Bi-Objective Workflow Scheduling in the Cloud: What is the Real State-of-the-Art?

Yury Semenov [1] and Oleg Sukhoroslov [2]

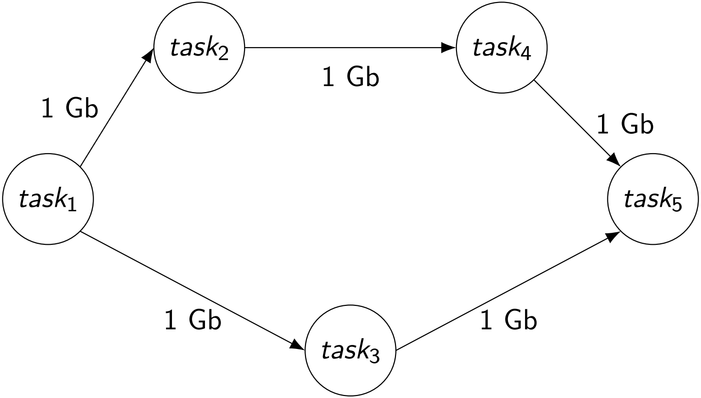
[1]HSE

[2]IITP RAS

Russian Supercomputing Days 2024

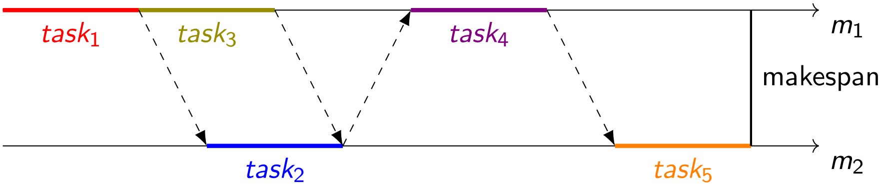
Workflow

A workflow is a set of computational tasks with data dependencies between them.



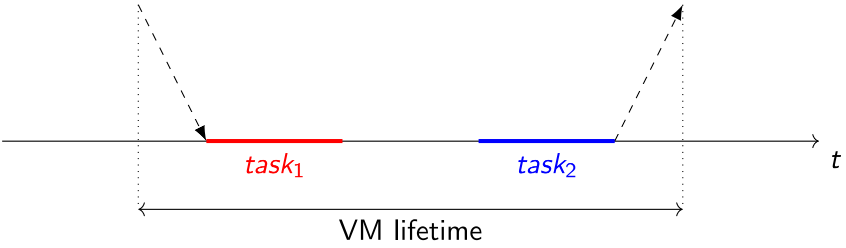
Workflow Scheduling

- Tasks can't overlap or be preempted.
- Tasks have to wait for transfer of all required data.
- Scheduling goal is to minimize makespan.



Cloud Resources

- The cloud offers a set of VM types, characterized by CPU speed, CPU cores, and price.
- Lifetime l_v of VM v must include all data transfers to/from v and all tasks executed on v.
- Cost of VM v equals $p_v \cdot \lceil \frac{l_v}{I} \rceil$ where I is the billing interval and p_v is the price of VM v.
- Same network bandwidth between all VMs.
- Minimizing execution cost is another scheduling goal.



Bi-Objective Optimization

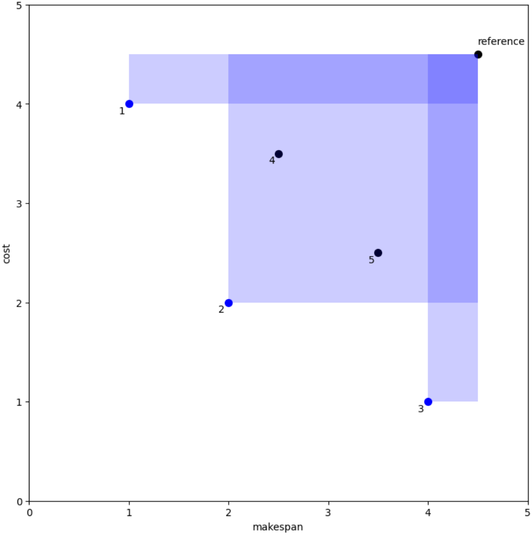


Figure 1: An example Pareto front (points 1, 2, 3). Points 4 and 5 are dominated

Algorithms

1. KAMSA: a genetic algorithm from *Zhang, H., Zheng, X.: Knowledge-driven adaptive evolutionary multi-objective scheduling algorithm for cloud workflows.*
2. VCAES: a genetic algorithm from *Li, J., Xing, L., Zhong, W., Cai, Z., Hou, F.: Decision variable contribution based adaptive mechanism for evolutionary multi-objective cloud workflow scheduling.*
3. VMALS: an ant colony algorithm from *Wang, Y., Zuo, X., Wu, Z., Wang, H., Zhao, X.: Variable neighborhood search based multiobjective ACO-list scheduling for cloud workflow.*
4. CMSWC: a list scheduling algorithm from *Han, P., Du, C., Chen, J., Ling, F., Du, X.: Cost and makespan scheduling of workflows in clouds using list multiobjective optimization technique.*

