

Aufgabenstellung

Im Rahmen des Praktikums „Speicher- und Datennetze im IoT“ (SDN) erstellen Sie eine Technologiestudie mit einer Demonstration und evaluieren Use Cases einer Technologie. Sie erstellen und untersuchen dabei verschiedene Fragestellungen, die Sie anhand von Spezifikationen, Dokumentationen und praktischer Erprobung beantworten.

Ablauf des Praktikums

In der Praktikumsgruppe bilden Sie jeweils Gruppen aus zwei Personen. Jede Gruppe bearbeitet dabei jeweils eine andere Kombination aus gewählter Technologie und Schwerpunkt und hiervon abgeleiteten Fragestellungen. Sie wählen dabei Ihre jeweilige Kombination zu Beginn. Sie können aus einer Reihe von grob vorgegebenen Fragestellungen, Evaluationskriterien und Use Cases wählen. Alternativ können Sie auch eine eigene Zielsetzung in Absprache mit Ihrem Betreuer erarbeiten, die eine in der Vorlesung vorgestellte Technologie evaluiert. Im weiteren Verlauf untersuchen Sie anhand dieser Zielsetzung die jeweilige Technologie und entwickeln eine Demonstration. Die Ergebnisse stellen Sie in Form einer Präsentation mit Online-Demo im Rahmen des Praktikums den anderen Studierenden vor.

Inhalt der Praktikumstermine

Für die Zeiten der Praktikumstermine gilt eine Anwesenheitspflicht. Die Bearbeitung erfolgt sowohl während der Praktikumstermine als auch im Rahmen Ihrer Vor- und Nachbereitung.

Das Praktikum gliedert sich in folgende Termine:

- | | |
|--------------|--|
| 1. Termin | Auswahl einer Technologie und Schwerpunkt der Evaluation |
| 2. Termin | Verfeinerung der Zielsetzung an Hand erster Prototypen
Durchsprache von Zwischenergebnissen |
| 3. Termin | Durchsprache der zu präsentierenden Ergebnisse |
| 4.+5. Termin | Vorträge mit Online-Demo der Praktikumsgruppen
Dauer ca. 10 Minuten pro Gruppe mit anschließendem Frageteil
von max. 10 Minuten. |

Ablauf während der Praktikumstermine und Bearbeitung

Die Bearbeitung der festgelegten Zielstellung erfolgt eigenständig. Die aufgeführten Quellen sollen Ihnen einen ersten Einstieg geben. Weitere Quellen suchen Sie selbstständig.

Machen Sie sich **vor dem 1. Praktikumstermin** mit den möglichen Schwerpunkten vertraut und ermitteln Sie, ob sie bereits für Sie nützliche Vorkenntnisse der hierzu eingesetzten Programmiersprachen, Technologien und Frameworks haben.

Im **1. Termin wählen Sie Ihre Technologie und Ihren Schwerpunkt** und sprechen mögliche Verfeinerungen der Zielsetzung durch. Sie finden hierzu jeweils drei Schwerpunkte für die drei Technologien CoAP, MQTT und GCS im weiteren Text. Jeder Schwerpunkt (1...9) soll in Ihrer Praktikumsgruppe nur durch eine 2er-Gruppe bearbeitet werden. Sie wählen während des Termins Ihren Schwerpunkt in Abstimmung mit der Gruppe aus. Alternativ können Sie sich selbst einen Schwerpunkt zu einer Technologie definieren, die in der Vorlesung vorgestellt wird, für die aber kein Schwerpunkt ausformuliert ist (z.B. für Amazon Simple Storage Service S3 oder eine Blockchain-Technologie).

Formulieren Sie Ihren Vorschlag Ihrer verfeinerten Zielsetzung in Ihren eigenen Worten bis zum **Montag vor (!) Ihrem 2. Termin** in Moodle ab. Spezifizieren Sie Komponenten (Hardware, Software) und die exakten Evaluationsziele. Sie finden Beispiele für eine Verfeinerung im weiteren Text für Schwerpunkte zu NFS als Technologie.

Im **2. Termin**. sprechen Sie Ihre verfeinerte Zielsetzung mit den geplanten Evaluationsergebnissen und Ihre Ansätze zur Erstellung von Prototypen durch. Weiterhin erfolgt die Aufteilung der Gruppe in Präsentationstermine für den 4. und 5. Praktikumstermin.

Bis zum **3. Termin** sollten Sie in der Erstellung eines Demos soweit sein, dass Sie etwaige Probleme in der Aufgabenstellung oder technische Hürden absehbar überwunden haben.

