollnachrichten ohne und mit Resource Broker

### 3.3.4 Vergleich der gesendeten Nachrichten auf IP-Ebene

Bild 5a und b zeigt die Anzahl der versandten Nachrichten bzw. IP-Paketen zwischen den einzelnen Knoten bzw. Datenübertragungseinrichtungen. Zunächst bemerkt man die recht hohe Zahl von Nachrichten, die durch das Fluten des Netzes zur Dienstfindung und den Nachrichten zur Verhandlung eines Dienstes bedingt ist. Es fällt auf, dass die Variation der Dynamik keinen erheblichen Einfluss auf die Menge der Nachrichten hat, wohl aber die der Konzentration der Dienste. Sowohl mit als auch ohne Koordinator steigt die Nachrichtenanzahl an. Während bei geringen Dichtestufen die Zahl der Nachrichten zwischen den beiden Verfahren nur gering unterschiedlich ist, wird mit steigender Dichtestufe dieser Unter-

schied größer. Dies ist begründet durch eine höhere Anzahl an Nachrichten, die dem Koordinator zugeführt werden müssen. In Dichte 4 werden mit Koordinator ca. 10 Milliarden Nachrichten versandt, ohne Koordinator knapp 9 Milliarden, was einen Unterschied von 10% ausmacht.

Fasst man die Ergebnisse zusammen, so wird die Robustheit des dezentralen Ansatzes bei variierender Verfügbarkeit der Ressourcen ersichtlich: Während die zentrale Ressourcenallokation in Bezug auf Skalierbarkeit deutlich Einbußen erfährt und mit der Netzdynamik nicht geeignet umgehen kann, reagiert das katalaktische, dezentrale Verfahren weit geringer auf Dichteveränderungen und Dynamikveränderungen. Steigende Dynamik kann etwa mit Erhöhen der Verteilung der Dienste begegnet werden.

## 4 Fazit und Ausblick

Der vorliegende Beitrag bezieht sich auf Dienstausswahlverfahren in serviceorientierten Overlaynetzen. Auf die Problematik der Dienstselektion bzw. Dienstausswahl im on-demand Computing wurde zunächst verstärkt eingegangen. Die in Kapitel 2 vorgestellten Verfahren weisen jedoch Mängel und Unvollständigkeiten auf, teilweise in Bezug auf Skalierbarkeit und in der Qualität der Allokation, so dass sie nicht uneingeschränkt zu empfehlen sind.

Mit der vorgestellten und durchgeführten Simulation konnte exemplarisch gezeigt werden, dass die Dienstausswahl zunächst überhaupt einen Einfluss auf die Leistungsfähigkeit hat und weiterhin, dass dezentrale Dienstausswahlmethoden unter Ausschluss eines Resource Brokers mit ökonomischen Verfahren in verteilten, dynamischen Netzen im Vergleich bessere Ergebnisse erreichen. Die Simulation hat überdies gezeigt, dass Netze durch dezentrale Mechanismen koordiniert werden können, z.B. mit Hilfe ökonomischer, rationaler Verfahren, wie z.B. der Katalaktaxie. Die Robustheit gegenüber wachsender Ausfallwahrscheinlichkeit und breiter Verteilung der Dienste ist deutlich erkennbar. Der prototypische Beweis der Eignung des Konzeptes in einer existierenden, heterogenen, serviceorientierten Infrastruktur steht aber noch aus und wird derzeit innerhalb eines EU-Folgeprojektes erarbeitet [CATN03b].

Themen wie „IT-Utility“, „Adaptive Computing“ oder „Utility Computing“, also der beliebige Bezug von IT-Dienstleistungen je nach Bedarf werden in der Wissenschaft als ein neues Paradigma diskutiert. Für Unternehmen werden diese Begriffe aber erst dann eine realistische Option zur Eigenbereitstellung von Diensten, wenn sich das Thema On-demand Computing so weit etabliert hat, dass die nötigen Kapazitäten zur Verfügung stehen und die geforderten architektonischen Eigenschaften erfolgreich umgesetzt sind.



