Anfrageverarbeitung und kollaborative verteilte Arbeit in Umgebungen mit unzuverlässigen Knoten

Holger Steinhaus
Otto-von-Guericke-Universität Magdeburg
hsteinha@cs.uni-magdeburg.de

Zusammenfassung

Eine zentralisierte Datenspeicherung und Koordination stellt für viele verteilte Algorithmen einen bedeutenden Flaschenhals hinsichtlich Skalierbarkeit und Zuverlässigkeit dar. Im Folgenden wird ein Ansatz vorgestellt, der u.a. zur Realisierung eines vollständig dezentralen verteilten Webcrawlers eingesetzt wird. Seine wesentlichen Eigenschaften bestehen in Durchsetzung einer Verpflichtung aller Knoten zur Mitarbeit und in der weitgehenden Resistenz des Ansatzes gegenüber gezielten Angriffen durch bösartige Knoten.

Dieses Szenario erfordert gezielte Strategien zur Koordination, Datenspeicherung und zur Anfrageverarbeitung. Während das grundsätzliche Ziel des Systementwurfs in der Beantwortung möglichst aller Anfragen von Knoten liegen sollte, würde eine unbedingte Beantwortung ohne Berücksichtigung der vom anfragenden Knoten erbrachten Leistungen die Motivationsgrundlage für die Mitarbeit in Frage stellen. Andererseits birgt jeder Mechanismus, der bestimmte Knoten diskriminiert, auch eine Gefahr: Oftmals kann dieser durch einen bösartigen Angreifer gezielt zur Durchführung eines Denial-of-Service-Angriffs und damit zur (Zer-)Störung des gesamten Systems missbraucht werden. Die vorgestellten Mechanismen wurden daher in Form von Simulationsexperimenten auf Stabilität und Effizienz untersucht, mögliche Nebenwirkungen werden durch die Simulation von Angriffen betrachtet.

1 Einleitung

Viele Anwendungen im Umfeld des World Wide Web erfordern leistungsfähige Webcrawler. Die aktuelle Größe des Webs wird auf ca. 11.5 Mrd. Seiten geschätzt (Stand 2005, siehe [GS05]). Will man diese in endlicher Zeit komplett verarbeiten, muss das Crawling massiv parallel stattfinden. Dabei werden je nach konkreter Anwendung für jede dieser Seiten teilweise hunderte Datensätze extrahiert und gespeichert. Die resultierende Größe einer solchen Datenbank kann dabei die Summe der Größen der ursprünglichen Webseiten[1] überschreiten. Dieses Volumen liegt weit ausserhalb dessen, was ein einzelner Datenbank-Server effizient verwalten kann. Weiterhin spielen eine Vielzahl von Eigenschaften, die herkömmliche DBMS auszeichnen, in unserer speziellen Anwendung eine untergeordnete Rolle. So ist die Struktur der zu speichernden Daten und auch der darauf ausgeführten Anfragen vergleichsweise einfach und homogen. Garantien gegen Datenverlust, Transaktionseigenschaften und Konsistenzsicherung werden nicht benötigt und sind in Systemen mit sehr großen Knotenzahlen ohnehin nur unter Inkaufnahme hoher Kosten realisierbar.

Stattdessen liegt der Schwerpunkt der Anforderungen in einer außerordentlichen Skalierbarkeit: Das Ziel unserer Arbeit liegt in der Entwicklung einer Architektur, die für Systeme in der Größenordung von einigen 10000 bis zu mehreren 100k Knoten ausgelegt ist. Lösungen zur verteilten Datenspeicherung, die diese Größenordnungen abdecken, existieren in Form diverser Ansätze zur Realisierung von verteilten Hashtabellen [R$^+$01, S$^+$01]. Diese lassen jedoch den wichtigen Aspekt der Zuverlässigkeit von Knoten außer Acht: Insbesondere sehr große Systeme mit Tausenden von Knoten werden selten unter

[1] Legt man die durchschnittliche Größe einer Webseite mit 10-20 kB zu Grunde, führt das zu einem Gesamtvolumen von 115-230 TByte

der Kontrolle einer einzigen Organisation stehen. Vielmehr erscheint es attraktiv, diese massive Verteilung in einem offenen Peer-to-Peer-Verbund zu realisieren, der eine Vielzahl von Betreibern in einer mehr oder weniger anonymen Umgebung einbindet. Aus diesen Gründen ist es unerlässlich, Knoten, die sich anderes als beabsichtigt verhalten, von vornherein in Entwurfsüberlegungen einzubeziehen.