Im **3. Termin** demonstrieren Sie ihren Betreuern mittels einer Live-Demo ihren Bearbeitungsstand. Gegebenfalls können Sie Modifikationen Ihrer Zielsetzung durchsprechen. Weiterhin sollten nach dem Termin alle Fragen zu den zu erzielenden Evaluationsergebnissen geklärt sein.

Im **4. Termin** erfolgt die Vorstellung der Ergebnisse der ersten Hälfte der Praktikumsgruppe.

Im **5. Termin** erfolgt die Vorstellung der Ergebnisse der zweiten Hälfte der Praktikumsgruppe.

Bis zu dem **Montag vor (!) dem 5. Termin** geben alle Teilnehmer Ihre Dokumentation ab. Die Präsentierenden im 4. Termin geben die Dokumentation inklusive der präsentierten Folien nach Ihrem Vortrag ab. Die Abgabe der Dokumentation durch die Präsentierenden des 5. Termins vor (!) Ihrem Vortrag.

Abgabetermine in Moodle

Termine zur Abgabe Ihrer verfeinerten Zielsetzung nach dem 1. Termin:

X-Gruppe 30.04.2018

Y-Gruppe: 23.04.2018

Termine zur Abgabe der Dokumentation:

X-Gruppe: 11.06.2018

Y-Gruppe 18.06.2018

Die Testierung der **erfolgreichen Teilnahme** erfolgt für alle Teilnehmer im Anschluß an den 5. Termin. Die Anwesenheit ist an allen Terminen erforderlich.

Dokumentation

Die Dokumentation erfolgt in Form einer

- Präsentation
- einer schriftlichen Dokumentation
- durch die erstellten Quelltexte
- der Anleitung zur Reproduktion des Online-Demos
- Evaluation der Einarbeitung

Zur Evaluation der Einarbeitung beantworten Sie im Rahmen Ihrer Dokumentation folgende Fragen:

Wie lange benötigten Sie (geschätzt in Stunden)...

- sich in die jeweilige Dokumentation einzuarbeiten?
- bis Sie ein erstes Beispiel zum Laufen bekamen?
- für die Erstellung Ihrer Demo?
- für die Erstellung Ihrer Dokumentation?

Hilfsmittel

Die Bearbeitung kann auf Labor-PCs, eigenen Geräten und mittels kostenfreien Cloud-Services von Anbietern erfolgen, die Ihnen als Studierende als rechtliche Person zur Verfügung stehen. Wenn zur Erstellung des Online-Demos unterschiedliche Softwareimplementierungen zur Verfügung stehen, ist die Auswahl der jeweiligen Implementierung nach Kriterien selbst Teil der Evaluation. Im EST Labor stehen Raspberry 3 zur Verfügung, die gegen eine Kautionsentlohnung entliehen werden können. Wenn Sie einen Raspberry verwenden wollen, führen Sie dies als Teil Ihrer Zielsetzung mit auf und klären die Verfügbarkeit im Rahmen des 1. oder 2. Praktikumstermins.

Die Betreuung während des Praktikums konzentriert sich auf Festlegung und Klärung der Zielsetzung und gegebenenfalls Absprache zur Änderungen von Zielsetzungen. Schwerpunkte der Darstellung der Ergebnisse in Präsentationsform, Evaluationsansätze und -kriterien werden diskutiert.

Testat

Zu jedem Termin wird der Fortschritt im Ablauf überprüft. Ein deutliches Abweichen von dem erwarteten Stand führt zur „gelben Karte“, bei fortgesetzter Abweichung im Folgetermin zum Ausschluss vom Praktikum. Die Abschlusspräsentation muss die erarbeiteten Fragestellungen beantworten. Eine lauffähige Online-Demo zur Präsentation im 4. oder 5. Termin muss existieren. Alle Arbeitsergebnisse müssen in nachvollziehbarer und gut dokumentierter Form in Moodle vorliegen.

Labornutzung

Das EST-Labor kann während der Praktikumszeiten und in freien Zeiten während der üblichen Arbeitszeiten nach Vorgaben der Laborordnung genutzt werden. In freien Zeiten ist es erforderlich, dass Sie mindestens zu zweit anwesend sind. Beachten Sie die Laborordnung und halten Sie diese ein.

Schwerpunkt 1: Vergleich von HTTP und CoAP

Sie wählen einen üblichen HTTP-Server aus und vergleichen diesen mit den Eigenschaften von CoAP. Hierzu implementieren Sie einen CoAP-Server und mehrere – Clients und evaluieren mit Testdaten unterschiedlicher Größenordnung für Requests (von 0, 1, 2 Bytes über die 10-er Potenzen 10^1 bis 10^7). Verwenden Sie zum Netzwerkmonitoring Tools wie Wireshark.