Es bleibt abzuwarten, ob die Rechenleistung verschiedener Anbieter tatsächlich frei austauschbar wird, so wie es bei den ganzheitlichen Ansätzen Grid Computing und On-demand Computing geplant ist. Es wäre sicherlich ein beispielhafter Erfolg, wenn die Anbieter nicht versuchen würden, offene Standards durch proprietäre zu tauschen um damit Kunden durch hohe Wechselkosten in den Lock-in Effekt zu führen um damit Gewinne zu maximieren.

Die Frage, ob und bis wann on-demand ein etabliertes Phänomen wird, bleibt. Die IBM will in den nächsten Jahren bis zu 10 Milliarden USD jährlich ausgeben, um ODC erfolgreich auf dem Markt zu platzieren [Farb02]. Aber wird es ein nachhaltiges Modell für die IT-Bereitstellung in Unternehmen? Dies wird schließlich abhängig sein von einer großen Anzahl von Einflussfaktoren, einschließlich des breiten Einsatzes der Kerntechnologien, um die Initiative in den nächsten Jahren unterstützen. Zusätzlich müssen IT-Manager die Herausforderung annehmen, diesen kulturellen Wandel zu führen [Shan04].

## 5 Literaturverzeichnis

- [AdHu00] Adar, E.; Huberman, B.A.: Free Riding on Gnutella. *First Monday* 5(10) (2000).
- [Akam04] Akamai Webseite, 2004, <http://www.akamai.com>, Abruf am 200405-21.
- [Ami<sup>+</sup>98] Amir, E.; McCanne, S.; Katz, R.H.: An active service framework and its application to real-time multimedia transcoding. In: *Proceedings of ACM SIGCOMM'98*, Vancouver, Canada, 1998.
- [Ard<sup>+</sup>01] Ardaiz, O.; Freitag, F.; Navarro, L.: Multicast Injection for Application Network Deployment. 26<sup>th</sup> IEEE Conference on Local Computer Networks. Tampa, USA, 2001.
- [Ard<sup>+</sup>03] Ardaiz, O.; Artigas, P.; Freitag, F.; Navarro, L.; Eymann, T.; Reinicke, M.: Decentralized Resource Allocation in Application Layer Networks. In: *Proceedings of the 3<sup>rd</sup> IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003)*, 12.-15. Mai 2003, Tokyo, Japan.
- [Bal<sup>+</sup>03] Balakrishnan, H.; Kaashoek, F.; Karger, D.; Morris, R.; Stoica, I.: Looking up data in P2P systems, *Communications of the ACM*, February 2003, S. 43-48.
- [Bel<sup>+</sup>03] Bell, W.; Cameron, D.; Capozza, L.; Millar, P.; Stockinger, K.; Zini, Floriano: OptorSim - A Grid Simulator for Studying Dynamic Data Replication Strategies. *International Journal of High Performance Computing Applications*, 17(4), 2003.
- [Best95] Bestavros, A.: Demand-based Dissemination for Distributed Multi-media Application. In: *Proceedings of the ACM/ISMM/IASTED International Conference on Distributed Multimedia Systems and Applications*, Stanford, Kalifornien, 1995.
- [Bloo03b] Bloomberg, J.: Just what is "on demand," anyway? [http://searchwebservices.techtarget.com/tip/0,289483,sid26\\_gci900710,00.html](http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci900710,00.html), Abruf am 10.11.2003. {nur für geschlossene Benutzergruppe zugänglich}