Experiment Setup

- The algorithms were implemented in DSLab DAG simulator [1].
- Size of the Pareto front ≤ 100.
- At most $200n$ objective function evaluations for n-task workflow.
- For each workflow the objectives are normalized by maximum makespan and cost, and the hypervolume is calculated with reference point $(1.1, 1.1)$.
- Mean hypervolume of 10 runs.
- 1 hour and 1 second billing intervals.

[1] https://github.com/osukhoroslov/dslab

Experiment Setup: Workflows

We use workflows provided by the WfCommons project[2] with ≤ 250 tasks.

application	domain	#	width	depth
1000Genome	bioinformatics	7	28-156	3
BLAST	bioinformatics	10	40-100	3
BWA	bioinformatics	5	100	3
Cycles	agroecology	5	32-108	4
Epigenomics	bioinformatics	7	9-59	9
Montage	astronomy	4	18-108	8
Seismology	seismology	2	100-200	2
SoyKB	bioinformatics	3	50-100	11
SRA Search	bioinformatics	25	11-51	3-4

[2] https://github.com/osukhoroslov/wf-scheduling-benchmark

Experiment Setup: VM Types

name	speed	vCPU	hourly price
m5.large	3100	2	0.096
m5.xlarge	3100	4	0.192
m5.2xlarge	3100	8	0.384
m5.4xlarge	3100	16	0.768
c5.large	3600	2	0.085
c5.xlarge	3600	4	0.17
c5.2xlarge	3600	8	0.34
c5.4xlarge	3600	16	0.68

Results

Table 1: Average hypervolume (1-hour billing interval).

DAGs	KAMSA	VCAES	VMALS	CMSWC
1000genome (small)	0.805	0.904	0.932	0.933
1000genome (medium)	1.004	1.056	1.072	1.071
blast (small)	0.990	0.991	1.012	1.011
bwa (small)	1.124	1.132	1.158	1.157
cycles (small)	1.068	1.147	1.180	1.178
cycles (medium)	1.092	1.166	1.194	1.192
epigenomics (small)	0.849	0.955	1.027	1.027
epigenomics (medium)	1.010	1.068	1.111	1.112
montage (small)	0.964	1.104	1.145	1.139
montage (medium)	1.110	1.161	1.186	1.182
seismology (small)	1.165	1.167	1.193	1.191
seismology (medium)	1.181	1.182	1.200	1.199
soykb (small)	0.959	1.035	1.061	1.068
soykb (medium)	1.086	1.124	1.143	1.146
srasearch (small)	1.041	1.113	1.177	1.173
all DAGs (small)	1.011	1.070	1.118	1.115
all DAGs (medium)	1.053	1.104	1.129	1.129

Results

Table 2: Average hypervolume (1-second billing interval).

DAGs	KAMSA	VCAES	VMALS	CMSWC
1000genome (small)	0.824	0.916	0.943	0.944
1000genome (medium)	0.991	1.045	1.064	1.063
blast (small)	0.749	0.749	0.780	0.780
bwa (small)	1.113	1.115	1.150	1.149
cycles (small)	1.068	1.147	1.180	1.178
cycles (medium)	1.091	1.165	1.194	1.191
epigenomics (small)	0.849	0.955	1.028	1.027
epigenomics (medium)	1.010	1.067	1.111	1.112
montage (small)	0.962	1.101	1.143	1.137
montage (medium)	1.101	1.150	1.178	1.174
seismology (small)	1.165	1.167	1.193	1.191
seismology (medium)	1.181	1.182	1.200	1.199
soykb (small)	0.970	1.036	1.061	1.066
soykb (medium)	1.027	1.097	1.131	1.132
srasearch (small)	0.980	1.042	1.141	1.137
all DAGs (small)	0.939	0.993	1.059	1.057
all DAGs (medium)	1.039	1.095	1.124	1.124

Conclusion

- Algorithm 3 (VMALS) consistently outperforms the competitors in almost all cases.
- Algorithm 4 (CMSWC) offers comparable performance to algorithm 3 while being much simpler to implement.

Thank You!