SCC: Project

# General details

Topic: Multi-objective task scheduling system

Strategy: MOHEFT [1]

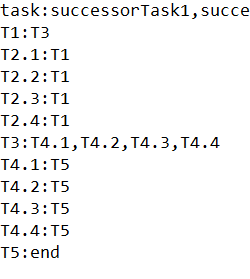
Workflow: Custom

Objectives: makespan & cost

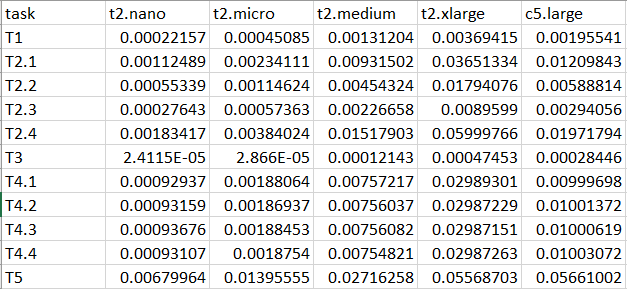
# diagram Overview of approach



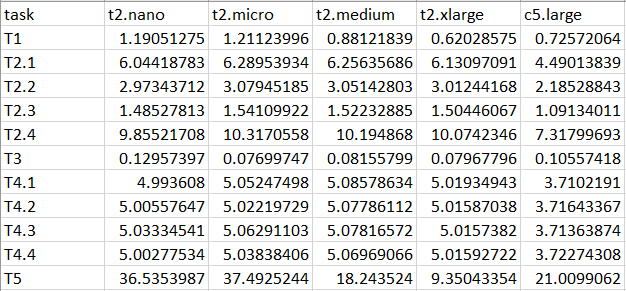
# Workflow Overview

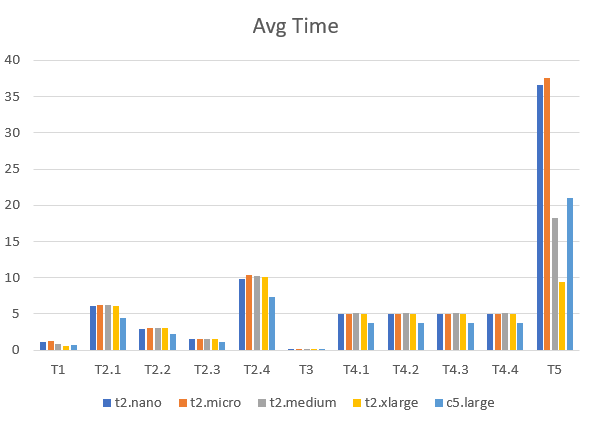
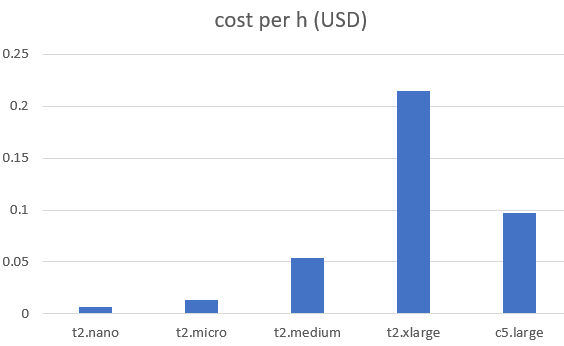
We start with 2.1, 2.2, 2.3 !

Costs.CSV



Makespan.CSV





# TASK SCHEDULING ALGORITHM

We based our task scheduling algorithm on the Cloud-aware MOHEFT algorithm presented in [1].

# IMPLEMENTATION

The task scheduling algorithm based on MOHEFT, as well as the calculation of the Nadir point and Utopia point were realized using Java 10. For the calculation of the hypervolume, we reused a Python-based script developed by a member of

# Evaluation

1. Calculate optimal solution (feasible due to small problem)
2. Perform multiple runs of the task scheduling algorithm to compute approximated solutions
3. Compare solutions from different runs with hypervolume
4. Evaluate distance between best approximation and optimum

# RESULTS

# EXAMPLE USAGE



# EXAMPLE USAGE (continued)



# References

|  |  |
| --- | --- |
| [1] | J. Durillo and R. Prodan, “Multi-objective workflow scheduling in Amazon EC2,” *Cluster computing,* vol. 17, no. 2, pp. 169--189, 2014.  [2] Pure Python-based Hypervolume calculation script by Simon Wessing (TU Dortmund University), obtainable from https://ls11-www.cs.tu-dortmund.de/rudolph/hypervolume/start |

It should be more like a combination of a technical paper and program documentation, where full description of the developed scheduling algorithm should be presented.

Within the 6-8 pages, you should shortly describe the problem area. Then you should proceed with short definition of the problem that you want to solve.

Then you should present your approach. This description should contain detailed algorithm.

Afterwards you proceed by giving details on the implementation of the algorithm (such as which programing languages and technologies you used).

At the end you simply finish with a simple running tutorial and simple execution results.