UNIVERSITÄT SIEGEN

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

Excercise Sheet 6

(To be processed until 05.06.2020)

Lecture Distributed Systems Summer Term 2020

Exercise 1: Java Security Manager

Java Security has been implemented since Java 2 with the help of security policies and the Java Security Manager (see chapter 3.4.2 of the lecture).

a) Briefly explain how the security manager works. What is protected by the security manager and from what?

b) Expand the code example in the archive `u06eFiles.zip`[1] on the lecture's web page so that the client can only write files located in the `/tmp` directory. Provide the JVM with a security manager for this purpose:

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

Now start the client:

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

What happens and why?

Now create a `policy` file with the appropriate security policy and restart the client:

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

c) Now create your own security manager that additionally(!) prevents the client from writing files with the extension `.java` (even if they are under `/tmp`). Extend the class `SecurityManager` and overwrite the method `public void checkWrite(String file)`. Read about the method in the Java documentation for the `SecurityManager` class.

Exercise 2: Load balancing strategies

a) How do static, dynamic and preemptive dynamic load balancing differ from each other? Which advantages and disadvantages do the methods have and which prerequisites must be given in each case in order to be able to apply one of the methods?

b) All load distribution strategies base their decisions on some *load metric*. The simplest, often considered load metric for theoretical analysis of algorithms is the number of (ready) processes per node. Why is this metric often not sufficient in practice? What would be a better metric and what new problems arise then?

Exercise 3: Graph partitioning and list scheduling

In the lecture (chapter 5.1.1) two types of distributed scheduling algorithms based on graphs were presented:

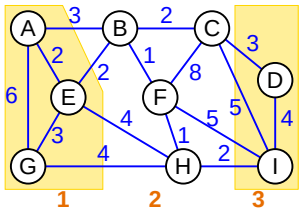
- load distribution through graph partitioning and
- list scheduling.

What similarities and differences can you find between these two classes of algorithms? Under what conditions are they applicable?

[1] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/u06eFiles.zip`

Exercise 4: Loda Distribution via Graph Partitioning

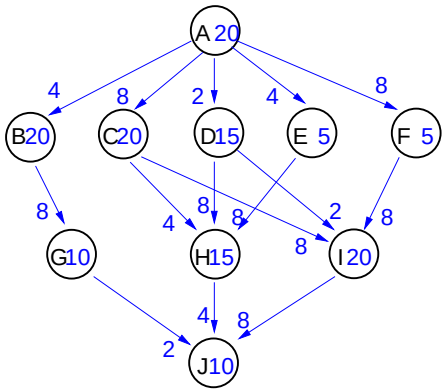
Remember the load distribution by graph partitioning (see lecture chapter 5.1.1). Assume that in the figure below the process H changes from node 2 to node 3. How large is the resulting (total) network traffic between the nodes?



Exercise 5: List Scheduling

The lecture presented different approaches to distributed scheduling (see chapter 5.1.1). Two frequently used list scheduling algorithms are: *High Level First with Estimated Time* (HLFET) and *Earliest Task First* (ETF). The first one was already shown in the lecture (see animation on slide 175: List Scheduling with HLFET). For the second algorithm you can find a detailed description in the article by Hagras and Janacek (accessible via the lecture page or via the URL https://ojs.cvut.cz/ojs/index.php/ap/article/download/490/322) on page 19.

Given is a system with 3 processors and the DAG shown below, which models a program consisting of several dependent tasks. The nodes contain tasks with execution times and the edges show the necessary communication with transmission time.



How are the tasks scheduled on the three processors (with HLFET / with ETF)? Draw one Gantt chart for each. Suppose a local communication doesn't cost any time.

Exercise 6: Process Migration

Some multicomputers allow the migration of running processes from one node to another. Is it sufficient to stop the process, freeze the memory image, and transfer the whole thing to another node? Specify two non-trivial problems that must be solved for the model to work.

Exercise 7: Policies of Dynamic Load Balancing Systems

a) Describe the four policies (strategies) that are usually components of a dynamic load balancing system. Specify for each strategy:

- an intuitive description of the purpose of this strategy,
- an example of a specific implementation of this strategy.

b) Why are receiver-initiated procedures for dynamic load balancing sometimes undesirable?