1. Wie unterscheidet sich der Protokoll-Overhead zwischen den beiden Protokollen in Abhängigkeit von der Größe der Testdaten? Vergleichen Sie auf Netzwerkebene (on the wire) inklusive aller Bytes, die durch die unteren Netzwerkschichten hinzugefügt werden. Vergleichen Sie lesende und schreibende Zugriffe.
2. Messen Sie die Antwortzeiten für Anfragen über die beiden Protokolle. Wie unterscheiden sich die Antwortzeiten zwischen den beiden Protokollen? Wie verändern sich die Antwortzeiten in Abhängigkeit der übertragenen Datenmengen?
3. Messen Sie den Bedarf an Arbeitsspeicher der Server der beiden Protokolle. Wie verhält sich der Bedarf an Arbeitsspeicher bei größeren Datenmengen, die übertragen werden? Wie sieht der Bedarf auf Serverseite aus, wenn viele Clients gleichzeitig Daten abrufen bzw. ein Client viele Anfragen stellt?

Schwerpunkt 2: Ressourcenbedarf der Implementierung von CoAP-Komponenten

Sie wählen mindestens zwei Implementierungen in unterschiedlichen Programmiersprachen jeweils für CoAP-Client und -Server aus. Sie erstellen hieraus zwei Client-/Server-Implementierungen, die Anfragen dauerhaft in einem konstanten Zeitintervall stellen und durchführen z.B. den Abruf oder das Senden von 1 KB an Testdaten. Untersuchen Sie die beiden Implementierungen mit der gleichen Zielhardware.

1. Untersuchen Sie den entstehenden Hauptspeicher- und CPU-Bedarf für Client und Server in Abhängigkeit von der Laufzeit über einen längeren Zeitraum, über den Sie in gleichmäßigen Abständen (z.B. alle 100 Millisekunden) von mehreren Clients Daten senden bzw. empfangen.
2. Variieren Sie die Anzahl der Clients. Wie verhält sich der Hauptspeicher- und CPU-Bedarf in Abhängigkeit von der Anzahl?
3. Variieren Sie den zeitlichen Abstand, in denen die Clients die Daten abrufen. Wie verhält sich Hauptspeicher- und CPU-Bedarf? Können Sie eine Überlastung des Servers erreichen? Wann erreichen Sie diese?

Schwerpunkt 3: QoS-Eigenschaften von CoAP

Sie wählen eine Implementierung für CoAP-Clients und -Server aus. Sie erstellen hieraus eine Client-/Server-Implementierung, die dauerhaft Anfragen mit Testdaten stellen und durchführen.

1. Wählen Sie für die Anfragen die QoS-Modi Non-Confirmed und Confirmed. Untersuchen Sie die Antwortzeiten der Implementierung. Variieren Sie den Umfang der Testdaten für unterschiedliche Größenordnungen für Requests (von 0, 1,2 Bytes über die 10-er Potenzen 10^1 bis 10^7). Wie verändert sich das Zeitverhalten im Vergleich zwischen den beiden QoS-Modi?
2. Untersuchen Sie mit den unterschiedlichen Requestgrößen den Protokolloverhead auf Netzwerkebene für die beiden QoS-Modi im Vergleich zu den jeweils übertragenen Nutzdaten. Wie ändern sich die Verhältnisse in Abhängigkeit zu dem Umfang der Testdaten?
3. Limitieren Sie den Netzwerkdurchsatz künstlich durch einen Traffic-Shaper wie tc bzw. Wondershaper. Verwenden Sie Testdaten konstanter Größe (z.B. 1 KB) und variieren Sie den zeitlichen Abstand der Abrufe. Wie stellt sich das Zeitverhalten im Vergleich zwischen den beiden QoS-Modi dar? Können Sie eine Überlastung des Netzwerks simulieren? Wann erreichen Sie diese?

Schwerpunkt 4: Ressourcenbedarf der Implementierung von MQTT-Clients

Wählen Sie neben einem MQTT-Broker (wie Mosquitto) mindestens zwei Implementierungen für MQTT-Clients aus. Sie erstellen hieraus zwei Implementierungen, die jeweils eine Anzahl gleicher Clients als Publisher und Subscriber enthalten. Publisher senden dauerhaft in einem konstanten Zeitintervall neue Nachrichten, die durch Subscriber empfangen werden. Untersuchen Sie die beiden Implementierungen jeweils auf der gleichen Zielhardware.

