

Geschwisterkampf - Oracle 12c und Oracle NoSQL im Cloud Benchmark -

Jascha Knack, Harm Knolle

**Hochschule Bonn-Rhein-Sieg
Sankt Augustin**

Schlüsselworte: NoSQL, Benchmark, Cloud Computing, YCSB

Abstract: Ist die erreichbare Performance von NoSQL-Datenbanken zur Unterstützung spezieller Cloud-Anwendungen tatsächlich so viel höher als die traditioneller relationaler Datenbanken? Dieser Frage soll mit dem von Yahoo! entwickelten „Cloud Serving Benchmark Framework“ (YCSB) nachgegangen werden. Im Benchmark treten die „Oracle NoSQL Database“ als NoSQL-Vertreter sowie die „Oracle Database 12c“ als Kandidat der relationalen Systeme gegeneinander an. Der Benchmark, die Workloads und die Resultate der Messungen werden erläutert und hinsichtlich der festgestellten Performance bewertet, verglichen und interpretiert.

Einleitung

Einen der letzten großen „Aufstände“ in der Welt der Datenbanken hat es in den 80er Jahren mit dem Aufkommen der objektorientierten Datenbanken gegeben. Bereits damals wollte man weg von streng normalisierten relationalen Datenstrukturen und hin zu mehr Semantik und komplex strukturierten Objekten auf Ebene der Speichertechnik. Die rein objektorientierten Systeme scheiterten damals jedoch nicht selten an den Performance-Anforderungen der überwiegenden Anzahl traditionell administrativ-betriebswirtschaftlicher Anwendungen. Die relationalen Datenbanken haben dagegen die objektorientierten Konzepte aufgegriffen, speichern intern weiterhin relational und bezeichnen sich seit den 90er Jahren vielfach als objektrelational.

Inzwischen gehören „Social Media“, „Big Data“ und „Cloud“-Technologien zu den signifikanten Stichworten, die die technische Entwicklung der letzten Jahre entscheidend mitgeprägt haben. Längst machen sich die Global Player wie Oracle, Google, Microsoft, Apple, Facebook oder Twitter Gedanken darüber, wie sich Terabytes von in der Cloud anfallenden Informationen besonders effektiv verwalten und zielgerichtet auswerten lassen [StMn11]. Längst reicht die eher administrativ-betriebswirtschaftlich geprägte „Data Warehousing“-Philosophie nicht mehr aus, gleichzeitig drei Fronten zu berücksichtigen: die rasant wachsende Datenmenge, die hohe Geschwindigkeit permanent einströmender Daten und die breite Vielfalt an strukturierten und unstrukturierten Daten. Nach fast 30 Jahren „Ruhe und Gelassenheit“ und einem kurzen Hype der XML-Datenbanken um die Jahrtausendwende befindet sich die Datenbankwelt derzeit inmitten einer „kambrischen Explosion“ neuartiger so genannter NoSQL-Speichertechniken [Edl14]. Hierbei handelt es sich vielfach um horizontal skalierbarer und einfach strukturierte Datenspeicher, die im Unterschied zu relationalen Systemen oft auf schemalosen Datenstrukturen basieren [EFH+11]. Derartige NoSQL-Systeme erfreuen sich mittlerweile nicht nur im Umfeld des Cloud-Computings, sondern vermehrt auch bei einfachen Datenbankanwendungen steigender Beliebtheit [StMn11, EFH+11].

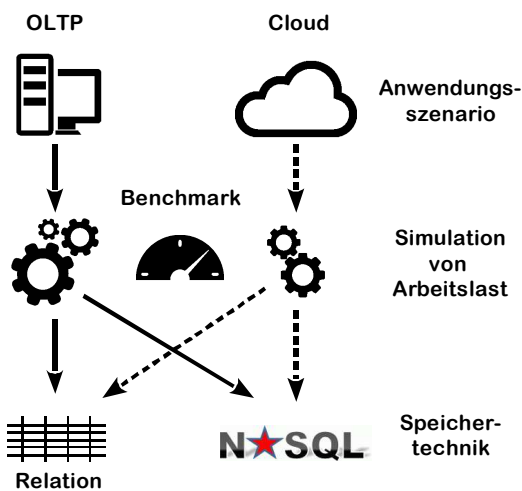


Abb. 1: Leistungsbewertung „NoSQL vs. Relational“

Die steigende Beliebtheit der NoSQL-Systeme hat mittlerweile auch zu einer Polarisierung innerhalb der Datenbank-Community geführt. Befürworter weisen auf die performance-fördernden Möglichkeiten der horizontalen Skalierung hin, über die zusätzlich Ausfallsicherheit und Verfügbarkeit garantiert werden kann. Die relational geprägte Welt bemängelt dagegen die Abschwächung der Konsistenz und das Fehlen mächtiger deskriptiver Anfragesprachen wie SQL. Eine unmittelbare Implementierung der Datenzugriffe im Anwendungsprogramm trägt zwar deutlich zur Erhöhung der Zugriffsgeschwindigkeit bei, ein traditioneller „join“ muss allerdings häufig mittels MapReduce-Techniken [DnGh04] aufwendig selbst programmiert werden [EFH+11]. Es stellt sich die Frage, ob es sich bei den NoSQL-Systemen um eine nachhaltige evolutionäre oder revolutionäre Entwicklung in der Datenbankwelt handelt oder ob sie

insbesondere einen Hype verkörpern. Ein wichtiges Kriterium für das Überleben einer Datenbanksystemtechnik ist vielfach der mit dem System erreichbare Grad an Performance.

In diesem Papier soll einerseits der Frage nachgegangen werden, ob NoSQL-Systeme zur Unterstützung von Cloud-Anwendungen tatsächlich deutlich schneller als traditionelle Datenbanksysteme sind. Andererseits soll analysiert werden, wie sich NoSQL-Systeme im Gegenzug bei der Unterstützung eher traditioneller administrativ-betriebswirtschaftlicher Anwendungen aus dem Bereich des Online Transaction Processings (OLTP) behaupten. Zur Klärung bieten sich Benchmarks im Rahmen einer Leistungsbewertungen an, wobei charakteristische OLTP- und Cloud-Arbeitslasten auf Basis sowohl relationaler als auch von NoSQL-Speichertechniken analysiert und gegenübergestellt werden (siehe Abbildung 1). Das Papier ist wie folgt gegliedert. Im nächsten Kapitel folgt eine kurze Einführung in das Thema der Leistungsbewertung im Cloud-Umfeld. Anschließend wird ein konkreter Benchmark ausgewählt, die Metrik bestimmt und die Lastparameter festgelegt. Danach folgt die Auswahl der gegenüberzustellenden relationalen und NoSQL-Speichertechniken sowie deren Einbindung in die Benchmark-Umgebung. Anschließend werden die Resultate der Messungen erläutert und hinsichtlich der festgestellten Performance analysiert. Das Papier schließt mit einer kurzen Zusammenfassung und Würdigung der wichtigsten Erkenntnisse und gibt einen Ausblick auf zukünftige Entwicklungen.

