Integrated Data Placement and Task Assignment for Scientific Workflows in Clouds

Kamer Kaya

Ümit V. Çatalyürek(Ohio State University)
Bora Uçar(CNRS, ENS Lyon)

08/06/2011

Scientific workflows

- Scientific applications $\longrightarrow$ scientific workflows.

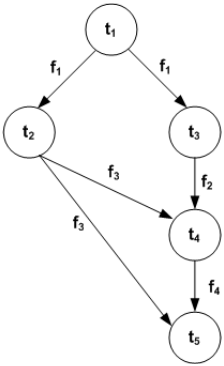


Figure: A toy workflow $\mathcal{W} = (\mathcal{T}, \mathcal{F})$ with $N = 5$ tasks and $M = 4$ files.

Cloud model

- K execution sites: $S = \{s_1, s_2, \cdots, s_K\}$
  - used for storing files and executing tasks,
  - with different characteristics: storage, computation power, cost etc.,
  - with different desirabilities.

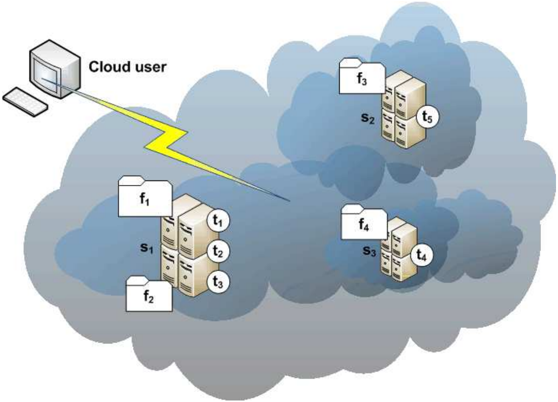


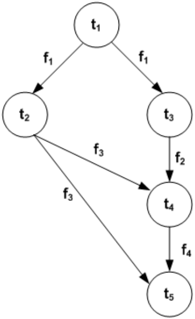
Figure: A simple cloud and assignment of the tasks and files in toy workflow.

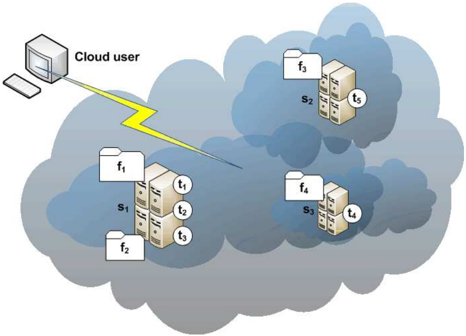
Notation

- $size(f_i)$: size of file f_i.
- $exec(t_j)$: computational load of a task t_j.
- The desirability of each site:
  - $des_f(s_k)$: storage desirability of site s_k.
  - $des_t(s_k)$: computational desirability of site s_k.
  - $\sum_{k=1}^{K} des_f(s_k) = \sum_{k=1}^{K} des_t(s_k) = 1$.
- After the assignment, for each site s_i, we want

$$\frac{size(files(s_i))}{size(\mathcal{F})} \approx des_f(s_i) \text{ and } \frac{\sum_{t_j \in tasks(s_i)} exec(t_j)}{\sum_{t_j \in \mathcal{T}} exec(t_j)} \approx des_t(s_i)$$

Costs and loads





- Total communication: $size(f_2) + 2 \times size(f_3) + size(f_4)$
- Computation and storage load for s_1:

$$\frac{\sum_{i=1}^{3} exec(t_i)}{\sum_{i=1}^{5} exec(t_i)} \text{ and } \frac{\sum_{i=1}^{2} size(f_i)}{\sum_{i=1}^{4} size(f_i)}$$

