Verteilte Systeme – Praktikum 3
Prof. R. Moore
WS 2014/15

Auswertung

Gruppe Fr2y / Fr Y E – Lukas Köhler, Moritz Hilberg

Einleitung und Versuchsbeschreibung

In diesem Szenario laufen auf 4 VMs jeweils folgende Prozesse: 1 „AdminServer“ als Hausverwalter, 2 „HouseServer“ als Wohnungsserver mit jeweils 10 „Sensor“ Prozessen/Threads als Sensoren.
Dieser Versuch baut auf Praktikum 2 auf, d.h. die Funktionalität der Sensoren und Wohnungsserver bleibt erhalten und dem Hausverwalter wird eine Message Oriented Middleware hinzugefügt, in Form von Apache ActiveMQ, um mit anderen Hausverwaltern zu kommunizieren. So ist jeder Hausverwalter Publisher seiner eigenen Wohnungsdaten. Einmal in eine Queue für die gesamten Werte und einmal in eine Queue nur für die Extremwerte (Temperatur > 30° oder < 10°, Stromverbrauch > 2 KW). Zudem ist es dem Hausverwalter möglich, über eine Eingabe im Terminal, die Nachrichten der anderen Hausverwalter zu subscribieren. In unserem Programm ist die Eingabe „+Nr“, wobei Nr der Name des jeweils anderen Hausverwalters ist. Dieser muss eine positive, ganze Zahl sein. Mit der Eingabe wird das Abonnement von anfangs allen Meldungen, auf nur Alarmmeldungen, nach nochmaliger Eingabe auf keine Meldungen und dann wieder auf alle Meldungen gesetzt (Abonieren/Stornieren der jeweiligen Queues).

Die abonnierten Meldungen werden in der Ausgabe durch eine START und eine STOP Zeile gekennzeichnet. Beispielausgabe (Mitte gekürzt):

```
START Status From AdminServer: null: moritz-ThinkPad-Edge-S430/127.0.1.1
--------------
Flat No. 0:
--------------
Room 0 - Temp:  23      - Power:        0
Room 1 - Temp:  19      - Power:        10
Room 2 - Temp:  26      - Power:        2
--------------------------------------------------------------------------
Room 7 - Temp:  28      - Power:        0
Room 8 - Temp:  11      - Power:        2
Room 9 - Temp:  24      - Power:        1
HighestTemp of House No. 1 :    28
LowestTemp of House No. 1 :     -7
TotalPower of House No. 1 :     24
STOP Status From AdminServer null: moritz-ThinkPad-Edge-S430/127.0.1.1
```

Der ActiveMQ Broker muss auf eine VM heruntergeladen, eventl. konfiguriert und gestartet werden.
Startbefehl : /<PfadZuActiveMQ>/*apache-activemq-5.9.0/bin/activemq start*

Die 3 benötigten Jars müssen auch auf allen VMs vorhanden sein. Der Einfachheit halber haben wir ein Shellscript geschrieben das die Konfiguration auf den VMs startet.

```
#!/bin/bash

java -jar Sensor.jar -r 10 -udpPort 9998 &
java -jar HouseServer.jar -sp 8080 -ep 8081 -udpPort 9998 -tcpPort 9999 &
java -jar Sensor.jar -r 10 -udpPort 9997 &
java -jar HouseServer.jar -sp 8081 -ep 8082 -udpPort 9997 -tcpPort 9996 &
#TODO: set '-n' to 2-4 and change '-mqIp' to brokers IP
java -jar AdminServer.jar -sp 8080 -ep 8082 -t 5000 -ip localhost -n 2 -mqIp 192.168.0.20 -mqPort 61616 -o status
```