Leistungsbewertung im Cloud-Umfeld

Bevor mit einer aussagekräftigen Leistungsbewertung begonnen wird, gilt es eine Reihe von Entscheidungen zu treffen:

- **Anwendungsumfeld:**
Welche Anwendungsszenarien und welche Arbeitslast aus dem Umfeld der traditionell administrativ-betriebswirtschaftlichen und der Cloud-Anwendungen sollen im Rahmen der Leistungsbewertung berücksichtigt werden?
- **Systemumfeld:**
Welche Vertreter der relationalen und der NoSQL-Speichersysteme sollen auf Basis welcher Hardware und Betriebssysteme gegeneinander antreten?

- **Bechmarks und Metriken:**
Wie soll die Arbeitslast gemessen werden und über welche Kennzahlen soll die Leistung festgestellt und letztendlich verglichen werden?
- **Werkzeuge zur Leistungsbewertung:**
Welche Benchmark-Werkzeuge müssen entwickelt oder können genutzt bzw. angepasst werden, die sich einerseits zur Simulation beider Anwendungsumgebungen eignen und andererseits auch auf Grundlage unterschiedlicher Speichertechniken herangezogen werden können?

Cloud-Anwendungen

Das „National Institute of Standards and Technology“ (NIST) umschreibt „Cloud Computing“ als ein Modell, anhand dessen jederzeit ein ortsunabhängiger und transparenter Zugriff auf einen Pool im Netz verteilter konfigurierbaren Rechnerressourcen möglich ist (z.B. Netze, Server, Speichersysteme, Anwendungen und Dienste), die schnell und mit minimalem Verwaltungsaufwand oder geringer Serviceprovider-Interaktion zur Verfügung gestellt werden können [MIGr11].

Bei der Suche nach einer möglichst repräsentativen Cloud-Anwendung stößt man schnell auf unterschiedliche Service- und Bereitstellungsmodelle, auf eine Vielzahl von Anwendungsklassen sowie auf zahlreiche Nutzungsarten [MIGr11, StMn11]:

Servicemodelle

- **Software as a Service (SaaS):**
Endanwendungen in der Cloud.
- **Platform as a Service (PaaS):**
Integrierte Laufzeit- und Entwicklungsumgebungen in der Cloud.
- **Infrastructure as a Service (IaaS):**
Rechnerinfrastruktur in der Cloud.

Bereitstellungsmodelle

- **Public Cloud:**
Für die breite Öffentlichkeit.
- **Private Cloud:**
Innerhalb der eigenen Organisation.
- **Hybrid Cloud:**
Kombiniert Public und Private Clouds.
- **Community Cloud:**
Für einen kleineren Nutzerkreis.

Die unterschiedlichen Anwendungsklassen umfassen neben den ursprünglichen Cloud-Anwendungen wie z.B. den Online-Speichern mittlerweile Dienste aus fast allen IT-Bereichen:

Anwendungsklassen (Auswahl)

- Dateisynchronisation zwischen verschiedenen Rechnern (Online-Speicher)
- Web-basierte E-Mail-Dienste
- Web-basierte Office-Suiten
- Dienste zur Foto Speicherung
- Soziale Netzwerke
- Streaming-Dienste (Audio, Video)
- Online Gaming
- Online-Krankenakten Speicher
- Steuerberatung
- Buchhaltung und Lohnbuchhaltung
- E-Collaboration
- High Performance Computing

Ebenso vielfältig sind die Nutzungsarten, die neben den einfachen dynamischen Web-Seiten mittlerweile ebenfalls Arbeitsweisen fast aller Bereiche der IT umfassen:

Nutzungsarten (Auswahl)

- Datenbankgestützte Web-Anwendungen
- Kommunikation
- Backends für mobile Anwendungen
- Bandbreiten-, speicherintensive Nutzung
- Entwicklung, Test
- Online Transaction Processing (OLTP)
- Datenanalyse, Big Data

Bereits dieser kurze Blick in die Welt des Cloud-Computings verdeutlicht die sich hieraus ergebenden signifikant unterschiedlichen Lastprofile der potenziellen Anwendungsszenarien. Eine für die Leistungsbewertung repräsentative Cloud-Anwendung existiert daher praktisch nicht. Vielmehr soll im Folgenden auf eine für Cloud-Umgebungen charakteristische Anwendung zurückgegriffen werden. Die Wahl fällt auf ein einfaches Diskussionsforum. Derartige datenbankgestützte Anwendungen auf Basis dynamischer Web-Seiten gehören mit ihrer Nutzungsart der „threaded Conversations“ zur Anwenderklasse der sozialen Netzwerke und sind vielfach im Servicemodell SaaS im Rahmen fast aller Bereitstellungsmodelle anzutreffen.

Auswahl der NoSQL-Speichertechnik

Während die relationalen Systeme aus Sicht der Speichertechnik grundsätzlich vergleichbar sind, so besteht die Gemeinsamkeit der NoSQL-Systeme häufig lediglich in ihrer Schemalosigkeit. Sie lassen sich je nach verwendeter Speichertechnologie mehreren Kategorien zuordnen:

- **Key Value Systeme:**
Sie speichern einfache Werte, die über einen Schlüssel adressiert werden können.
- **Column Families:**
Hier kann ein Wert wiederum aus einer internen Menge von Schlüssel/Wert-Paaren bestehen, wodurch mehrdimensionale Strukturen entstehen. Gleichzeitig lassen sich ähnliche Eigenschaften unterschiedlicher Datensätze zusammengefasst (spaltenorientiert) speichern.
- **Document Stores:**
Sie erlauben die Speicherung hierarchisch strukturierter Datensätze (Dokumente).
- **Graph Databases:**
Mittels Beziehungen zwischen den Datensätzen ermöglichen sie die Speicherung semantisch ausdrucksstarker Datennetze.

Eine generelle Gegenüberstellung „NoSQL vs. Relational“ ist daher nur unter Berücksichtigung einer Vielzahl von NoSQL-Kategorien sinnvoll. Das geplante Diskussionsforum als Vertreter der Cloud-Anwendungen lässt sich durchaus auf Grundlage aller NoSQL-Kategorien realisieren. Aus Sicht ihrer geringen Strukturierungsmöglichkeiten unterscheiden sich die Key Value Systeme jedoch am deutlichsten von den relationalen Systemen. Eine Gegenüberstellung „Key Value vs. Relational“ verspricht somit den größten Interpretationsspielraum und soll daher im folgenden Performance-Vergleich berücksichtigt werden.