Hypergraph partitioning problem

- $\mathcal{H}=(\mathcal{V},\mathcal{E})$: a set of vertices $\mathcal{V}$ and a set of nets (hyperedges) $\mathcal{E}$.
- Weights can be associated with the vertices and costs can be associated with nets.
  - $w(v_i)$: weight of a vertex $v_i \in \mathcal{V}$,
  - $c(n_j)$: cost of a net $n_j \in \mathcal{E}$.
- A K-way partition Π satisfies the following:
  - $\mathcal{V}_k \neq \emptyset$ for $1 \leq k \leq K$,
  - $\mathcal{V}_k \cap \mathcal{V}_\ell = \emptyset$ for $1 \leq k < \ell \leq K$,
  - $\bigcup_k \mathcal{V}_k = \mathcal{V}$.
- We use the *connectivity - 1* metric with the net costs:

$$cutsize(\Pi) = \sum_{n_j \in \mathcal{E}_C} c(n_j)(\lambda_j - 1)$$

where λ_j is the number of part n_j touches.

Hypergraph partitioning problem

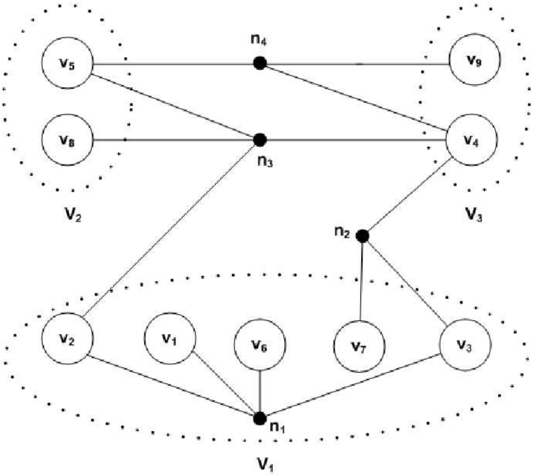


Figure: A toy hypergraph with 9 vertices 4 nets, and a partitioning with $K = 3$. Cutsize (w.r.t. to the *connectivity - 1* metric) is $c(n_2) + 2 \times c(n_3) + c(n_4)$.

Hypergraph partitioning problem

- A K-way vertex partition of $\mathcal{H}$ is said to be balanced if

$$W_{max} \leq W_{avg} \times (1+\varepsilon)$$

  where W_{max} and W_{avg} are the maximum and average part weights, respectively, and ε is the predetermined imbalance ratio.
- Multi-constraint hypergraph partitioning:
  - Multiple weights $w(v,1),\ldots,w(v,T)$ are associated with each $v \in \mathcal{V}$.
  - The partitioning is balanced if

$$W_{max}(t) \leq W_{avg}(t) \times (1+\varepsilon(t)), \quad \text{for } t = 1,\ldots,T.$$

Proposed hypergraph model

Given a workflow $\mathcal{W} = (\mathcal{T}, \mathcal{F})$, we create a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ as follows:

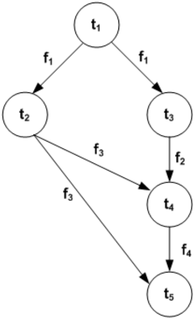
- We have two types of vertices in $\mathcal{V}$:
  1. Task vertices (v_i) which correspond to tasks $t_j \in \mathcal{T}$
     - ★ $w(v_i, 1) = exec(t_j)$ and $w(v_i, 2) = 0$.
  2. File vertices (v_i) which correspond to files $f_k \in \mathcal{F}$.
     - ★ $w(v_i, 1) = 0$ and $w(v_i, 2) = size(f_k)$.
- For each file $f_i \in \mathcal{F}$, we have a net $n_i \in \mathcal{E}$:
  - n_i is connected to the vertices corresponding to f_i itself, and the ones corresponding to tasks $\mathcal{T}$ which use f_i.
  - $c(n_i) = size(f_i)$.

