

VU Scheduling Approaches in Distributed Systems.

Task 2 – "Modelling and Optimization using OpenDSE".

Student/Group:

- Tobias Kupek
- Simon Kleinfeld

General Feedback:

Timing Evaluator: Super cool. Very clear and well-structured solution. Can't really add much here.

Cost Evaluator: Also good solution. You do not need to take care of thread-safety here. When evaluating multiple individuals in parallel, it is not necessary since each individual has its own implementation. The different evaluators, on the other hand, are applied sequentially on the individuals.

Code Feedback:

- A matter of taste, but personally, I would not write function which return null (line 109). It can be quite ugly to debug since a null can be handed through a large number of stack levels before causing a difficult-to-interpret NullPointerException far away from the point where it was caused. As a suggestion, at this point I would let the method throw an exception if it does not find a root. This would inform any user of the method that there is a possibility that the graph does not have a root and force them to handle this within a catch. Alternatively, you could also always throw an exception if you do not have a root (and provide a hasRoot() method to circumvent this case). (And yeah, I know we do return null in case that we don't change the implementation. Not my code, not a fan of the solution :))
- wow, the map processing with looking for the max is super cool AND readable. Gotta look into that style myself :)
- Sanity check on the number of chosen resources for a task – awesome
- Just as a note: The thing you do in the updateUsage method is exactly why we use Property-Services in OpenDse (the AbstractPropertyService features methods for checking whether attribute is set, returning it if set etc.). Of course, unnecessary for this small example, but in general, you could put this into a property service.
- Second time I read over the stream syntax. It is still awesome :)

Summary:

Cool solution. Taught me some cool new syntax :) .

Additional Questions:

-

Best regards,

Fedor