Benchmark und Metrik

Leistungsbewertungen basieren vielfach auf Benchmarks. Ein Benchmark (Maßstab) ist ein Test oder eine Reihe von Tests, die speziell dazu entwickelt wurden, die Leistung eines Computersystems gegen die Leistung eines anderen zu vergleichen [SPEa14]. Ein solcher Test basiert auf Metriken. Unter ei-

ner Software-Qualitätsmetrik versteht man wiederum eine Funktion, die eine Software-Einheit in einen Zahlenwert abbildet, welcher als Erfüllungsgrad einer Qualitätseigenschaft der Software-Einheit interpretierbar ist [IEEE98]. Zu den häufig verwendeten Metriken zählen u.a. [SPEb14]:

- Antwortzeiten (Latenz) zwischen Anfrage und Ergebnis
- Durchsatz in Arbeitseinheiten pro Zeiteinheit
- Variabilität, Wiederholbarkeit der Testergebnisse (Standardabweichungen)
- Dichte der Arbeitslast (Parallelität)
- Kosten pro Transaktion pro Zeiteinheit
- Energiebedarf
- Grad der Anpassbarkeit an Änderungen der Anwendungsklasse
- Dauerhaftigkeit von Änderungen bzw. Wahrscheinlichkeit von Datenverlusten
- Grad der Zuverlässigkeit der Ausführung unter gleich bleibenden Systembedingungen für einen gegebenen Zeitraum
- Grad der Verfügbarkeit

Das Interesse der einzelnen Metriken hängt stark von der jeweiligen Zielgruppe ab, z.B.:

- IaaS: Energiebedarf, Durchsatz, Dichte, Verfügbarkeit, ...
- Endanwender im Cloud-Umfeld: Antwortzeit, Kosten, Dauerhaftigkeit, ...

Anbieter von Benchmarks

Zu den traditionellen und namhaften Anbietern von Benchmarks zählen bereits seit vielen Jahren der „Transaction Processing Performance Council“ (TPC) und die „Standard Performance Evaluation Corporation“ (SPEC). Der TPC unterstützt u.a. Benchmarks im Umfeld von Application Server und Web Services. Sie stellt zudem ein Forum zur Veröffentlichung und Gegenüberstellung von Benchmarkergebnissen zur Verfügung. Allerdings basieren die angebotenen Benchmarks auf relativ fester und wenig skalierbarer Systemlast und setzen traditionelle Datenbanksysteme voraus [TPCa14]. SPEC hingegen verfügt über eigene Cloud-Untergruppen, die sich der Messung von Leistung und Skalierbarkeit von Cloud Computing-Services aber auch der Definition von Benchmark-Szenarien widmen [SPEC14]. Dennoch wird derzeit kein spezieller Cloud-Benchmark angeboten.

Darüber hinaus existieren mittlerweile auch Anbieter spezieller Cloud-Benchmarks. Hierunter zählen u.a. CloudStone, CloudCmp, malgen und YCSB [StMn11, SPEb14]. Besonders viel Aufmerksamkeit findet hierbei der „Yahoo! Cloud Serving Benchmark“ (YCSB) [CST+10], ein von Yahoo entwickeltes auf Java basierendes Framework zur Erstellung und Durchführung von standardisierten Benchmarks im Umfeld des Cloud-Computings [Cpr14]. Die Leistungsbewertung konzentriert sich hierbei insbesondere auf den Speicherdienst der Cloud-Architekturen. Als Arbeitslast dient ein synthetischer Mix aus Einfüge-, Änderungs-, Lese- und Scan-Operationen. Es stehen fertig konfigurierte Benchmarks zur Simulation einer Reihe von Cloud-Szenarien wie z.B. „User Status Updates“ oder das „Foto Tagging“ in sozialen Netzwerken zur Verfügung. Aufgrund der anwendungsspezifischen Anpassbarkeit sowie der Unterstützung sowohl traditioneller als auch NoSQL-Speichertechniken, soll die folgende Leistungsbewertung auf Grundlage des YCSB durchgeführt werden.

„Yahoo! Cloud Serving Benchmark“ (YCSB)

Der YCSB [CST+10] bietet einen zweistufigen Benchmark an: *Leistung* und *Skalierung*. Zwei weitere Stufen *Verfügbarkeit* und *Replikation* sind geplant. *Leistung* beinhaltet Metriken, mit denen sich traditionelle Messungen von Durchsatz vs. Belastung (Operationen pro Sekunde) bei kontinuierlich ansteigender Systemlast bestimmen lassen. Im Rahmen der *Skalierung* bietet der YCSB zwei weitere Metriken an: *Scaleup* und *Elastic Speedup*. *Scaleup* ermöglicht die proportionale Angleichung von Arbeitslast und Datenumfang an Veränderungen der Systemumgebung wie z.B. der horizontalen oder vertikalen Skalierung. Die Performance sollte hierbei konstant bleiben. *Elastic Speedup* ermöglicht zudem Messungen während Systemkapazitäten dynamisch während des Betriebs hinzugefügt werden. In diesem Fall ist zu erwarten, dass die Performance nach einer kurzen Reorganisationsphase ansteigen wird.

Die Architektur des YCSB-Client ist in Abbildung 2 dargestellt. Der *Workload-Executer* generiert die zu simulierende Arbeitslast anhand von zuvor spezifizierten Last- und Laufzeitparameterwerten. Mit Hilfe der *Lastparameter* lassen sich zahlreiche Anwendungsszenarien simulieren. Auch wenn die Datenbank insgesamt nur aus einer Tabelle besteht, so lässt sich die Größe über d Datensätze mit jeweils a Attributen zu je b Bytes an ASCII-Zeichen frei skalieren. Darüber hinaus existiert ein Primärschlüssel-Attribut. Die Operationen werden zufällig im Rahmen von parametrisierbaren Zugriffswahrscheinlichkeiten ausgewählt. Ein *Insert* fügt einen Datensatz ein, ein *Update* ändert ein zufälliges Attribut eines Datensatzes, ein *Read* liest einen Datensatz, ein zufälliges Attribut oder alle Attribute während ein *Scan* ausgehend von einem Datensatz eine zufällige Anzahl nachfolgender Datensätze scannt. Darüber hinaus bietet der YCSB eine Parametrisierung der Lastverteilung an. *Uniform* greift dabei mit gleicher Zugriffswahrscheinlichkeit auf die Datensätze zu. *Zipfian* ermöglicht eine Konzentration auf Hotspot-Datensätze, während *Latest* Zugriffe auf die jüngsten Datensätze bevorzugt.

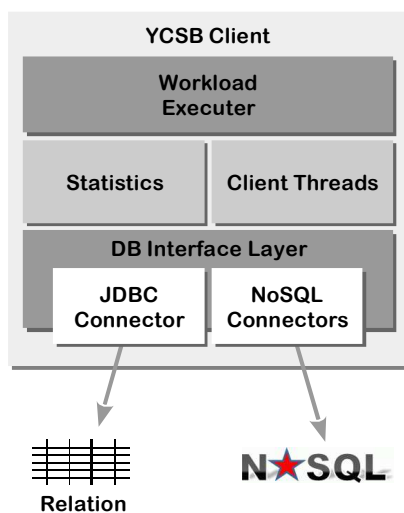


Abb. 2: Die Architektur des YCSB-Client [CST+10]

Darüber hinaus ermöglichen *Laufzeitparameter* die Simulation der zu den Anwendungsszenarien passenden Systemumgebungen. Mit ihnen kann u.a. die Anzahl der parallelen Zugriffe (*Client Threads*) skaliert, Grenzwerte für die Last (Operationen pro Sekunde) bestimmt aber auch Parameter der jeweils zugrunde liegenden Speichersysteme wie z.B. speziell verfügbare Konsistenzstufen angesprochen werden.