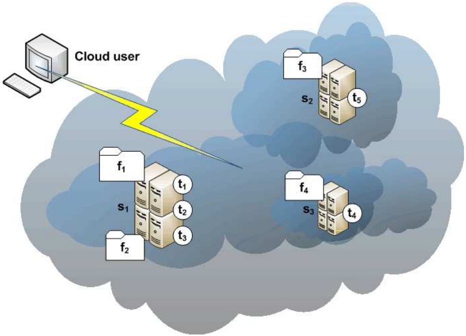
Integrated file and task assignment

- We partition the generated hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ into K parts.
- The *connectivity - 1* metric is equal to the total amount of file transfers.
- While minimizing the cutsize, we have two constraints:
  1. $des_t(s_i)$ values are not exceeded for each execution site s_i.
  2. $des_f(s_i)$ values are not exceeded for each execution site s_i.
- Multi-constraint hypergraph partitioning tool is (only) satisfied by PaToH [Çatalyürek and Aykanat, 1999].
- Problem: Non-unit net costs and target part weights are not available in PaToH v3.1.
- Solution: We improved PaToH by implementing these features and made them available in PaToH v3.2.

Integrated file and task assignment

Just to remember:





Integrated file and task assignment

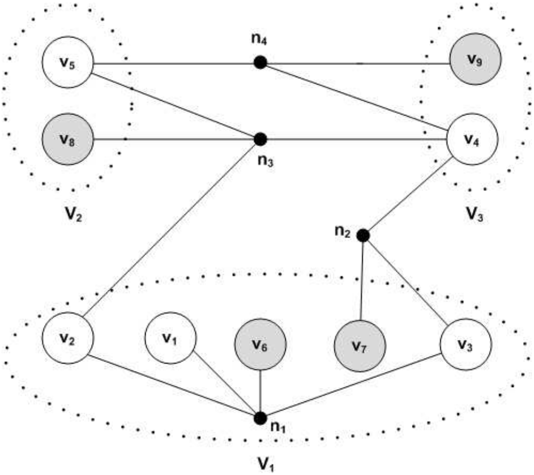


Figure: A simple 3-way partitioning for the toy workflow. The white and gray vertices represent, respectively, the tasks and the files in the corresponding workflow.

Another approach

A similar approach by [Yuan et al., 2010]:

- Files are clustered with respect to task usage and assigned to execution sites.
- A task is then assigned to the site having most of its required files.
- If a new file is generated, it is assigned to a similar cluster.

We adapted their ideas to our case:

- Files are partitioned by using MeTiS [G. Karypis and V. Kumar, 1998].
- Tasks are visited in decreasing order of their execution times.
- A task is assigned to a suitable site which has the largest amount of required files.

Experimental results

- We compared two approaches:
  1. DP: existing (consecutive) approach.
  2. DPTA: proposed (integrated) approach.
- Algorithms are run 10 times and the averages are listed.
- Both approaches were fast. For the largest workflow
  1. DP runs in 7 seconds,
  2. DPTA runs in 3 seconds

  on a 2.53 GHz MacBook Pro

Experimental results: Data set

We used the following workflows from Pegasus web page: (`https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator`)

- `CYBERSHAKE.n.1000.0`, referred to as C-shake in table;
- `GENOME.d.11232795712.12`, referred to as Gen-d,
- `GENOME.n.6000.0`, referred to as Gen-n,
- `LIGO.n.1000.0`, referred to as Ligo;
- `MONTAGE.n.1000.0`, referred to as Montage;
- `SIPHT.n.6000.0`, referred to as Sipht.

We also used three synthetically generated workflows.

Experimental results: Data set

Name	N	M	# files per task avg	min	max	# tasks per file avg	min	max
C-shake	1000	1513	3	1	5	2	1	92
Gen-d	3011	4487	3	2	35	2	1	736
Gen-n	5997	8887	3	2	114	2	1	1443
Ligo	1000	1513	6	2	181	4	1	739
Montage	1000	843	7	2	334	8	1	829
Sipht	6000	7968	65	2	954	49	1	4254
wf6k	6000	6000	9	1	18	9	1	17
wf8k	8000	8000	9	1	18	9	1	17
wf10k	10000	10000	9	1	19	9	1	17

Table: The data set contains six benchmark workflows (first six in the table) from Pegasus workflow gallery, and three synthetic ones.

