UNIVERSITÄT SIEGEN

Fakultät IV
Betriebssysteme und verteilte Systeme
Prof. Dr. rer. nat. Roland Wismüller

Aufgabenblatt 6

(Zu bearbeiten bis 05.06.2020)

Vorlesung Verteilte Systeme Sommersemester 2020

Aufgabe 1: Java Sicherheitsmanager

Java Security wird seit Java 2 mit Hilfe von Security-Policies und dem Java-Security-Manager umgesetzt (siehe Kap. 3.4.2 der Vorlesung).

a) Erklären Sie kurz die Funktionsweise des Security-Managers. Was wird durch den Security-Manager geschützt und wovor?

b) Erweitern Sie das Code-Beispiel im Archiv `u06Files.zip`[1] auf der Vorlesungswebseite so, daß der Client nur Dateien schreiben kann, die im Verzeichnis `/tmp` liegen. Versehen Sie die JVM dazu zunächst mit einem Security-Manager:

```
System.setSecurityManager(new SecurityManager());
```

Starten Sie nun den Client:

```
java Client /tmp/test
```

Was passiert und warum?

Erstellen Sie jetzt eine Datei `policy` mit der entsprechenden Sicherheitsrichtlinie und starten Sie den Client erneut:

```
java -Djava.security.policy=policy Client /tmp/test
```

c) Erstellen Sie nun einen eigenen Security-Manager, der zusätzlich(!) verhindert, daß der Client Dateien mit der Endung `.java` schreiben kann (selbst wenn diese unter `/tmp` liegen). Erweitern Sie dazu die Klasse `SecurityManager` und überschreiben Sie die Methode `public void checkWrite(String file)`. Informieren Sie sich über die Methode mit Hilfe der Java-Dokumentation zur Klasse `SecurityManager`.

Aufgabe 2: Lastverteilungsstrategien

Lastverteilung in verteilten Systemen hat das Ziel, die auftretende Last möglichst gleichmäßig zu verteilen, um Überlast auf einzelnen Knoten zu verhinden. Ein ehrgeizigeres Ziel wäre es, die Last so zu verteilen, dass der Gesamtdurchsatz maximiert wird.

a) Wie unterscheiden sich statische, dynamische und präemptiv-dynamische Lastverteilung voneinander? Welche Vor- und Nachteile haben die Verfahren und welche Voraussetzungen müssen jeweils gegeben sein, um eins der Verfahren anwenden zu können?

b) Alle Lastverteilungsstrategien gründen ihre Entscheidungen auf irgendeine *Lastmetrik*. Die einfachste, oft betrachtete Lastmetrik bei theoretischer Analyse der Algorithmen ist die Anzahl der (rechenbereiten) Prozesse pro Knoten. Warum ist diese Metrik in der Praxis oft nicht ausreichend? Was wäre eine bessere Metrik und welche neuen Probleme treten dann auf?

[1] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/u06Files.zip`

Aufgabe 3: Graphpartitionierung und List-Scheduling

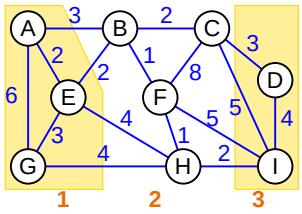
In der Vorlesung (Kap. 5.1.1) wurden zwei Arten von verteilten Scheduling-Algorithmen vorgestellt, die auf Graphen basieren:

- Lastverteilung durch Graphpartitionierung und
- List-Scheduling.

Welche Gemeinsamkeiten und Unterschiede können Sie zwischen diesen beiden Klassen von Algorithmen feststellen? Unter welchen Voraussetzungen sind sie anwendbar?

Aufgabe 4: Lastverteilung durch Graphpartitionierung

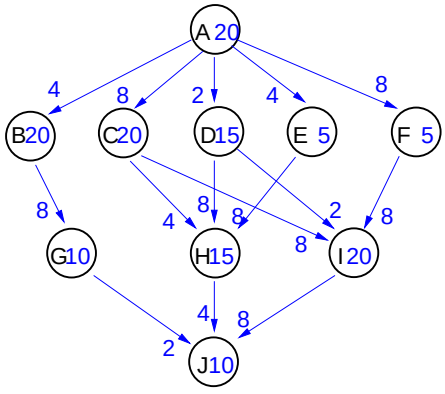
Erinnern Sie sich an die Lastverteilung durch Graphpartitionierung (s. Vorlesung Kap. 5.1.1). Nehmen Sie an, daß in der unten stehenden Abbildung der Prozess H von Knoten 2 auf Knoten 3 wechselt. Wie groß ist danach der (gesamte) Netzwerkverkehr zwischen den Knoten?



Aufgabe 5: List-Scheduling

In der Vorlesung wurden verschiedene Ansätze zum verteilten Scheduling vorgestellet (s. Kap. 5.1.1). Zwei häufig verwendete List-Scheduling Algorithmen sind: *High Level First with Estimated Time* (HLFET) und *Earliest Task First* (ETF). Der erste war schon in der Vorlesung dargestellt (s. Animation auf Folie 175: List-Scheduling mit HLFET). Für den zweiten Algorithmus finden Sie eine genaue Beschreibung im dem Artikel von Hagras und Janacek (über die Vorlesungsseite oder https://ojs.cvut.cz/ojs/index.php/ap/article/download/490/322) auf Seite 19.

Gegeben sei ein System mit 3 Prozessoren und der unten gezeigte DAG, der ein Programm, bestehend aus mehreren abhängigen Tasks, modelliert. Die Knoten beinhalten Tasks mit Ausführungszeiten und die Kanten zeigen die notwendige Kommunikation mit Übertragungsdauer.



Wie werden die Tasks auf die drei Processoren gescheduled (mit HLFET / mit ETF)? Zeichen Sie jeweils ein Gantt-Diagramm. Nehmen Sie an, dass eine lokale Kommunikation keine Zeit kostet.

Aufgabe 6: Prozeßmigration

Einige Multicomputer gestatten die Migration laufender Prozesse von einem Knoten zum anderen. Ist es ausreichend, den Prozeß anzuhalten, das Speicherabbild einzufrieren und das Ganze auf einen anderen Knoten zu übertragen? Geben Sie zwei nicht-triviale Probleme an, die gelöst werden müssen, damit das Modell funktioniert.

Aufgabe 7: Policies dynamischer Lastverteilungs-Systeme

a) Beschreiben Sie die vier Policies (Strategien), die normalerweise Bestandteile eines dynamischen Lastverteilungs-Systems sind. Geben Sie für jede Strategie an:

- eine intuitive Beschreibung des Zweckes dieser Strategie,
- ein Beispiel einer spezifischen Implementierung dieser Strategie.

b) Warum sind empfänger-initiierte Verfahren zum dynamischen Lastausgleich manchmal nicht erwünscht?