Die festgestellten Messwerte werden von der Komponente *Statistics* aufbereitet und zur weiteren Verarbeitung zur Verfügung gestellt. Die Kopplung des YCSB-Client an ein zu bewertendes Speichersystem geschieht im *DB Interface Layer* über einen *Connector*. Hierbei handelt es sich um spezielle Java-Klassen, welche die Arbeitslast auf die jeweils erforderliche Datenbanksprache wie z.B. SQL oder JSON des Zielsystems abbilden. Neben zahlreichen speziellen Konnektoren für die NoSQL-Systeme existiert auch ein allgemeiner JDBC-Konnektor zur Bewertung relationaler Systeme.

Spezifikation der Leistungsbewertung

Es folgt zunächst eine Festlegung der mit dem Benchmark verfolgten Ziele. Anschließend werden die zur Lastsimulation erforderlichen Lastparameter spezifiziert. Schließlich wird mit der Auswahl der konkreten Hard- und Software-Systeme die für den Benchmark vorgesehene Systemumgebung festgelegt.

Der Benchmark

Der folgende Benchmark soll insbesondere das Verhältnis von Antwortzeit und steigender Belastung feststellen. NoSQL-Systeme verdanken ihre hohe Popularität u.a. der horizontalen Skalierbarkeit auf mehrere Rechner. Traditionelle Systeme greifen bei Performanceengpässen häufig auf die Möglichkeiten der vertikalen Skalierung zurück, indem z.B. der Hauptspeicher vergrößert oder die Leistungsfähigkeit der CPU erhöht wird. Daher soll der Benchmark auch ergründen, in wie weit sich die durch Skalierung erzielbare Lastverteilung in beiden Szenarien tatsächlich positiv auf die Performance auswirkt. Für die Cloud-Umgebung wird ein horizontal skalierbarer Verbund mehrerer Rechner auf Basis eines NoSQL Key Value Systems als Testumgebung dienen. Die OLTP-Umgebung wird dagegen von einem einzelnen Rechner gebildet, dessen Leistungsfähigkeit in etwa der des konkurrierenden Clusters entsprechen soll, aber vertikal skalierbar ist. Diese Umgebung ermöglicht die Anwendung folgender Metriken, die im Benchmark hinsichtlich lesender, ändernder sowie einfügender Zugriffe berücksichtigt werden sollen:

Workload	OLTP 	Cloud 
Szenario	eher administrativ, betriebswirtschaftlich, OLTP	eher Cloud, Gästebuch, Diskussionsforum
Tabelle	1	1
Spalten	10 + 1 Primärschlüssel	1 + 1 Primärschlüssel
Datentyp	Jeweils 100 ASCII-Zeichen	1.000 ASCII-Zeichen
Datensätze	500.000	500.000
Operationen Art	Read: 80 % Update: 10 %, Insert: 10 %	Read: 80 % Insert: 20 %
Operationen Anzahl	100.000	100.000
Verteilung	Uniform	Latest

Speichertechnik

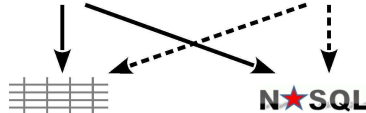


Abb. 3: Geplante Arbeitslast des Benchmark

Die Lastparameter

Die eher OLTP-typische Simulation basiert auf einer Tabelle mit 10 Spalten zu je 100 Zeichen. In Anlehnung an den TPC-E-Benchmark [TPCb14] soll hier ein 80/20-Verhältnis von lesenden (*read*) und schreibenden (*insert, update*) Operationen angenommen werden. Zudem wird von einer gleichmäßigen Verteilung der Operationen auf alle Datensätze ausgegangen (*Uniform*). Die für ein Cloud-Umfeld charakteristische Last soll am Beispiel des Diskussionsforums auf Basis eines Key-Value-Speichers simuliert werden. Die Tabelle enthält neben dem Schlüssel (*Key*) daher nur eine Spalte (*Value*) mit allerdings 1.000 Zeichen. Da bei Diskussionsforen im wesentlichen Einträge gelesen

- Antwortzeit vs. steigende Belastung (Operationen pro Sekunde) bei konstanter Systemumgebung
- Antwortzeit vs. horizontale Skalierung (Anzahl Rechner) bei konstanter Belastung
- Antwortzeit vs. vertikale Skalierung (CPU-Taktfrequenz, Arbeitsspeicher) bei konstanter Belastung

werden und vielfach nur begrenzt Möglichkeiten zur Korrektur der Einträge bestehen, wird von einem 80/20-Verhältnis von *Read*- und *Insert*-Operationen ausgegangen. Der Verzicht auf explizite *Update*-Anweisungen trägt erheblich zur Reduzierung potenzieller Konsistenzprobleme bei. Diese Sicht der Anwendung kommt dem NoSQL-Paradigma in replizierten und verteilten Umgebungen sehr entgegen. Hier wird der Fokus auf Verfügbarkeit und Partitionstoleranz gelegt, während Fragen zur Sicherung der Konsistenz in den Aufgabenbereich der Anwendung verlagert werden müssen (CAP-Theorem, [Brw00, EFH+11]). Zudem wird davon ausgegangen, dass überwiegend aktuelle Einträge gelesen werden, was der Verteilung (*Latest*) entspricht. In beiden Lastsimulationen sollen 500.000 Datensätze berücksichtigt werden. Somit ist die Größe der Datenbank mit ca. 500 MB für beide Systemumgebungen vergleichbar. Pro Lasttest sind jeweils 100.000 Operationen vorgesehen. Die Werte der jeweils verwendeten Lastparameter werden in Abbildung 3 zusammengefasst.

Die Systemumgebung und Anbindung des YCSB-Clients

Die Spezifikation der Systemumgebung umfasst neben der Auswahl von Hardware, Betriebssystem, NoSQL- und relationaler Speichertechnik auch die Festlegung der Laufzeitparameter (siehe Abbildung 4). Aus pragmatischen Gründen wird sich bei beiden Benchmarks mit CentOS 7 [Cnt14] für eine aktuelle Linux-Distribution entschieden. Spannender wird es dagegen mit dem aktuellen Oracle 12c [Orcb14], einen Marktführer als Vertreter der relationalen Speichertechnik zu analysieren. Für die Rolle des NoSQL Key Value-Systems bietet sich ebenfalls ein vergleichbar populärer Kandidat wie z.B. redis [Rds14] oder Riak [Bsh14] an. Seit einiger Zeit bietet auch Oracle mit der Oracle NoSQL Database ein eignes Key Value System an [Orcc14]. Das System basiert auf der BerkeleyDB [Orca14] und besitzt zusätzlich Mechanismen zur verteilten Datenhaltung auf mehreren Rechnern. Im folgenden Benchmark „Oracle 12c vs. Oracle NoSQL Database“ wird es somit zu einem „Geschwisterkampf“ in der Familie der Oracle-Datenbanksysteme kommen.

Die Laufzeitparameter ermöglichen die Steuerung der Systemlast. Die Systemlast erfolgt in Operationen pro Sekunde. Sie lässt sich für jede Messreihe nach oben begrenzen und so schrittweise von Messreihe zu Messreihe steigern.