Nach unseren Erkenntnissen ist es günstig, die verteilte Datenspeicherung in Verbindung mit den Mechanismen der Koordination der verteilten Arbeit zu betrachten: Es gibt hier eine Vielzahl von Parallelen, die eine enge Kopplung dieser beiden Funktionalitäten nahelegen: Wir stellen im Folgenden einen integrierten Ansatz zur Koordination vor, der u.a. auf verteilt gespeicherte Informationen über die Verlässlichkeit von einzelnen Knoten zurückgreift. Die Speicherschicht ihrerseits ist wiederum auf die Fähigkeiten der Koordinationsschicht zur Begrenzung des Einflusses von unzuverlässigen Knoten angewiesen.

2 Systemarchitektur

Der eingangs geforderte Verzicht auf zentrale Koordination ist dann möglich, wenn ein allen Knoten bekanntes Verfahren existiert, das das ursprüngliche verteilt zu lösende Problem in unabhängig lösbare Teilprobleme zerlegt. Zur verteilten Speicherung von gleichartig strukturierten Daten haben sich hierzu verschiedene Verfahren etabliert (z.B. [S^+01, R^+01]), die das Prinzip einer verteilten Hashtabelle (Distributed Hashtable, DHT) implementieren. Dazu wird eine allen Knoten bekannte Hashfunktion auf einen Schlüssel angewendet, der damit in den Wertebereich der Hashfunktion abgebildet wird. Dieser (vorab bekannte) Wertebereich kann unabhängig von den Eigenschaften der Schlüsseldaten oder der Hashfunktion auf die teilnehmenden Knoten verteilt (im Folgenden: partitioniert) werden, ohne das dazu eine zentrale Koordination nötig würde. Durch die Verwendung mehrerer verschiedener Hashfunktionen lässt sich auf einfachem Wege eine kontrollierte Replikation erreichen. Der strukturierte Schlüsselraum bietet die Möglichkeit, Nachrichten an Teilnehmer weiterzuleiten, ohne den eigentlichen Adressaten direkt zu kennen. Dieser Vorgang ist u.a. in [R^+01, BB04] detailliert dargestellt.

Auch für die Koordination von verteilter Arbeit, wie z.B. des eingangs erwähnten Webcrawlings ist eine solche Struktur geeignet, sofern ein geeigneter Schlüsselwert und eine entspechende Hashfunktion zur Verfügung stehen. In unserem Beispiel bietet es sich an, Adressen von Webseiten als Input für eine Hashfunktion, wie z.B. MD5, zu verwenden. Durch die Verwendung von Hashfunktionen mit identischem Wertebereich lassen sich so eine oder mehrere Relationen verteilt speichern sowie verschiedenartige Arbeitsaufgaben verteilen und parallelisieren.

3 Verhalten von Knoten

Wodurch kommt nun Unzuverlässigkeit auf der Ebene einzelner Knoten zu Stande? Zum einen sind hier technisch bedingte Ausfälle und unzuverlässige Kommunikationswege zu betrachten. Dieses Problem wurde in der Vergangenheit intensiv untersucht und soll hier nicht weiter diskutiert werden. Die Wahrscheinlichkeit für Datenverluste durch zufällige Ausfälle kann dabei durch Wahl eines geeigneten Replikationsgrades nahezu beliebig reduziert werden. Im Fokus unserer Arbeit steht dagegen Unzuverlässigkeit, die durch bestimmtes (absichtliches) Verhalten einzelner Knoten verursacht wird. Dieses kann grundsätzlich in zwei Kategorien eingeordnet werden.

Rationales Verhalten liegt vor, wenn einzelne Knoten ihre Kosten bei vorgegebenen Erfolg minimieren. Daraus kann für das Gesamtsystem durchaus Unzuverlässigkeit entstehen. Räumt z.B. ein System neuen Teilnehmern einen Vorschuss an Vertrauen ein, fordert ansonsten aber eine Beteiligung an der verteilten Arbeit, existiert nur eine rationale Strategie: Ein Knoten tritt dem System bei und verbraucht diesen Vorschuss. Anschließend verlässt er das System und tritt unter einer neuen Identität erneut bei. Der Wechsel der Identität ist ohne eine zentrale Kontrollinstanz kaum zu verhindern. Allerdings kann ein geeignetes Protokoll, das unerwünschtes Verhalten entsprechend teuer macht, solche rationale Knoten zur *kooperativen* Mitarbeit motivieren. Abhängig von der Ausgestaltung dieser Me-