1. Untersuchen Sie den entstehenden Hauptspeicher- und CPU-Bedarf für Publisher und Subscriber in Abhängigkeit von der Größe der gesendeten Testdaten (von 0, 1,2 Bytes über die 10-er Potenzen 10^1 bis 10^7) über einen längeren Zeitraum. Ein Publisher veröffentlicht dabei in gleichmäßigen Abständen Daten (z.B. alle 100 Millisekunden). Vergleichen Sie die Ergebnisse zwischen den beiden Implementierungen.
2. Untersuchen Sie den entstehenden Hauptspeicher- und CPU-Bedarf für Publisher und Subscriber in Abhängigkeit von dem zeitlichen Abstand von veröffentlichten Daten (von 0, 1,2 Bytes über die 10-er Potenzen 10^1 bis 10^7) über einen längeren Zeitraum. Vergleichen Sie die Ergebnisse zwischen den beiden Implementierungen.
3. Untersuchen Sie den entstehenden Hauptspeicher- und CPU-Bedarf für Publisher und Subscriber bei langen Laufzeiten für Testdaten gleicher Größe. Vergleichen Sie das Verhalten zwischen den beiden Implementierungen.

Schwerpunkt 5: QoS-Eigenschaften von MQTT

Sie wählen eine Implementierung für MQTT-Clients und -Broker aus. Sie erstellen hieraus eine Implementierung, in der ein Publisher dauerhaft Testdaten veröffentlicht. Lassen Sie einen Subscriber dann als Publisher agieren, der die Testdaten wiederum veröffentlicht. Der ursprüngliche erste Publisher subscribed sich hierauf. So ermitteln Sie für einen vollen Roundtrip eine Latenzzeit auf Gerät des ersten Publishers.

1. Wählen Sie für die Roundtrips die drei unterschiedlichen QoS-Modi aus. Untersuchen Sie die Latenzzeiten der Implementierung. Variieren Sie den Umfang der Testdaten für unterschiedliche Größenordnungen für Requests (von 0, 1,2 Bytes über die 10-er Potenzen 10^1 bis 10^7). Wie verändert sich die Latenzzeit im Vergleich zwischen den QoS-Modi?
2. Untersuchen Sie mit den unterschiedlichen Requestgrößen den Protokolloverhead auf Netzwerkebene für die QoS-Modi im Vergleich zu den jeweils übertragenen Nutzdaten. Wie ändern sich die Verhältnisse in Abhängigkeit zu dem Umfang der Testdaten?
3. Limitieren Sie den Netzwerkdurchsatz für den Broker künstlich durch einen Traffic-Shaper wie tc. Verwenden Sie Testdaten konstanter Größe (z.B. 1 KB) und variieren Sie den zeitlichen Abstand der Veröffentlichungen (Anzahl pro Sekunde). Wie stellt sich die Latenzzeit im Vergleich zwischen den QoS-Modi dar? Können Sie eine Überlastung des Netzwerks simulieren?
4. Erzeugen Sie künstliche Paketverluste mittels tc. Verwenden Sie Testdaten konstanter Größe (z.B. 1 KB) und veröffentlichen diese mit konstantem zeitlichen Abstand (Anzahl pro Sekunde). Variieren Sie die Verlustrate. Wie stellt sich die Gesamtverlustrate für Roundtrips nach längerer Zeit im Vergleich zwischen den QoS-Modi in Abhängigkeit von der Paketverlustrate dar?

Schwerpunkt 6: Vergleich von MQTT mit einer weiteren nachrichtenorientierten Middleware

Sie wählen eine frei verfügbare nachrichtenorientierte Middleware (message oriented Middleware MOM) aus, welche nach dem Publish-Subscribe-Muster mit Topic-Konzept Nachrichten austauschen kann. Für die Programmiersprache Java können Sie z.B. Implementierungen des Java Message Service (JMS) wählen, die einen JMS-Provider zur Verfügung stellen. Erstellen Sie jeweils einen Prototypen auf Basis von MQTT und auf Basis einer weiteren MOM-Implementierung mit mindestens einem Publisher und einem Subscriber. Evaluieren Sie die beiden Prototypen mit Testdaten unterschiedlicher Größenordnung für Requests (von 0, 1,2 Bytes über die 10-er Potenzen 10^1 bis 10^7). Verwenden Sie zum Netzwerkmonitoring Tools wie Wireshark.

4. Wie unterscheidet sich der Protokoll-Overhead zwischen den beiden Protokollen in Abhängigkeit von der Größe der Testdaten? Vergleichen Sie auf Netzwerkebene (on the wire) inklusive aller Bytes, die durch die unteren Netzwerkschichten hinzugefügt werden.
5. Erstellen Sie einen Roundtrip, indem der Publisher eines ersten Topics gleichzeitig als Subscriber für ein zweites Topic agiert. Der Subscriber agiert beim Erhalt von Nachrichten zu dem ersten Topic gleichzeitig als Publisher für das zweite Topic. So können Sie die Laufzeit des Versendens zweiter Nachrichten messen. Messen Sie die Antwortzeiten für Anfragen über die beiden Protokolle. Wie unterscheiden sich die Antwortzeiten zwischen den beiden Protokollen? Wie verändern sich die Antwortzeiten in Abhängigkeit der übertragenen Datenmengen?