- [Bren96] Brenner, T.: Learning in a Repeated Decision Process: A Variation-Imitation-Decision Model. Report No. #9603. Max-Planck-Institut für die Erforschung von Wirtschaftssystemen: Jena 1996.
- [Buyy02] Buyya, R.: Economic-based Distributed Resource Management and Scheduling for Grid Computing. Ph.D. Thesis. Monash University, Melbourne, Australia, 2002, <http://www.buyya.com/thesis/thesis.pdf>, Abruf am 2004-06-19.
- [CATN03a] CATNET Project: Catallaxy Evaluation Report. Report No. D3. Barcelona: 2003a, <http://research.ac.upc.es/catnet/pubs/D3.pdf>, Abruf am 2003-07-24
- [CATN03b] CATNETS Projektantrag. Nichtveröffentlichter Antrag an die europäische Kommission, 2003b.
- [Cave00] Cave, D.: The Mojo solution, (9.10.2000), [http://archive.salon.com/tech/view/2000/10/09/mojo\\_nation](http://archive.salon.com/tech/view/2000/10/09/mojo_nation), erstellt am 2000-11-09.
- [Cha<sup>+</sup>98] Chandra, P.; Fischer, A.; Kosak, C.; Ng, E.; Steenkiste, P.; Takahashi, E.; Zhang, H.: Darwin: Customizable Resource Management for Value-Added Network Services. In: Proceedings of the Sixth IEEE International Conference on Network Protocols (ICNP'98), Austin, Texas, 1998.
- [Clar99] Clark, I.: A distributed decentralised information storage and retrieval system. 1999, <http://freenet.sourceforge.net/Freenet.ps>. Abruf am 8. Dezember 2002.
- [Eggs01] Eggs, H.: Vertrauen im Electronic Commerce – Herausforderungen und Lösungsansätze. Deutscher Universitätsverlag, Wiesbaden 2001.
- [Eyma00] Eymann, T.: Avalanche – ein agentenbasierter dezentraler Koordinationsmechanismus für elektronische Märkte. Ph.D. Thesis. Albert-Ludwigs-Universität Freiburg, 2000, <http://www.freidok.uni-freiburg.de/volltexte/147/>, Abruf am 2004-05-01.
- [Farb02] Farber, D.: On-demand computing: What are the odds?. <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2896789,00.html>, erstellt am 2002-06-11.
- [FoKe97] Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputing Applications 11(2) (1997), S. 115-129.
- [Fran04] Frank, H.-J.: IT-Outsourcing: Zwischen Ökonomie und Nouvelle Cuisine. [http://www.dbresearch.de/PROD/DBR\\_INTERNET\\_DE-PROD/PROD0000000000073793.PDF](http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD0000000000073793.PDF), Abruf am 2004-04-12.
- [Fre<sup>+</sup>02] Frey, J.; Tannenbaum, T.; Livny, M.; Foster, I.T.; Tuecke, S.: Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing 5(3) (2002), S. 237-246.
- [IBM03] Robust. Dynamisch. Wertschöpfend. Flexibel. [http://www-5.ibm.com/de/software/sysmgmt/pdf/tiv\\_bro\\_v13\\_DEU\\_1212.pdf](http://www-5.ibm.com/de/software/sysmgmt/pdf/tiv_bro_v13_DEU_1212.pdf), Aufruf am 2004-07-24.
- [JSIM03] The J-Sim Website. <http://www.j-sim.org>. Abruf am 2003-02-21.
- [KaRo95] Kagel, J.H., Roth, A.E.: The handbook of experimental economics. Princeton, N.J: Princeton University Press 1995.

- [KoDa99] Korupolu, M.R.; Dahlin, M.: Coordinated Placement and Replacement for Large-Scale Distributed Caches. Proceedings of IEEE Workshop on Internet Applications, 1999.
- [Kril04] Krill, P.: IBM proposes convergence of Web services, grid computing. [http://www.infoworld.com/article/04/01/20/HNgridspecs\\_1.html](http://www.infoworld.com/article/04/01/20/HNgridspecs_1.html), 2004-01-20, Abruf am 2004-07-27.
- [Lit<sup>+</sup>88] Litzkow, M. J.; Livny, M.; Mutka, M.: Condor – a hunter of idle workstations. In: Proceedings of the 8th International Conference of Distributed Computing Systems, San Jose, Kalifornien, 1988.
- [Liu<sup>+</sup>02] Liu, C.; Yang, L.; Foster, I.; Angulo, D.: Design and Evaluation of a Resource Selection Framework for Grid Applications. In: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 2002 (HPDC'02) (2002), S. 63.
- [Matt03] Frank, M.: On-Demand-Computing ist eine Managementaufgabe. <http://www.computer-zeitung.de/O/50/Y/84021/VI/10061386/default.aspx>, Abruf am 2003-11-11.
- [MiDr88] Miller, M.; Drexler, E.: Markets and Computation: Agoric Open Systems. In: The Ecology of Computation, Bernardo Huberman (Hrsg.) Elsevier Science Publishers/North-Holland 1988.
- [Mil<sup>+</sup>02] Milojevic, D.S.; Kalogeraki, V.; Lukose, R.; Nagaraja, K.; Pruyne, J.; Richard, B.; Rollins, S.; Xu, Z.: Peer-to-Peer Computing. Report No. HPL-2002-57. Palo Alto: Hewlett Packard Labs 2002, <http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html>, Abruf am 2002-05-01.
- [Mojo03] Mojo Nation Webseite, 2003, <http://www.mojonation.net>, Abruf am 2003-07-21.
- [NGG03] Next Generation Grid(s), European Grid Research 2005-2010, Expert Group Report. <http://www.cordis.lu/ist/grids/index.htm>. Abruf am 2003-07-18.
- [Pic<sup>+</sup>96] Picot, A.; Reichwald, R.; Wigand, R.: Die grenzenlose Unternehmung. Information, Organisation und Management. Gabler, Wiesbaden 1996
- [Pre<sup>+</sup>02] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: Numerical Recipes in C++ - The Art of Scientific Computing. Cambridge University Press: Cambridge, Massachusetts, 2002.
- [Pru81] Pruitt, D.G.: Negotiation behavior. In: Organizational and occupational psychology. New York: Academic Press 1981
- [RaAg99] Rabinovich, M.; Aggarwal, A.; RaDaR: A scalable architecture for a global Web hosting service. The 8th International World Wide Web Conference, Toronto 1999.
- [Rat<sup>+</sup>01] Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R.; Shenker, S.: A Scalable Content-Addressable Network. [http://citeseer.nj.nec.com/ratnasamy01\\_scalable.html](http://citeseer.nj.nec.com/ratnasamy01_scalable.html), Abruf am 2004-06-20.
- [Rech04] Rechenkraft.net. <http://www.rechenkraft.de>, Abruf am 2004-07-26.

