Adaptive Resource Allocation with Dynamic Job-Clustering in Distributed Computing Systems

Masnida Hussin, Young Choon Lee, Albert Y. Zomaya
School of Information Technology

FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

1.INTRODUCTION

Large-scale distributed computing systems (LDCS) coordinate resource sharing in dynamic environments and have emerged as a popular computing platform for computationally intensive jobs.

However, diverse characteristics in both jobs and resources impose a great burden on resource allocation.

We present a dynamic job-clustering technique that enables judicious resource allocation decisions achieving better performance and utilization.

Objectives:

- Minimize response time
- Maximize resource utilization

2. PRELIMINARIES

- Each job *i* has [*w, d*], job weight ratio, $(w/d)_i$ and deadline factor df_i.
- 3 types of job:

  (i) w ≤ d , (ii) w ≈ d or (iii) w ≥ d
- Each site *x* has various compute nodes and each node *j* is characterized by [*p*, *a*]
- Global scheduler acts as decision maker for *i*; required to be map into *j*.
- It may not be considered task-mapping if the availability of *j* is very low (i.e., overloaded)

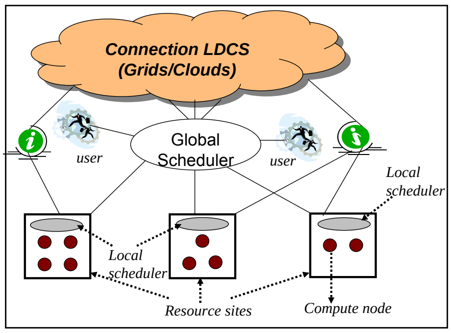


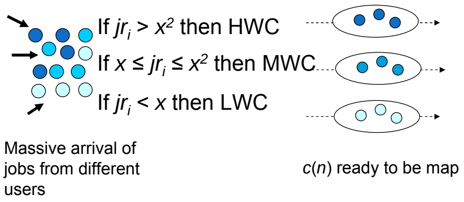
Figure 1. System Model

3. METHODOLOGY

- To address the problem of allocating resources to jobs in a dynamic fashion.
- 2 heuristic methods are involved :
  - (i) job-clustering and
  - (ii) mapping scheme.
- Integrate clustering and mapping into resource allocation decisions.
- Help organize mapping alternatives.
- Improve performance and availability

(i) Job clustering

Jobs group by jr_i and priority level, *pl* [1]. Given, *x* and x^2 are low *pl* and high *pl*, respectively. The jobs been grouped called *c*(*n*) where *n* is cluster name (i.e., **H**eavy, **M**edium or **L**ight **W**eight **C**luster).


If $jr_i > x^2$ then HWC

If $x \le jr_i \le x^2$ then MWC

If $jr_i < x$ then LWC

Massive arrival of jobs from different users

c(*n*) ready to be map

Each *c*(*n*) has :

- Different total jobs, $val_{c(n)}$.
- Total jr_i and total w_i, based on value of $val_{c(n)}$.
- Deadline factor $df_{c(n)}$ [1] from normalized deadline $nd_{c(n)}$.
- Closeness rate $crate_{c(n)}$; denotes 1/total differences of jr_i in *c*(*n*).

  (If $crate_{c(n)} \ge 0.1$ then high correlation; otherwise its low correlation).

Table 1. Job features in each *c*(*n*)

Cluster Name	Computational size of each job, w_i.	Deadline of each job, d_i.	Deadline factor, $df_{c(n)}$.
LWC	Small	Long	≤ 0.25
MWC	Medium	Moderate	≤ 0.50
HWC	Large	Short	≤ 1.00

(ii) Task-resource mapping

Each *c(n)* associated with its computational worth; given as total jr_i x $df_{c(n)}$. The task needs to be mapped onto an *Available* node regardless of site autonomy.

```
If (waiting time ≤ average execution time) then
    Resource is set to Available
Else
    Resource is set to Not_Available
```

For mapping event, the fitness value is used that defined as worth/ETC, where ETC = (total w_i / p_j).

```
If $crate_{c(n)}$ is high correlation then
    maps to $j$ that gives highest fitness value
If $crate_{c(n)}$ is low correlation then
    maps to $j$ that gives second highest
    fitness value
```

Advantage :

No reservation technique applied due to no preference is imposed for high, medium and low priority *c(n)* during mapping process.

4. RESULTS

Response Time

- It is observed that response time using our strategy (*Fit-Clustering*) is faster than other schemes; in [2] and [3].
- As we used fitness value; even if *c*(*n*) has low worth it may still be picked first if ETC is smaller.

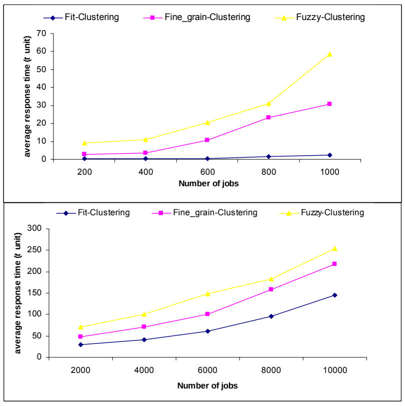


Figure 2. Average response time in moderate and large number of jobs

Utilization Rate

- In utilization rate, *Fit-Clustering* performs better, especially in large number of jobs by 10% on average.

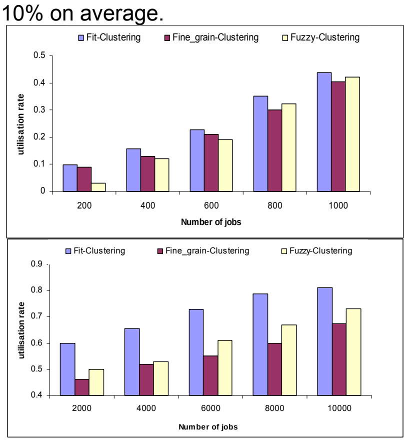


Figure 3. Utilization rate in moderate and large number of jobs

5. CONCLUSION

- We addressed the adaptive resource allocation problem with dynamically group of independent jobs; to paves the way in making LDCS efficiently usable for various types of jobs.
- Our clustering strategy effectively exploits both the diversity of jobs and heterogeneity of resources.
- Apparently, well-understandable properties in each group of jobs are important and worthy to investigate for judicious allocation.

REFERENCES

[1] J.-K. Kim, S. Shivle, H. J. Siegal *et al.*, “Dynamically mapping tasks with priorities and multiple deadline in a heterogeneous environment,” *J. Parallel and Distributed Computing,* vol. 67, no. 2, pp. 154-169, 2007.

[2] N. Muthuvelu, J. Liu, N. L. Soe *et al.*, “A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids,” in Australasian Workshop on Grid Computing and e-Research (AusGrid2005), 2005, pp. 41-48.

[3] F. Li, and D. Qi, “Research on Grid Resource Allocation Algorithm Based on Fuzzy Clustering,” *the 2nd Int Conf on Future Generation Communication and Networking*, pp. 162-166, 2008.