chanismen kann es also für einen Knoten rational sein, sich an der verteilten Arbeit und Datenhaltung zu beteiligen oder nicht. Ein Entwurfsschwerpunkt von dezentral koordinierten Systemen besteht daher in der Motivation zur Beteiligung. Untersuchungen in existierenden Peer-to-Peer-Systemen [AH00] haben gezeigt, dass ohne solche Mechanismen unter den Nutzern sich nur wenige befinden, die sich an der Erbringung der verteilten Arbeit beteiligen. Unser Systementwurf sieht daher die Möglichkeit zur Nutzung der Ergebnisse der verteilten Arbeit durch Anfragen an das System nur dann vor, wenn sich der entsprechende Knoten zuvor hinreichend an der verteilten Arbeit *beteiligt* hat. Die Definition einer hinreichenden Beteiligung ist dabei abhängig von der Art und Struktur der verteilten Arbeit selbst. Im Beispiel des verteilten Webcrawlers errechnet sich diese Beteiligung ausschließlich aus der Anzahl der gecrawlen Webseiten, da der Aufwand für das Crawling alle anderen Tätigkeiten (wie z.B. für die Datenhaltung) um Größenordnungen überwiegt. Problematisch ist dabei die Beobachtung dieser Beteiligung: Der einzige Knoten, der alle dazu benötigten Informationen besitzt, ist der Crawlende selbst. Dessen Auskunft über die eigene Leistung ist aber wenig glaubwürdig, jeder rationale Teilnehmer würde hier falsche Angaben verbreiten, wenn keine Sanktionsmöglichkeiten bestehen. Bei vielen verteilten Algorithmen ist es jedoch relativ leicht, das Ergebnis auf seine Korrektheit zu überprüfen. In [SB05] stellen wir einen entsprechenden Algorithmus für den Anwendungsfall des verteilten Webcrawlers vor.

Unser Entwurf sieht weiterhin die verteilte Speicherung eines *Reputationswertes* je Knoten vor, der Informationen über das bisherige Verhalten eines Knotens aggregiert. Erfolgreiche/fehlgeschlagene Überprüfungen, erfolgreiche bzw. unberechtigt verweigerte Antworten auf vorherige Anfragen gehen in diesen Wert ein, indem Knoten *Feedback* an den/die für die Verwaltung des betreffenden Reputationswertes zuständigen Knoten übermitteln. Die Menge der Reputationswerte aller Knoten stellt wiederum eine Relation dar, die, wie eingangs beschrieben in Form einer verteilten Hashtabelle ohne zentrale Koordination verteilt gespeichert werden kann. Eine geeignete Zahl von Replikaten verhindert dabei gezielte Manipulationen durch einzelne Teilnehmer (vgl. [AD01]).

Bösartiges Verhalten hingegen wird unterstellt, wenn ein Knoten bereit ist, eigene Arbeit und Ressourcen einzubringen, um dem System zu schaden. In der Realität ist dieses Verhalten häufig zu beobachten, wie z.B. verteilte Denial-of-Service-Angriffe im Internet zeigen. Die Inkaufnahme von Aufwand macht bösartige Teilnehmer unempfindlich gegenüber Anreizen und Strafen. Die einzige Möglichkeit, deren Auswirkungen zu bechränken, besteht in einem schnellen Ausschluss.

Im Gegensatz zu rationalen Knoten, deren Ziel in der Erlangung von Nutzen durch Beitritt zum System liegt, unterliegt bösartiges Verhalten keinen solchen Beschränkungen. Eine mögliche, nach unseren Experimenten im Sinne des Angreifers besonders wirksame Form bösartigen Verhaltens besteht darin, sich nicht an der verteilten Arbeit zu beteiligen und die dabei eingesparten Ressourcen ausschließlich zur Schädigung des Systems einzusetzen. Nach unserem derzeitigen Stand der Erkenntnisse besteht die einzige Gegenmaßnahme gegen solche Teilnehmer in einem möglichst schnellen Abbruch jeder Interaktion. Dazu ist das im vorangegangenen Abschnitt erwähnte Reputationssystem geeignet: jeder Knoten, der eine bestimmte minimale Reputation unterschreitet, wird als *ausgeschlossen* betrachtet, alle Knoten verwerfen sämtliche von einem solchen Knoten erhaltenen Nachrichten. In Verbindung mit einer angemessenen Vorleistung, die ein neu beitretender Knoten in Form eines Arbeitsbeweises (Proof of Work, siehe u.a. [BB04]) erbringen muss, erweist sich dieser Ausschlussmechanismus als sehr wirksam.