Schwerpunkt 7: Lasttest von Google Cloud Storage

Cloud Storage Services wie Google Cloud Storage GCS sollten genügend Bandbreite zur Verfügung stellen, so dass für Zugriffe durch Applikationen kein Engpass entsteht. Die limitierende Grenze sollte durch die lokale Hardware oder die Netzanbindung gegeben sein. Hierzu erstellen Sie einen Client auf Basis der GCS API (XML oder JSON), welcher in der Lage ist, möglichst viele Schreibrequests gleichzeitig durchzuführen. Da Caching-Effekte bei Leserequests auftreten können, wird

1. Schreiben Sie Objekte mit der Größe 0 Byte, so dass Sie als Daten nur diejenigen für das Protokoll zur GCS ohne Nutzdaten senden. Schreiben Sie die Objekte sequentiell und messen die Latenzzeit, mit der Ihre Zugriffe durchgeführt werden. Messen Sie über eine längere Zeit und ermitteln Sie, welche Schwankungen die Latenzzeit haben kann.
2. Führen Sie Schreibzugriffe wie unter 1. durch, wobei diese parallel gestartet werden. Erhöhen Sie die Anzahl der parallelen Zugriffe in 2er-Potenzen, bis Sie an die Grenze der Parallelität Ihrer Applikation und der zugrunde liegenden Hardware kommen. Versuchen Sie die limitierenden Faktoren zu ermitteln.
3. Führen Sie parallele Schreibzugriffe wie unter 2. durch. Ermitteln Sie die Latenzzeiten über eine längere Zeit und variieren Sie wieder jeweils die Anzahl paralleler Zugriffe in 2er-Potenzen. Beobachten Sie, ob die Anzahl der gleichzeitigen Zugriffe einen Einfluß auf die Latenzzeit hat.
4. Wiederholen Sie die Messungen mit Objekten der Größe 1 Byte, 1000 Byte und der maximal übertragbaren Paketgröße und vergleichen Sie die Einflüsse der Übertragung von Nutzdaten auf die Latenzzeiten.

Hinweise zum Umgang mit großen Request-Raten bei GCS

<https://cloud.google.com/storage/docs/request-rate>

Schwerpunkt 8: Zugriffskontrollmöglichkeiten von Google Cloud Storage

Cloud Storage Services wie Google Cloud Storage GCS verfügen über unterschiedliche Möglichkeiten der Zugriffskontrolle auf Objekte (Identitäts- und Zugriffsmanagement, Zugriffskontrolllisten, signierte URLs, signierte Richtlinien dokumente). Machen Sie sich mit den verschiedenen Methoden vertraut und ermitteln Sie typische Anwendungsfälle für die jeweilige Methode.

1. Implementieren Sie einen Client, der unterschiedliche Identitäten annehmen kann.
2. Implementieren Sie jeweils mindestens einen Anwendungsfall für jeden Kontrollmethode, mit dem Sie die jeweilige Zugriffskontrollmöglichkeit demonstrieren können.

Hinweise zu Zugriffsskontrollmethoden für GCS

<https://cloud.google.com/storage/docs/access-control/>

Schwerpunkt 9: Multiregionale Datenspeicherung von Google Cloud Storage

Cloud Storage Services wie Google Cloud Storage GCS verfügen über mehrere, weltweit verteilte Instanzen bzw. Rechenzentren, in denen Daten gespeichert werden können. So können Daten möglichst in der Nähe der Verarbeitung gehalten werden. Weiterhin können Daten gegen einen Ausfall einzelner Rechenzentren geschützt werden. Implementieren Sie einen Client, mit dem Sie den Ort der Speicherung auswählen können und so Buckets in unterschiedlichen Regionen erstellen können.

1. Schreiben Sie sequentiell Objekte in drei unterschiedliche Regionen (USA, Europa, Asien). Untersuchen Sie dabei die Latenzzeit über einen längeren Zeitraum. Verwenden Sie Objekte mit einer Größe von 0 Byte, so dass Sie nur die Zeit zur Verarbeitung des Zugriffs messen.
2. Verwenden Sie nun Objekte mit Nutzdaten von 1 Byte und in 2er-Potenzen aufsteigend. Vergleichen Sie die Latenzzeiten bei aufsteigender Größe der Nutzdaten über einen längeren Zeitraum.
3. Kopieren Sie nun Objekte zwischen den Buckets in den drei unterschiedlichen Regionen. Messen Sie die Übertragungsdauern und vergleichen Sie diese jeweils in beide Richtungen.