Experimental results

- File imbalance: $\max_i \left(1 + \frac{\left|\frac{size(files(s_i))}{size(\mathcal{F})} - des_f(s_i)\right|}{des_f(s_i)}\right)$
- Task imbalance: $\max_i \left(1 + \frac{\left|\frac{\sum_{t_j \in tasks(s_i)} exec(t_j)}{\sum_{t_j \in \mathcal{T}} exec(t_j)} - des_t(s_i)\right|}{des_f(s_i)}\right)$
- Communication cost: $\frac{\text{total file transfer}}{size(\mathcal{F})}$

Experimental results: real-world workflows

Data	K	DP			DPTA		
		Tasks	Files	Comm	Tasks	Files	Comm
C-shake	4	1.000	1.388	0.123	1.199	1.619	0.119
	8	1.002	1.388	0.294	1.192	1.465	0.489
	16	1.005	1.554	0.613	1.553	1.733	0.809
	32	1.031	2.865	0.780	1.932	2.670	0.882
Montage	4	1.003	1.007	0.932	1.002	1.001	0.564
	8	1.063	1.006	1.564	1.007	1.006	0.863
	16	1.181	1.254	1.931	1.023	1.121	1.153
	32	1.248	2.108	2.312	1.137	2.374	1.568
Sipht	4	1.000	1.001	1.223	1.000	1.000	0.604
	8	1.000	1.002	1.850	1.003	1.004	1.300
	16	1.000	1.030	3.781	1.016	1.014	2.923
	32	1.001	1.031	7.224	1.059	1.037	5.515
Average		1.000	1.000	1.000	1.124	1.048	0.615

Experimental results: synthetic workflows

		DP			DPTA		
Data	K	Tasks	Files	Comm	Tasks	Files	Comm
wf6k	16	1.008	1.030	4.546	1.005	1.002	2.044
	32	1.036	1.030	5.407	1.009	1.003	2.765
	64	1.348	1.030	6.032	1.130	1.052	3.184
wf8k	16	1.007	1.030	4.603	1.004	1.002	2.208
	32	1.026	1.030	5.462	1.009	1.003	2.975
	64	1.218	1.030	6.066	1.099	1.032	3.118
wf10k	16	1.003	1.030	4.614	1.003	1.001	2.076
	32	1.016	1.030	5.472	1.007	1.003	2.757
	64	1.141	1.030	6.095	1.176	1.074	3.228
Average		1.000	1.000	1.000	0.968	0.989	0.501

Conclusions

- We proposed an integrated approach for assigning tasks and placing files in the Cloud.
- We modeled a scientific workflow as a hypergraph.
- We enhanced the PaToH to encapsulate the arising partitioning problem.
- We claim that the proposed approach is extremely effective for data-intensive workflows.
- Dynamic workflows (repartitioning?)
- Replication (partitioning with replication?)
- Fixed location for files (partitioning with fixed vertices?)
- Makespan ?

References

D. Yuan, Y. Yang, X. Liu, and J. Chen. (2010)
A data placement strategy in scientific cloud workflows.
Future Generation Computing Systems, 26:12001214, October 2010.

Ü. V. Çatalyürek and C. Aykanat. (1999)
PaToH: A multilevel hypergraph partitioning tool, version 3.0.
Technical Report BU-CE-9915, Computer Engineering Department, Bilkent University, 1999.

G. Karypis and V. Kumar. (1998)
MeTiS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 4.0.
University of Minnesota, Department of Comp. Sci. and Eng., Army HPC Research Center, Minneapolis, 1998.