Aufgrund der zentralen Bedeutung der Reputation der einzelnen Knoten, bietet es sich für bösartige Teilneher an, genau hier anzusetzen, indem unwahres Feedback über Teilnehmer verbreitet wird. Optimalerweise würde dieses falsche Feedback zum einen alle regulären Teilnehmer belasten, andere Angreifer aber durch falsches positives Feedback schützen. Allerdings sind bösartiger Knoten, die diese Strategie verfolgen, relativ leicht an ihrer nicht erbrachten Arbeitsbeteiligung erkennbar. Eine andere Modellierung von bösartigem Verhalten, das dieses Problem umgeht, muss zwangsläufig auf eine Implementierung zurückgreifen, die sich an der verteilten Arbeit zumindest teilweise beteiligt. Damit schrumpft aber auch der Kostenvorteil eines bösartigen Teilnehmers: ein nicht entdeckbarer bösartiger Teilnehmer würde damit die gleichen Kosten haben wie ein regulärer Knoten, in der Reputationsberechnung hätte er ebenfalls das gleiche Gewicht. Zur Verursachung eines signifikanten Schadens muß daher zumindest eine in zur Knotenzahl äquivalente Größenordnung an Angreifern aufgebracht werden. Im

folgenden Abschnitt wird diese Betrachtung näher quantifiziert.

In einem dezentralen System können Mechanismen gegen bösartiges Verhalten und zur Sicherung der Beteiligung nur dann wirken, wenn sie von einer Mehrheit der Knoten auch angewendet werden. Das ist mit der Definition von rationalem Verhalten vereinbar, wenn man ein Grundinteresse an den Resultaten der gemeinsamen verteilten Arbeit und deren Resultaten unterstellt. Ein Knoten wird als *kooperativ* bezeichnet, wenn er die hier vorgestellten Mechanismen anwendet. Wir gehen im Folgenden davon aus, dass die Mehrheit der Knoten kooperativ ist.

4 Evaluierung

Nachdem im vorangegangenen Abschnitt Mechanismen gegen unzuverlässige Knoten vorgestellt wurden, soll hier nun deren Effektivität näher betrachtet werden. Dazu wurde eine Simulation implementiert, die die in den vorangegangenen Abschnitten vorgestellten Konzepte umsetzt.

Die Simulation erfolgt dabei zyklenweise, in jedem Zyklus wird pro Knoten eine Anfrage abgesetzt. Im weiteren Verlauf des Zyklus beantworten kooperative Knoten die eingegangenen Anfragen, wenn der Anfragende eine ausreichende (simulierte) Arbeitsbeteiligung vorgibt. Der antwortende Knoten akzeptiert diese vorgegebene Beteiligung mit einer von der Reputation des Anfragenden abhängigen Wahrscheinlichkeit, anderenfalls überprüft er diese, wie in Abschnitt 3 dargestellt. Bösartige Teilnehmer verweigern die Antwort grundsätzlich, da sie aufgrund der fehlenden Beteiligung nicht über die erforderliche Datengrundlage zur Beantwortung von Anfragen verfügen. Im letzten Schritt wird Feedback über beantwortete bzw. nicht beantwortete eigene Anfragen sowie über die Ergebnisse von eventuell durchgeführten Überprüfungen übermittelt und die Reputationswerte der entsprechenden Knoten angepasst. Dabei geben alle kooperativen Knoten wahrheitsgemäßes Feedback ab. Bösartige Teilnehmer hingegen bewerten ihresgleichen mit positivem, alle anderen jedoch mit negativem Feedback. Dieses Verhalten bildet damit einen *Koalitionsangriff* nach.

4.1 Auswirkungen bösartiger Teilnehmer

Unter Verwendung dieses Simulationsmodells soll nun die Einflussmöglickeiten einer Gruppe von Angreifern betrachtet werden. Dazu wird ein System aufgesetzt, das aus insgesamt 1000 Teilnehmern besteht. Abbildung 1 stellt dazu die Anzahl der kooperativen Teilnehmer (obere Fläche) sowie der Angreifer (untere Fläche) über den anfänglichen Anteil der Angereifer sowie über die Simulationszeit dar.