Die Aufrufe von HouseServer und Sensor sind analog zu Praktikum 2, wobei nun auch TCP und UDP Ports als Parameter angegeben werden können, um sie parallel auf einem Rechner laufen zu lassen.
Der Aufruf von AdminServer ist wie folgt strukturiert:
-sp <Startport XMLRPC> -ep <Endport XMLRPC> -t <Zeitintervall der Messung> -ip <IP des XMLRPC Servers> -n <Name 1-n> -mqIp <IP des ActiveMQ Brokers> -mqPort <Port des ActiveMQ Brokers> -o <Terminalausgabe der eigenen Werte 'status'/'alert'/'none'>

Die UDPServer der HouseServer schreiben ihre UDP Tests aus Praktikum 1b weiterhin in eine Logdatei. Der AdminServer schreibt seine XMLRPC und ActiveMQ Testergebnisse in eine separate Logdatei. Um die Frequenz zu ermitteln, wie viele Nachrichten Publisher und Listener pro Intervall austauschen können, werden alle erfolgreichen Aufrufe von MessageProducer.send(msg) und von MessageConsumer.receive() separat pro Zeitintervall gezählt und notiert. Dies geschieht auf Publisher- und Listenerseite und kann dann verglichen werden. Das Zeitintervall lässt sich flexibel über den Parameter '-t' beim Programmaufruf einstellen.

Der Lasttest erfolgt mit nur 2 VMs mit o.g. Konfiguration. Das dient dem schnellen Starten und Stoppen der Konfiguration mit verschiedenen Parametern. Last wird durch Verkürzung der fiktiven Zeit erzeugt. VM 1 publiziert alle t Sekunden seine Nachrichten in die Queues und VM 2 empfängt diese alle t Sekunden. Last wird erzeugt indem wir t gegen Null gehen lassen. So senden und empfangen die Anwendungen so viel wie dem System möglich ist. Die Messungen finden alle 10 Sekunden statt.

Versuchsdurchführung

Bemerkung: Die Summen der gesendeten und empfangenen Nachrichten (letzte Zeile) differieren maximal um die Anzahl an Nachrichten, die pro 10 Sekunden geschickt wurden. Das kommt daher, dass beide Prozesse ihre Logs zu unterschiedlichen Zeiten schreiben. Es gibt keine globale Uhr zur Synchronisierung.
Wir können aber davon ausgehen, dass keine Nachrichten verloren gehen, da MOM uns eine garantierte Auslieferung der Nachrichten zusichert. Durch persistente Zwischenspeicherung im Broker. Sind die Queues im Broker voll, blockiert der Sendevorgang der Publisher bis wieder Platz ist (siehe https://activemq.apache.org/producer-flow-control.html).

Zeitintervall t=1 Millisekunden

Publisher	Listener
Status Msg Published	Total Msg Received
4690	3986
6650	7303

6790	7875
9275	8715
8855	9380
9205	9870
Summe	
45465	47129

Zeitintervall t=0 Millisekunden

Publisher	Listener
Status Msg Published	Total Msg Received
4550	1540
7035	5425
7280	7385
10010	8190
9800	9523
9730	9924
Summe	
48405	41987

Zeitintervall t=0 Millisekunden, aber zusätzliche VM als zweiter Publisher

Publisher 1	Publisher 2	Listener
Status Msg Published	Status Msg Published	Total Msg Received
2380	3430	4465
3815	3780	8014
3430	4760	8836
5110	4760	9989
3745	5600	9891
5390	6335	11315
Summe		
23870	28665	52510
Pub1 + Pub2	52535	

Fazit

Der Durchsatz an Nachrichten ist sehr von der Netzwerktopologie und verwendeter Betriebssystemumgebung abhängig. In unserem Fall liegt der maximale Durchsatz bei etwa 9.000

Nachrichten in 10 Sekunden. Das entspricht 56.000 empfangenen Nachrichten in einer Minute.

Sehr schön ist in unserem Versuch der Producer-Flow-Control von ActiveMQ zu sehen. Schafft es ein Publisher im Schnitt 9.000 Nachrichten / 10 Sekunden in seine Queue zu schreiben, schaffen 2 Publisher (3. Versuch) in der gleichen Zeit je nur 4.500 Nachrichten im Schnitt.