Runtime	ORACLE DATABASE 12c	ORACLE NOSQL DATABASE
Version	Oracle12c	Oracle NoSQL Database
Server	1	4
Betriebssystem	CentOS 7	
CPU	Intel Dual Core E8400 3 GHz	je Intel Dual Core E8400 3 GHz
Hauptspeicher	1, 2, 4 GB	je 2 GB
Skalierung	vertikal 1, 2, 3 GHz CPU	horizontal 1, 2, 4 Rechner
parallele Threads	10	
Grenzwerte Operationen pro Sekunde	100, 200, 400, 800, 2.000, 5.000, 10.000	
Systemparameter	Standard JDBC	OLTP: Consistency=ABSOLUTE Durability=COMMIT_WRITE_NO_SYNC CLOUD: Consistency=NONE_REQUIRED durability=COMMIT_NO_SYNC

Bei allen Messungen soll die gesamte Last auf insgesamt zehn parallele Threads verteilt werden, was in etwa mit der Anzahl paralleler Anwender verglichen werden kann. Die horizontale Skalierung geschieht auf der Basis von bis zu 4 weniger leistungsfähigeren Rechnern, während die vertikale Skalierung mittels variablem Hauptspeicher und variabler Taktfrequenz der CPU erfolgen soll. Spezielle Systemparameter zur Optimierung der zugrunde liegenden Speichertechniken werden bei Oracle 12c nicht berücksichtigt. Bei Oracle

Abb. 4: Geplante Laufzeitparameter des Benchmark

NoSQL werden die Systemparameter *Consistency* und *Durability* genutzt. Hierbei sollen für das OLTP-Umfeld, bis auf die asynchrone Protokollierung von Änderungen, ACID-ähnliche Konsistenz-Kriterien gelten. Im Cloud-Umfeld werden diese Kriterien zu Gunsten der Laufzeiteigenschaften noch weiter eingeschränkt [Orca14].

Die Kopplung des YCSB-Clients mit den jeweiligen Speichertechniken erweist sich als sehr einfach, da die erforderlichen Konnektoren bereits in Form von Java-Klassen existieren [Cpr14]. Während für die Oracle NoSQL Database ein passgenauer Konnektor genutzt werden kann, ist bei Oracle 12c ein Umweg über den allgemeinen JDBC-Konnektor erforderlich.

Durchführung der Leistungsbewertung

Wie oben beschrieben sollen drei unterschiedliche Benchmarks durchgeführt werden.

Antwortzeit vs. steigende Belastung bei konstanter Systemumgebung

Die Messung der Antwortzeit bei konstanter Systemumgebung erfolgt in Abhängigkeit von der Last, die mit n Operationen pro Sekunde erzeugt wird, wobei $n \in \{100, 200, 400, 800, 2.000, 5.000, 10.000\}$. Pro Messung (n) werden jeweils 100.000 Operationen durchgeführt, deren einzelne Antwortzeiten zu einem Durchschnittswert verdichtet werden. Eine Messreihe setzt sich somit aus 7 Messungen zu jeweils 100.000 Operationen zusammen und bezieht sich auf eine der vorgesehenen Operationsarten *read*, *update* und *insert*, einem der beiden Anwendungsfälle sowie einem der beiden Systemumgebungen. Da im Cloud-Umfeld keine ändernden Operationen gemessen werden sollen, sind insgesamt 10 Messreihen auf der Basis von jeweils 700.000 Operationen vorgesehen. Im System Oracle 12c kommt ein Rechner mit einer 3 GHz CPU sowie 4 GB Hauptspeicher zum Einsatz. Das System Oracle NoSQL baut auf einem Cluster auf, dass aus vier gleichartigen Rechnern mit jeweils 3 GHz CPU sowie 2 GB Hauptspeicher besteht. Es ist zu erwarten, dass die Antwortzeiten mit steigendem n kontinuierlich zunehmen wird.

Antwortzeit vs. horizontale Skalierung bei konstanter Belastung

Die Messung der Antwortzeit bei der horizontalen Skalierung soll in Abhängigkeit einer konstanten Systemlast von 800 Operationen pro Sekunde und der variablen Anzahl r Rechner erfolgen, wobei $r \in \{1, 2, 4\}$. Die Rechner verfügen jeweils über eine CPU mit 3 GHz sowie über 2 GB Hauptspeicher. Die horizontale Skalierung wird nur in der NoSQL-Systemumgebung durchgeführt. Pro Messung (r) werden jeweils 100.000 Operationen durchgeführt, deren einzelne Antwortzeiten zu einem Durchschnittswert verdichtet werden. Eine Messreihe setzt sich somit aus 3 Messungen zu jeweils 100.000 Operationen zusammen und bezieht sich auf eine der vorgesehenen Operationsarten *read*, *update* und *insert* sowie einem der beiden Anwendungsfälle. Da im Cloud-Umfeld keine ändernden Operationen gemessen werden sollen, sind hier insgesamt 5 Messreihen auf der Basis von jeweils 300.000 Operationen vorgesehen. Es ist zu erwarten, dass die Antwortzeit mit steigendem r kontinuierlich abnehmen wird.

Antwortzeit vs. vertikale Skalierung bei konstanter Belastung

Die Messung der Antwortzeit bei der vertikalen Skalierung erfolgt in Abhängigkeit einer konstanten Systemlast von 800 Operationen pro Sekunde auf der Grundlage eines Rechners mit der variablen

Taktfrequenz t GHz der CPU bei h GB Hauptspeicher, wobei $t \in \{1, 2, 3\}$ und $h \in \{1, 2, 4\}$. Die vertikale Skalierung wird nur in der relationalen Systemumgebung durchgeführt. Pro Messung (t, h) werden jeweils 100.000 Operationen durchgeführt, deren einzelne Antwortzeiten zu einem Durchschnittswert verdichtet werden. Eine Messreihe setzt sich aus 3 Messungen (3 ausgewählte Kombinationen (t, h)) zusammen und bezieht sich auf eine der vorgesehenen Operationsarten *read*, *update* und *insert* sowie einem der beiden Anwendungsfälle. Da im Cloud-Umfeld keine ändernden Operationen gemessen werden sollen entstehen insgesamt 5 Messreihen auf der Basis von jeweils 300.000 Operationen. Es ist zu erwarten, dass die Antwortzeiten mit steigendem t bzw. h kontinuierlich abnehmen wird.

Auswertung der Leistungsbewertung

Die Auswertung hat zum Teil nachvollziehbare aber auch unerwartete Messwerte ergeben. Im Folgenden werden die Ergebnisse zusammengefasst, gegenübergestellt und interpretiert.

Antwortzeit vs. steigende Belastung bei konstanter Systemumgebung

Die Messergebnisse werden in den Abbildungen 5 bis 9 dargestellt. Bereits im ersten Versuch konnten die vorgesehenen Laufzeitparameter im System Oracle 12c nicht eingehalten werden. Die maximale Belastbarkeit lag hier bei ca. 2.200 Operationen pro Sekunde. Die darüber hinaus geplanten Systemlasten von 5.000 und 10.000 Operationen pro Sekunde ließen sich nur im System Oracle NoSQL erzielen.