Schwerpunkt X: Setzen Sie sich einen eigenen Evaluationsschwerpunkt

Wenn Sie einen eigenen Schwerpunkt definieren bearbeiten wollen, der nicht in der Liste oben enthalten ist, können Sie aus folgenden Listen an Fragen nach Eigenschaften und Zugriffsmöglichkeiten entsprechend passende Fragen aussuchen und hieraus Ihren eigenen Schwerpunkt für Fragestellungen ermitteln. Sie sollten hierzu mit der von Ihnen gewählten Technologie möglichst bereits vertraut sein.

Eigenschaften: Beispiele für Fragen und Aufgabenstellungen

- Welche Software- und Hardware-Eigenschaften gibt es?
- Beschreiben Sie die Eigenschaften des Systems.
- Welchen Einfluss hat die Hardware auf Softwareeigenschaften?
- Welche Eigenschaften lassen sich verändern und wie?
- Welche Rollen gibt es?
- Welche Eigenschaften sind für welche Rollen relevant?
- Welche Berechtigungen zum Verändern von Eigenschaften gibt es?
- Wie ist die Schnittstelle zum Ändern von Eigenschaften beschaffen?
- Welchen Einfluss haben QoS-Eigenschaften?
- Welche Eigenschaften haben Einfluss auf die Performance des Systems?
- Welche Garantien gibt es?
- Wie lassen sich Garantien des Systems durch Eigenschaften beeinflussen?
- Wie werden Änderungen der Eigenschaften umgesetzt (Neustart, zur Laufzeit)?
- Mit welchen Zeiteigenschaften werden Änderungen wirksam?
- Welche Ereignisse werden aufgezeichnet bzw. gelogged?
- Wie lässt sich ein System intern untersuchen, welche Debugging-Möglichkeiten gibt es?
- Welche Aspekte sind für eine Beurteilung des Systemdesigns nach dem CAP-Theorem relevant?
- Wie wirkt sich eine Skalierung des Systems auf dessen Eigenschaften aus?
- Wie dynamisch ist die Teilnahme von Knoten gestaltet?
- Können Sie Fehler induzieren (z.B. kurzzeitige Netzwerktrennung)? Wie ist das Verhalten?

Zugriffe: Beispiele für Fragen und Aufgabenstellungen

- Welche Arten von Zugriffen gibt es?
- Welche Möglichkeiten gibt es, die Art von Zugriffen zu beeinflussen?
- Wie werden Zugriffe auf dem Netzwerk dargestellt?
- Beschreiben Sie das Format von Zugriffen und seine Besonderheiten.
- Welche Zugriffsmuster sind typisch?
- Welche Entwurfskriterien lagen bei der Umsetzung von Zugriffen zu Grunde?
- Welche Hardwareeigenschaften haben einen Einfluss auf Zugriffe?
- Wie ist das zeitliche Verhalten von Zugriffen?
- Welche Einflussfaktoren gibt es bei der Dauer von Zugriffen?
- Welche zeitliche Verteilung ergibt sich bei einer großen Anzahl von Zugriffen?
- Wie ist das Verhältnis von Nutzdaten zur Gesamtmenge?
- Welchen Einfluss hat der Umfang der Nutzdaten?
- Welche Zugriffskontrollen gibt es?
- Welche Garantien für Zugriffe gibt es?
- Welche Fehlererkennungs- oder Fehlerbehebungsmöglichkeiten gibt es?
- Wie wirken sich Skalierungseigenschaften auf die Performanz von Zugriffen aus?

Beispiele für quantitative Analysen:

- Welche Latenzzeit ergibt sich aus Zugriffen mit unterschiedlicher Größe (z.B. 0, 1, 256, 4096, 1M Byte Nutzdaten)?
- Welcher Durchsatz ergibt sich aus Zugriffen mit unterschiedlicher Größe?
- Wie ist das Brutto-/Nettoverhältnis bei Zugriffen unterschiedlicher Größe?
- Wie unterscheiden sich Werte für Latenz und Durchsatz im Vergleich zwischen schreibenden und lesenden Zugriffen?
- Wählen Sie eine QoS-Eigenschaft und variieren Sie dessen Einstellungen. Analysieren Sie den Einfluss auf Performance-Werte (Latenz/Durchsatz).
- Erzeugen Sie parallele Zugriffe. Welche Einflüsse sehen Sie durch parallele Zugriffe auf Performance-Werte?

Weitere Aspekte bei quantitativen Analysen:

- Wie lassen sich einzelne Faktoren isolieren, die die Gesamtperformance bestimmen? Bestimmen Sie den Einfluss von Faktoren.
- Variieren Sie Ihren Netzwerkzugang (Bandbreite, Latenz). Welche Einflüsse können Sie bestimmen?
- Wie groß ist der Ressourcenbedarf (CPU, Speicher) zur Laufzeit?