Die Abbildung zeigt, das das hier verwendete Modell von bösartigem Verhalten effektiv ist und dazu führt, dass bei hohen Anteilen von bösartigen Knoten durchaus nennenswerte Schäden in Form von fälschlicherweise ausgeschlossenen kooperativen Knoten verursacht werden können. Ab einem Anteil von 20% betrifft das ca. 29% der kooperativen Teilnehmer. Diese Zahl sollte vor dem Hintergrund betrachtet werden, dass ab einem Anteil von 50% die Bedingung der kooperativen Mehrheit nicht mehr erfüllt ist und damit prinzipiell keine sinvolle Unterscheidung zwischen kooperativen und nicht kooperativen Teilnehmern mehr möglich ist.

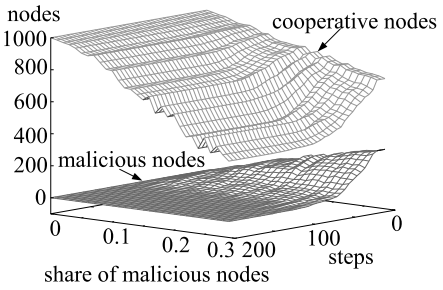

Abbildung 1: Entwicklung der Knotenzahl vs. Anteil bösartiger Knoten

4.2 Langzeitstabilität

Zur Analyse der Langzeitstabilität wird das Szenario leicht modifiziert: Einem System, das aus 1000 kooperativen Knoten besteht, tritt pro Zyklus ein neuerer Knoten bei. Dabei sind im Durchschnitt 10% der beitretenden Teilnehmer nicht kooperativ. Abbildung 2 stellt dabei sowohl die Anzahl der nicht kooperativen Knoten als auch den Aufwand dar, den die in Abschnitt 3 dargestellten Kontrollmechanismen

verursachen.

Die Entwicklung dieses Aufwands (normiert je Peer) ist von besonderer Bedeutung, da ein über die Knotenzahl oder über die Zeit zunehmender Aufwand ein klares Anzeichen für Skalierungs- und Stabilitätsprobleme darstellt. Nach einer anfänglichen Phase mit vergleichsweise großem Kontrollaufwand, verursacht durch die zum Startzeitpunkt identische Reputation aller Peers, entwickelt sich dieser eindeutig rückläufig. Einzelne unkooperative Knoten im System werden vergleichsweise schnell erkannt und eliminiert, große Gruppen wie im vorangegangenen Experiment untersucht, können sich nicht entwickeln. Die absolute Zahl von Überprüfungen liegt nach dem einer gewissen Zeit des Einschwingens auf einem Niveau von 0.1 pro Knoten und Zyklus.

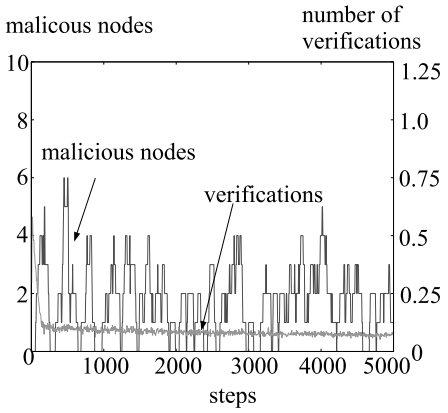

Abbildung 2: Beitritt neuer Knoten

5 Fazit

Die hier vorgestellten Mechanismen ermöglichen es, Systeme zur Verarbeitung von sehr großen Datenmengen zu realisieren. Mechanismen, die ursprünglich der reinen Datenspeicherung dienten, werden dabei erweitert, um zusätzlich eine vollständig dezentrale Koordination von verteilter Arbeit zu ermöglichen. In Verbindung mit entsprechenden Anreiz- und Kontrollmechanismen ist es möglich, in unzuverlässigen und sogar feindlichen Umgebungen einen effektiven Betrieb zu ermöglichen.

Literatur

[AD01] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, 2001.

[AH00] Eytan Adar and Bernardo A. Huberman. Free Riding on Gnutella. *Technical report, Xerox PARC*, 2000.

[BB04] E. Buchmann and K. Böhm. FairNet - How to Counter Free Riding in Peer-to-Peer Data Structures. In *Proc. of the International Conference on Cooperative Information Systems 2004, Agia Napa, Cyprus*, 2004.

[GS05] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05: 14th international conference on World Wide Web*, New York, NY, USA, 2005.

[R+01] Sylvia Ratnasamy et al. A scalable content-addressable network. In *Proceedings of the 2001 ACM SIGCOMM Conference*, 2001.

[S+01] Ion Stoica et al. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, 2001.

[SB05] Holger Steinhaus and Klemens Böhm. Verteiltes Webcrawling in einer Peer-to-Peer-Umgebung. *Datenbank Spektrum, Heft 13*, 2005.