In der Abbildung 5 werden zunächst die Lesezugriffe gegenübergestellt. Hier fällt sofort auf, dass die Zugriffszeiten bei steigender Systemlast offenbar nicht langsamer, sondern minimal schneller werden. Das ist insofern bemerkenswert, da bei jedem Messpunkt die gleiche Anzahl von Daten auf Basis eines neu aufgesetzten Systems gelesen werden. Die Lesezugriffe von Oracle 12c sind etwa doppelt so schnell wie die des konkurrierenden NoSQL-Systems. Auch dieses überrascht, da man den NoSQL-Systemen gerade bei lesenden Zugriffen extreme Schnelligkeit bescheinigt. Eine Erklärung könnte sein, das Oracle 12c über besonders leistungsfähige Zugriffspfadstrukturen verfügt. Vielleicht wirkt sich die horizontale Verteilung der Daten auf mehrere Rechner auch negativ aus, falls lesende Zugriffe überwiegend wenig selektiven Charakter haben?

Die lesenden Cloud-Zugriffe schneiden im Vergleich zu den lesenden OLTP-Zugriffen auf beiden Plattformen etwas besser ab. Das könnte daran liegen, dass sich die Cloud-Datensätze mit nur einer Spalte insgesamt flexibler verarbeiten lassen.

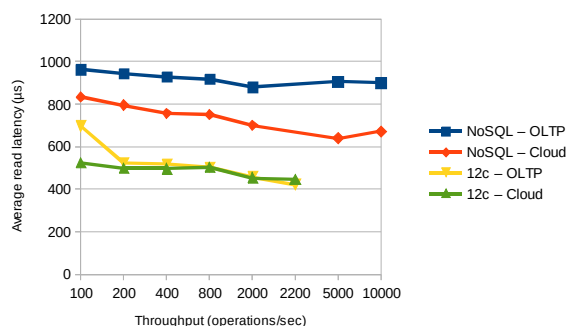


Abb. 5: Read, Antwortzeit vs. Belastung

In den Abbildungen 6 und 7 werden die ändernden Zugriffe gegenübergestellt. Offenbar führt eine höhere Systembelastung auch bei ändernden Zugriffen nicht zwangsläufig zu höheren Antwortzeiten. Dieses verdeutlicht das NoSQL-System erneut eindrucksvoll. Die Werte von Oracle 12c sind dagegen nachvollziehbar: mit steigender Systemlast steigen auch die Antwortzeiten deutlich.

Auffällig ist zudem, dass das NoSQL-System im Unterschied zu den lesenden Zugriffen bei den ändernden Zugriffen...

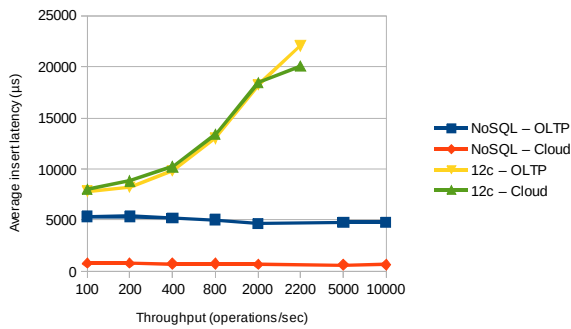


Abb. 6: Insert, Antwortzeit vs. Belastung

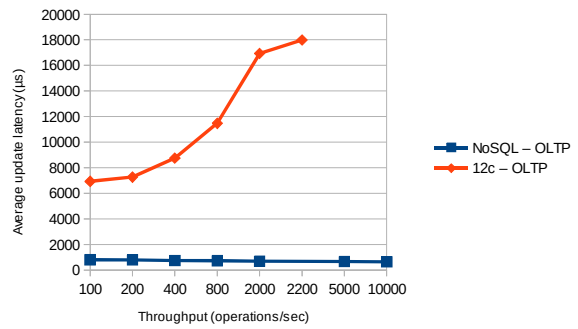


Abb. 7: Update, Antwortzeit vs. Belastung

dernden Zugriffen nun deutlich schneller ist, als Oracle 12c. Hier geht der Erklärungsversuch zum einen in die Richtung, dass die Änderungen im relationalen Umfeld synchron auf sekundäre Speichermedien protokolliert wurden. Das NoSQL-System protokolliert die Änderungen dagegen asynchron (Systemparameter *Durability*, siehe Abbildung 4). Hierdurch lassen sich zwar die Antwortzeiten verbessern, ein potentieller Datenverlust kann in dieser Betriebsart jedoch nicht immer ausgeschlossen werden. Eine weitere Erklärung könnte sein, dass das NoSQL-System die Änderungen lediglich sequentiell ablegt, während Oracle 12c bei jeder Änderung zudem eine Vielzahl interner Zugriffspfadstrukturen anpassen muss. Diese Sichtweise könnte auch erklären, dass Oracle im Gegenzug bei den Lesezugriffen schneller abgeschnitten hat (s.o.).

Bei den ändernden Zugriffen schneiden die Cloud-Zugriffe in beiden Systemen erneut besser als die OLTP-Zugriffe ab (s.o.). Im NoSQL-Umfeld sind sie sogar deutlich schneller. Letzteres könnte an der einfachen Key-Value-Struktur der Cloud-Daten liegen, die weitestgehend auf eine Interpretation der internen Value-Struktur verzichtet. Die für das NoSQL-System vorgesehenen OLTP-Daten verfügen dagegen im Value über zehn Attribute, die nicht vom Key-Value-System, sondern aufwendig im Rahmen der Anwendung interpretiert werden müssen.

In den Abbildungen 8 und 9 werden die Messergebnisse noch einmal systemspezifisch gegenübergestellt. Bei Oracle 12c wird erneut sichtbar, dass sich lesende Zugriffe bei steigender Last wider Erwarten leicht verbessern, während schreibende Zugriffe bei steigender Last erwartungsgemäß langsamer werden. Im NoSQL-System wird der Phänomen der Leistungssteigerung bei zunehmender Systemlast sogar bei allen Zugriffsarten festgestellt. Ähnliche Beobachtungen konnten auch bei vorangegangenen Leistungsbewertungen des NoSQL-Systems MongoDB [Mng14], einem Vertreter der Do-

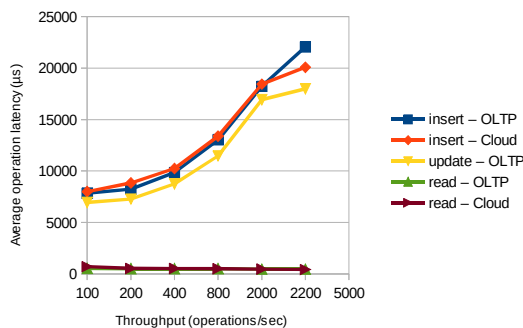


Abb. 8: Oracle 12c, Antwortzeit vs. Belastung

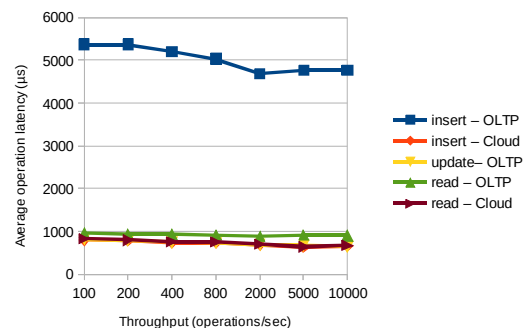


Abb. 9: Oracle NoSQL, Antwortzeit vs. Belastung

cument Stores, gemacht werden [KnKn14]. Erste Erklärungsversuche gingen dort in die Richtung, dass ein bedingt ausgelastetes System aufgrund des häufigeren Wechsels in Leerlaufzustände eine Belastung erfährt und somit weniger leistungstark arbeitet, als ein voll ausgelastetes System, dass weniger zwischen aktiven und passiven Zuständen hin und her wechseln muss.