Quellensammlung & Einstieg

Google Cloud Storage:

Einstiegsseite

<https://cloud.google.com/storage/>

Cloud Storage Referenz

<https://cloud.google.com/storage/docs/reference/libraries>

Konzepte

<https://cloud.google.com/storage/docs/concepts>

CoAP:

Informationen zu CoAP und Übersicht zu Implementierungen

<http://coap.technology/>

Spezifikation

<https://tools.ietf.org/html/rfc7252>

Folien zu CoAP auf einem Raspberry

<https://de.slideshare.net/carlosralli/curso-coap-v3slideshare>

Kleines Tutorial zu CoAP auf einem Raspberry

cs.iupui.edu/~xiaozhon/course_tutorials/Coap_tutorial_RPi.pdf

MQTT:

Building Smarter Planet Solutions with MQTT and IBM

www.redbooks.ibm.com/redbooks/pdfs/sg248054.pdf

MQTT V3.1 Protocol Specification

<http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>

John C. Shovic, „Raspberry Pi IoT Projects“, apress, online in h_da Bibliothek

<http://www.hivemq.com/blog/mqtt-essentials/>

Eclipse Mosquitto

<https://mosquitto.org/>

Explore MQTT and the Internet of Things service on IBM Bluemix

<https://www.ibm.com/developerworks/cloud/library/cl-mqtt-bluemix-iot-node-red-app/>

demnächst bei weiteren Arbeiten mit MQTT

Lucy Rogers, Andy Stanford-Clark:

„Wiring the IoT: Connecting Hardware with Raspberry Pi, Node-Red, and MQTT“

Erscheint voraussichtlich August 2017

Technologien ohne vordefinierten Schwerpunkt:

Amazon S3:

Einstiegsseite zu S3

<https://aws.amazon.com/de/s3/>

S3 API Referenz

<http://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html>

Entwicklerhandbuch

<http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>

Sample Code und Libraries

<https://aws.amazon.com/code>

Blockchain:

Satoshi Nakamoto: "Bitcoin: A Peer-to-Peer Electronic Cash System"

<https://bitcoin.org/bitcoin.pdf>

Weitere Quellen:

Bitcoin Developer Reference (working paper) - Krzysztof Okupski

<http://enetium.com/resources/Bitcoin.pdf>

Developer Guide

<https://bitcoin.org/en/developer-guide#block-chain>

Network protocol

https://en.bitcoin.it/wiki/Protocol_documentation

Bitcoin core

<https://bitcoin.org/en/download>

Bitcoin Developer Reference

Find technical details and API documentation.

<https://bitcoin.org/en/developer-reference>

Developer examples

<https://bitcoin.org/en/developer-examples>

Bitcoin test net

<https://en.bitcoin.it/wiki/Testnet>

GitHub spec

<https://github.com/minium/Bitcoin-Spec>

IBM developerWorks

<https://developer.ibm.com/blockchain/>

IBM Blockchain auf Bluemix (Start Developer Plan beta)

<https://console.ng.bluemix.net/docs/services/blockchain/index.html>

Beispiel für verfeinerte Zielsetzungen anhand der Technologie NFS

Gewählte Technologie: NFS

Schwerpunkt: Zugriffe

Titel: Vergleich schreibendes Random I/O vs. Streaming I/O über LAN

Use-Case-Szenario:

Es wird eine Applikation erstellt, die mittels eines NFS-Client auf einen NFS-Server zugreift. Es soll untersucht werden, inwieweit sich das Schreibverhalten über ein LAN mit guter Netzanbindung zwischen Random und Streaming I/O unterscheidet.

Hardware- und System-Setup:

Als NFS-Client wird ein Raspberry Pi 3 mit Raspian genutzt, der NFS-Server läuft auf einem älteren Laptop mit HDD-Festplatte (wie wäre es mit einer aktuellen SSD?) unter Linux. Die Netzwerkverbindung zwischen den Geräten ist über LAN realisiert. Das Caching von NFS ist abgeschaltet (Mount-Option noac).

Zu erstellende Demo-Software:

Die Applikation kann unterschiedliche Arbeitslast in Form von schreibenden Zugriffen auf Dateien generieren. Es können große Dateien (256MB) als Streaming-Data in sequentiell aufeinanderfolgenden Segmenten (4KB) geschrieben werden. Alternativ können Segmente (4KB) an zufälligen Stellen innerhalb von mehreren großen Dateien (256MB) geschrieben werden (Random I/O). Die Applikation ist mehrfach parallel startbar.

