Prof. Dr. Oscar Nierstrasz
Pascal Gadient, Leonel Merino

Solution
Series 12 — 06.12.2017 – v1.0
Architectural Styles for Concurrency

Deadline for submission: 10-Dec-2017, 1159 pm

Exercise 1 (7 Points)

Answer the following questions:

a) What is a *Software Architecture*? What is its benefit? **Answer:**

A Software Architecture *defines a system in terms of computational components and interactions amongst those components. It breaks the overall complexity down to several smaller blocks that are easier to handle.*

b) What are the potential disadvantages when using layered architectures? **Answer:**

It increases the development overhead in small projects. Further, it restricts the freedom of choice in terms of coding styles.

c) Provide an example in which the pattern *Specialist Parallelism* could be a legitimate architectural choice. Justify your answer! **Answer:**

Specialist Parallelism *makes very much sense for different workloads that require different computation strategies. Web-servers, for example, have very often distinct threads for network communication and data processing. These two tasks have quite different requirements: network communication threads have to be short and fast (to increase the throughput), whereas data processing threads may request more time to complete as they potentially rely on external data that first needs to arrive. Concludingly, we could use specialized threads for each of these purposes to increase the overall responsiveness (short network related tasks won't get delayed) and to reduce the complexity of the system (by avoiding the introduction of different use cases for the complex threads).*

d) The concepts *Result Parallelism*, *Specialist Parallelism* and *Agenda Parallelism* represent three ways of thinking about the problem. Can you tell on what they focus? Provide one sentence for each one of them. **Answer:**

Result Parallelism *focuses on the shape of the finished product.* Specialist Parallelism *focuses on the makeup of the work crew.* Agenda Parallelism *focuses on the list of tasks to be performed.*

e) What is a *Flow Architecture*? What are *Blackboard Architectures*? **Answer:**

In Flow Architectures *the synchronization is managed by arranging things in a way that information only flows in one direction from sources over filters to sinks.* Blackboard Architectures *put all synchronization in a "coordination medium" where agents can exchange messages.*

f) Which blackboard style should be preferred when we have multiple processors? Why? **Answer:**

The Agenda Parallelism *should be selected, as it allows you to instantiate as many threads as CPU cores exist.*

g) What are Unix pipes and how do you use them? **Answer:**

Unix pipes (established through the "`|`" symbol) are bounded buffers that connect producer and consumer processes. They can be used in shell commands to "connect" different input / output streams together to easily perform complex data manipulations. For example, `ls -f | wc -l` *that first lists each file (and the* `.` *and* `..` *directory entries) in a single line with help from the application* `ls`*. Then this output gets processed by the application* `wc` *which counts the amount of lines. In the end, we retrieve the number of files in a specific folder +2 because of the two directory references that also will be included.*

Exercise 2 (3 Points)

Now we change roles: It is your turn to send me questions of topics you are still not familiar with for the next practical session. The best questions will be presented in front of you. I will try my best to answer all the questions you submit. For each question you ask you will retrieve a point (maximum of 3 points, no bonus this time :). **Answer:**

Please refer to the course website `http://scg.unibe.ch/teaching/cp`.