Die Abbildung 8 zeigt den großen Unterschied zwischen lesenden und schreibenden Zugriffen auf dem relationalen System. Das ist nicht weiter verwunderlich, da Änderungen häufig auch die Anpassung von Zugriffspfaden und die Berücksichtigung von Constraints zur Folge haben. Diese deutliche Gruppierung von lesenden und schreibenden Zugriffen kann im NoSQL-System nicht beobachtet werden. Hier geht der Erklärungsversuch in die Richtung, dass die relationalen Systeme eine synchrone Protokollierung der Änderungen auf sekundäre Speichermedien betreiben, während die Protokollierung im NoSQL-System asynchron erfolgt (Systemparameter *Durability*, siehe Abbildung 4). Hierdurch lassen sich die Antwortzeiten der schreibenden Zugriffen im NoSQL-Umfeld deutlich reduzieren und entsprechen so in etwa dem Niveau lesender Zugriffen. Dieses geschieht allerdings auf Kosten von potenziellen Datenverlusten. Eine Ausnahmen bilden jedoch die einfügenden OLTP-Zugriffe im NoSQL-System.

In der Abbildung 9 wird gut sichtbar, dass einfügende OLTP-Zugriffe nicht nur im allgemeinen Vergleich zu den Cloud-Zugriffen (s.o.), sondern auch im speziellen NoSQL-System verhältnismäßig schwach abschneiden. Dieses lässt sich mit den etwas strengeren Konsistenzkriterien erklären, die speziell für die OLTP-Zugriffe im NoSQL-Umfeld festgelegt wurden (Systemparameter *Consistency* und *Durability*, siehe Abbildung 4) und u.U. ein häufigeres Protokollieren auf sekundäre Speichermedien zur Folge haben.

Antwortzeit vs. horizontale Skalierung bei konstanter Belastung

Die Messergebnisse werden in der Abbildungen 10 dargestellt. Um den Extremfall der horizontalen Skalierung mit einem aus nur einem Rechner bestehenden NoSQL-System messen zu können, war eine Reduzierung der geplanten Systemlast von 800 auf 120 Operationen pro Sekunde erforderlich. So sind die Werte zwar relativ vergleichbar, dennoch muss beachtet werden, dass nur 15% der Last vorgelegen hat. Zudem hat es bei den lesenden OLTP-Zugriffen einen extremen „Ausreißer“ gegeben.

Ansonsten zeigt sich im NoSQL-Umfeld ein eher nachvollziehbares Bild. Die Antwortzeiten gehen bei einem Verbund aus zwei Rechnern leicht hoch. Offenbar ist der mit zwei Rechnern verbundene interne Verwaltungsaufwand höher als die mit der Lastverteilung erreichbare Verbesserung der Ant-

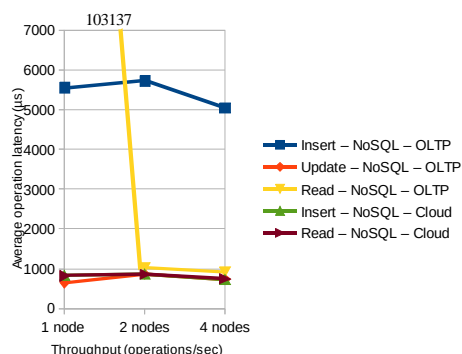


Abb. 10: Oracle NoSQL, horizontale Skalierung, Antwortzeit vs. Belastung

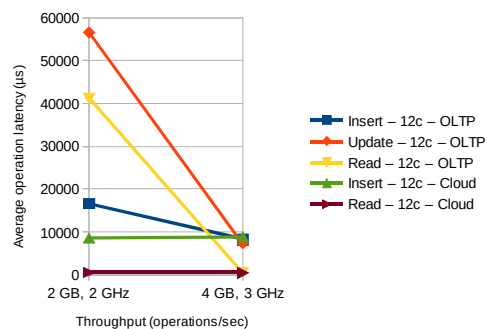


Abb. 11: Oracle 12c, vertikale Skalierung, Antwortzeit vs. Belastung

wortzeiten. Bei vier Rechnern wird zwar eine Verbesserung festgestellt, die jedoch bis auf die lesenden OLTP-Zugriffe nicht vollständig an die Werte des Einzelsystems herankommen. Hier ist zu vermuten, dass sich signifikante Leistungssteigerungen erst bei einer höheren Anzahl Rechner deutlich bemerkbar machen wird. Auffällig ist auch an dieser Stelle wieder, dass sich das NoSQL-System, unabhängig von der Anzahl Rechner, bei einfügenden OLTP-Zugriffen offenbar schwer tut (s.o.).

Antwortzeit vs. vertikale Skalierung bei konstanter Belastung

Die Messergebnisse werden in der Abbildung 11 dargestellt. Um die Auswirkungen der vertikalen Skalierung des Systems Oracle 12c bei einer Verminderung der Leistungsstärke auf nur noch 2 GB Hauptspeicher und 2 GHz Taktfrequenz stabil messen zu können, war auch hier eine Reduktion der geplanten Systemlast, diesmal von 800 auf 200 Operationen pro Sekunde, erforderlich. Auf die geplante Messung von 1 GB Hauptspeicher bei 1 GHz Taktfrequenz musste vollständig verzichtet werden. Gleichzeitig hat sich gezeigt, dass die Taktfrequenz der CPU keine signifikante Auswirkung auf die Messungen hatte. Ausschlaggebend war insbesondere die Größe des Arbeitsspeichers.

Die Ergebnisse sind insgesamt weniger spektakulär. Wie zu erwarten war, fallen die Antwortzeiten bei einer Erhöhung der Leistungsfähigkeit des Rechners deutlich. Auffällig ist, dass sich die ohnehin guten lesenden und schreibenden Cloud-Zugriffe beim leistungsstärkeren Rechner kaum verbessern. Die schlechten Antwortzeiten der lesenden OLTP-Zugriffe bei einem leistungsschwachen System werden mit großer Wahrscheinlichkeit vom zu geringen Hauptspeicher abhängen. Die lesenden Cloud-Zugriffe werden dagegen auch ohne großen Arbeitsspeicher zügig abgearbeitet. Letztendlich scheinen die Antwortzeiten bei leistungsstärkeren Rechnern insbesondere davon abzuhängen, ob es sich um Lese- oder Schreibzugriffe handelt. Dieser Zusammenhang wurde oben bereits erörtert (siehe Abbildung 8).