Evaluation:

1. Maximale Schreibrate: Die Applikation wird gestartet und schreibt als Streaming-Data immer wieder neue Dateien. Die erreichte Datenrate für einen Stream wird ermittelt. Danach wird ermittelt, ob ein zweiter (dritter, ...) Stream zu einer höheren Datenrate führt. Dies wird solange wiederholt, bis die maximale Gesamtdatenrate über den Netzwerkadapter erreicht ist oder sinkt.
2. Schreibrate Random I/O): Die Applikation wird gestartet und schreibt Random I/O. Die erreichte Datenrate wird ermittelt. Danach wird die Applikation mehrfach gestartet, um zeitlich parallel erfolgende Zugriffe zu erzeugen. Es wird ermittelt, welche Datenrate sich aus zweifachen, dreifachen, ... parallelisierten Random-Zugriffen ergibt.

Präsentationsziele (Dauer ca. 10 Minuten):

- Darstellung der Aufgabenstellung (1 Folie)
- Zu beachtende Aspekte für Zugriffe (1 Folie)
- Zu beachtende Besonderheiten (z.B. synchroner mount) (1 Folie)
- Abgrenzung von nicht betrachteten Einflussfaktoren (z.B. Netzwerktraffic) (1 Folie)
- Darstellung der Evaluationsergebnisse (2 Folien)
- Bewertung / Schlussfolgerung aus der Ergebnissen (1 Folie)
- Darstellung des Arbeitsumfangs für die Erstellung, Eindruck zur Technologiereife (1 Folie)
- Live Online-Demo

Gewählte Technologie: NFS

Schwerpunkt: Eigenschaften

Titel: Abbildung von Rollen und Rechten in NFS mit zeitlichen Aspekten

Use-Case-Szenario:

Es wird eine Applikation erstellt, die mittels eines NFS-Client auf einen NFS-Server zugreift. Es soll untersucht werden, inwieweit unterschiedliche Rollen und Rechte in NFS abgebildet sind und welche zeitliche Aspekte diese Dateieigenschaften haben. Dabei wird das klassische bitorientierte Schema (All|Group|User] untersucht.

Hardware- und System-Setup:

Sowohl für NFS-Client als auch für den –Server wird jeweils eine VM unter Linux verwendet und über das interne Netzwerk verbunden. Das Caching von NFS ist abgeschaltet (Mount-Option noac). Spezifizieren Sie zwei Dateisystemen, welche über NFS exportiert werden. Wählen Sie ein Dateisystem, welches synchron mount-bar ist (1). Ein zweites Dateisystem soll asynchron gemountet werden (2).

Zu erstellende Demo-Software:

Die Applikation kann Dateien und Verzeichnisse erzeugen und als root bzw. von unterschiedlichen Usern gestartet werden. Als root erlaubt die Applikation, die Eigentümerschaft der Dateien zwischen den verschiedenen Usern zu wechseln und die Zugriffsmöglichkeit unterschiedlicher User zu prüfen.

Ermittlung von Eigenschaften:

Welchen Einfluss haben die einzelnen Bits im bitorientierten Schema von Zugriffsrechten? Welche Zugriffsmöglichkeiten sind bei der Kombination aus Verzeichnissen und Dateien in einer hierarchischen Struktur möglich? Wie sehen Anwendungsszenarios aus, die komplexere Rechte erfordern?

Evaluation:

1. Es werden Dateien erzeugt, deren Eigentümerschaft durch die Applikation geändert wird. Wie oft lässt sich Eigentümerschaft pro Sekunde ändern? Vergleichen Sie dies zwischen den beiden System-Setups 1 und 2.

2. Lassen Sie mit der Applikation prüfen, ob der Zugriff auf Dateien erlaubt ist oder nicht. Wie oft ist ein Prüfen der Eigentümerschaft pro Sekunde möglich? Vergleichen Sie dies zwischen den beiden System-Setups 1 und 2.

Präsentationsziele (Dauer ca. 10 Minuten):

- Darstellung der Aufgabenstellung (1 Folie)
- Zu beachtende Aspekte für Zugriffe (1 Folie)
- Zu beachtende Besonderheiten (z.B. synchroner mount) (1 Folie)
- Abgrenzung von nicht betrachteten Einflussfaktoren
(z.B. komplexere Anwendungsszenarios wie Nutzung von ACL's, Abhängigkeiten durch Client/Server-Zugriffe auf das gleiche Massenspeichermedium) (1 Folie)
- Darstellung der Eigenschaften (1 Folie)
- Evaluationsergebnisse (1 Folien)
- Bewertung / Schlussfolgerung aus der Ergebnissen (1 Folie)
- Darstellung des Arbeitsumfangs für die Erstellung, Eindruck zur Technologiereife (1 Folie)
- Live Online-Demo