- [Sabb04] Sabbah, D.: Bringing Web Services Together. [http://www.globus.org/wsrfl/sabbah\\_wsrfl.pdf](http://www.globus.org/wsrfl/sabbah_wsrfl.pdf), Abruf am 2004-06-25.
- [Sear03] On-demand computing. [http://searchcio.techtarget.com/sDefinition/0,,sid19\\_gci903730,00.html](http://searchcio.techtarget.com/sDefinition/0,,sid19_gci903730,00.html), Abruf am 2003-11-10.
- [Shan04] Shankland, S.: IBM: On-demand computing has arrived. [http://news.com.com/IBM:+On-demand+computing+has+arrived/2100-7784\\_3-5106577.html?part=business2-cnet](http://news.com.com/IBM:+On-demand+computing+has+arrived/2100-7784_3-5106577.html?part=business2-cnet), Abruf am 2004-06-26.
- [Sil<sup>+</sup>03] Silberstein, M.; Factor, M.; Lorenz, D.: DYNAMO - DirectorY, Net Archiver and Mover. In: Proceedings of Grid Computing - GRID 2002: Third International Workshop, Baltimore, MD, USA, 18. November, 2002. Lecture Notes in Computer Science 2536, Springer-Verlag Heidelberg, 2003: S. 256 – 267.
- [Smit62] Smith, V.L.: An experimental study of competitive market behavior. In: Journal of Political Economy, Vol. 70 (1962): S. 111-137.
- [Snell03] Snelling, D.; Priol, T. et al.: Next Generation Grid(s). European Grid Research 2005 - 2010. Brussels: Information Society – DG, Grids for Complex Problem Solving 2003, <http://www.cordis.lu/ist/grids/index.htm>, Abruf 2004-06-25.
- [SOA04] Service-oriented Architecture (SOA) definition. [http://www.service-architecture.com/web-services/articles/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html), Abruf am 2004-06-23.
- [Tesf97] Tesfatsion, L.: How economists can get alive. In: Arthur, W.B., Durlauf, S., Lane, D.A. (eds.): The Economy as a Evolving Complex System II. Santa Fe Institute Studies. Redwood City, CA: Addison Wesley 1997: S. 533-564.
- [W3C01] W3C: Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl>, 2001-03-15, Abruf am 2004-06-27.
- [W3C04] SOAP Version 1.2 Part 1: Messaging Framework. <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>, Abruf am 2004-10-12.
- [Webo04] Webopedia: What is UDDI? – A Word Definition from the Webopedia Computer Dictionary. <http://www.webopedia.com/TERM/U/UDDI.html>, Abruf am 2004-06-27.