Zusammenfassung und Ausblick

Die Leistungsbewertung „Oracle 12c und Oracle NoSQL im Cloud Benchmark“ hat Ergebnisse hervorgebracht, die zum einen erwartet wurden, zum anderen aber nicht sofort nachvollziehbar sind. Während die Messungen im Rahmen der Skalierung keine überraschenden Werte ergeben haben, so konnte die Erwartungshaltung, dass bei höherer Systemlast die Leistungsfähigkeit der Systeme sinkt, nur teilweise bestätigt werden. Merkwürdigerweise glänzten beide Systeme mit leichten Verbesserungen der Antwortzeiten von Leseoperationen bei steigender Systemlast. Bei Oracle NoSQL traf diese Beobachtung sogar auf alle Zugriffsarten zu.

Als zweites Phänomen wurde wider Erwarten festgestellt, dass die lesenden Zugriffe beim relationalen Vertreter in beiden Anwendungsszenarien etwas schneller als beim NoSQL-System waren. Bei den schreibenden Zugriffen kehrt sich das Verhältnis um, wobei die Anwendungen auf dem NoSQL-System nun um ein Vielfaches schneller als im relationalen Umfeld abgeschnitten haben.

Trotz zahlreicher Erklärungsversuche steht ein nachhaltiger Beweis für die Ursachen der festgestellten Phänomene aus. Zusammenfassend kann bei der gegebenen Systemkonfiguration festgestellt werden:

- Das NoSQL-System hat sich insgesamt schneller als das relationale System verhalten. Außerdem wurde die Lastgrenze im relationalen Umfeld mit 2.200 Operationen pro Sekunde bereits wesentlich früher als beim NoSQL-System erreicht, wo bis zu 10.000 Operationen pro Sekunde möglich waren. Hier muss allerdings beachtet werden, dass die Last im NoSQL-Umfeld auf vier (aller-

dings schwächere) Rechner aufgeteilt wurde und der Geschwindigkeitsvorteil mit potenziellen Konsistenzeinbußen bezahlt werden muss.

- Die OLTP-Simulation war an einigen Stellen etwas langsamer als die Cloud-Simulation. Das kann an den erhöhten Konsistenzanforderungen und an der fehlenden Interpretierbarkeit der Attribute im Key-Value-System liegen. Im relationalen Umfeld hat sich die Unterscheidung „OLTP vs. Cloud“ jedoch kaum bemerkbar gemacht.

Insgesamt muss beachtet werden, dass beide Systeme standardmäßig installiert worden sind. Tuningmaßnahmen haben bis auf die physisch getrennte Verwaltung der Log-Dateien im relationalen System praktisch nicht statt gefunden. Darüber hinaus war die zur Verfügung stehende Hardware nur bedingt leistungsstark. In zukünftigen Arbeiten soll zunächst den oben diskutierten Hypothesen zu den festgestellten Phänomenen nachgegangen werden. Zu diesem Zweck bietet sich die Wiederholung der Benchmarks auf Basis weiterer NoSQL- und relationaler Systeme an. Die Frage, ob sich über höhere Lasten tatsächlich bessere Antwortzeiten erzielen lassen, lässt sich sicherlich über nach oben geöffnete horizontale und vertikale Skalierungen nachweisen.

Literatur

- [Brw00] Eric A. Brewer: *Towards Robust Distributed Systems (invited Talk)*. Proceedings of the ACM Symposium on Principles of Distributed Computing, Portland, Oregon, 2000
- [Bsh14] Basho Technologies: *Riak*. <http://basho.com/riak/>, 2014
- [Cnt14] CentOS: *CentOS*. <http://www.centos.org/>, 2014
- [Cpr14] Brian F. Cooper: *Yahoo! Cloud Serving Benchmark (YCSB)*. <https://github.com/brianfrankcooper/YCSB/wiki>, 2014
- [CST+10] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears: *Benchmarking cloud serving systems with YCSB*. Proceedings of the 1st ACM symposium on Cloud computing, Indianapolis, Indiana, USA, 2010
- [DnGh04] Jeffrey Dean, Sanjay Ghemawat: *MapReduce: Simplified Data Processing on Large Clusters*. Proceedings of the Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, 2004.
- [Edl14] Stefan Edlich: *NoSQL - Your Ultimate Guide to the Non - Relational Universe!*. <http://nosql-database.org/>, 2014
- [EFH+11] Stefan Edlich, Achim Friedland, Jens Hampe, Benjamin Brauer, Markus Brückner: *NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken*. Carl Hanser Verlag, 2011
- [IEEE98] Institute of Electrical and Electronics Engineers (IEEE): *IEEE Std 1061-1998. IEEE Standard for a Software Quality Metrics Methodology*. IEEE, 1998
- [KnKn14] Jascha Knack, Harm Knolle: *Yahoo! Cloud Serving Benchmark - MS SQL Server vs. MongoDB - . SQLSaturday #313 Rheinland*, <https://www.sqlsaturday.com/313/schedule.aspx>, 2014
- [MIGr11] Peter Mell, Timothy Grance: *The NIST Definition of Cloud Computing. (800-145)*. National Institute of Standards and Technology (NIST), Gaithersburg, MD, 2011
- [Mng14] MongoDB inc.: *MongoDB*. <http://www.mongodb.org/>, 2014
- [Orca14] Oracle Corporation: *Oracle Berkeley DB 12c*. <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html>, 2014
- [Orcb14] Oracle Corporation: *Oracle Database 12c*. <http://www.oracle.com/us/corporate/features/database-12c/index.html>, 2014
- [Orcc14] Oracle Corporation: *Oracle NoSQL Database*. <http://www.oracle.com/us/products/database/nosql/overview/index.html>, 2014

- [Rds14] redis: redis. <http://redis.io/>, 2014
- [SPEa14] Standard Performance Evaluation Corporation (SPEC): *Benchmark*.
<http://www.spec.org/spec/glossary/#benchmark>, 2014
- [SPEb14] Standard Performance Evaluation Corporation (SPEC), Open Systems Group (OSG), Cloud Working Group: *Report on Cloud Computing to the OSG Steering Committee*.
<http://www.spec.org/osgcloud/docs/osgcloudwgreport20120410.pdf>, 2012
- [SPEc14] Standard Performance Evaluation Corporation (SPEC), Open Systems Group (OSG): *OSG Cloud Subcommittee*. <http://www.spec.org/osgcloud>, 2014
- [StMn11] Dinkar Sitaram, Geetha Manjunath: *Moving To The Cloud: Developing Apps in the New World of Cloud Computing*. Syngress; 2011
- [TPCa14] Transaction Processing Performance Council (TPC): *Home*. <http://www.tpc.org>, 2014
- [TPCb14] Transaction Processing Performance Council (TPC): *TPC Benchmark™ E (TPC-E)*.
<http://www.tpc.org/tpce/>, 2014

Kontaktadresse:

Jascha Knack, Prof. Dr. Harm Knolle
Hochschule Bonn Rhein Sieg
Fachbereich Informatik
Grantham-Allee 20
53757 Sankt Augustin

Telefon: +49 (0) 2241 865 201

Fax: +49 (0) 2241 865 8253

E-Mail jascha.knack@smail.inf.h-brs.de, harm.knolle@h-brs.de

Internet: <http://www.inf.fh-bonn-rhein-sieg.de/Startseite.html>