

ABSTRACT

Next-generation computation-intensive applications in various domains feature large-scale computing workflows that are often modeled as directed acyclic graphs and deployed in distributed network environments.

With limited computing resources, it is often unavoidable to map multiple modules to the same computer node which results in a resource contention among concurrently executing independent modules. Considering the implications of such resource sharing in workflow mapping algorithms is crucial for improving the end-to-end performance of these distributed applications. We conduct a deep investigation into workflow execution dynamics under a fair share based resource sharing system and propose Recursive Critical Path with Fair Share- RCPFS algorithm. We will compare its performance with the existing RCP algorithm by using an extensive set of simulated data and comparing the resulting minimum end-to-end MED values. RCPFS will provide a lower value of MED as compared to RCP by including the exact resource sharing dynamics in workflow mapping.

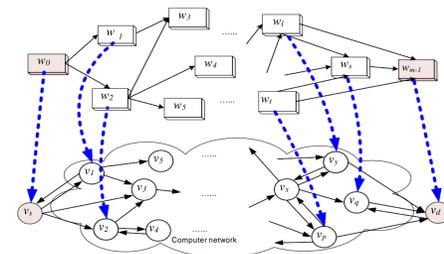
1. BACKGROUND

- Applications processing huge amounts of data that varies in volume, velocity, variety and veracity (also known as Big Data) are gaining an industry wide importance.[1, 2]
- Such applications are considered as a workflow of inter-dependent computing modules represented as a directed acyclic graph and mapped on a heterogeneous network.[3, 4, 5]
- The objective of such mapping is to select a set of computing nodes in the physical computer network and assign each computing module in the virtual task graph to one of the selected nodes to achieve minimum end to end delay (MED).
- The Efficient Linear pipeline configuration- ELPC algorithm is a dynamic programming based approach which models the computing workflow as a linear pipeline, partitions the modules of the task graph into groups and maps them onto a set of selected computing nodes in a network. [6] (Fig 4)
- The recursive critical path-RCP algorithm, improves upon ELPC algorithm by considering the task graph as a directed acyclic graph instead of a pipeline, which gives more realistic representation. [7] (Fig. 1, Fig. 5)
- When multiple modules get mapped to a common node they can compete for the processing power and bandwidth of the communication link. (Fig. 2)
- Including the implications of such resource sharing in the mapping algorithm can improve the MED value obtained from the mapping. (Fig. 3)

2. METHODS

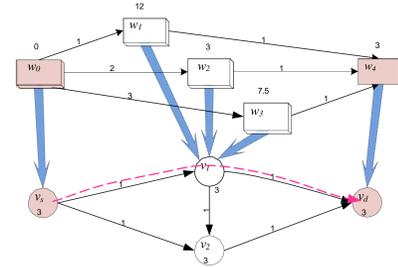
- We utilized the task and network graphs used in the RCP algorithm and generated simulation datasets by varying the number of modules, module complexity, network nodes, node processing power and network bandwidth between nodes.
- We implemented the Fair share based RCPFS algorithm to calculate the exact amount of processing power and bandwidth received by a module when mapped on a node with multiple other modules.
- We modified the RCP algorithm with above calculations for the mapping of the non-critical modules on heterogeneous network.
- We will run the modified datasets over the RCP and RCPFS algorithms and compare the MED values. (Fig. 6)

Figure 1



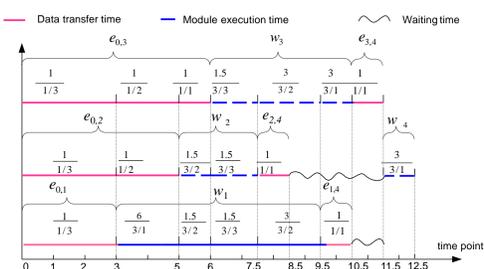
General DAG structured Workflow mapping
Taken From Q. Wu and Y. Gu, 2008 [7]

Figure 2



A simple workflow mapping example with resource sharing
Taken From Y. Gu, Q. Wu, and N. S. Rao, 2011 [8]

Figure 3

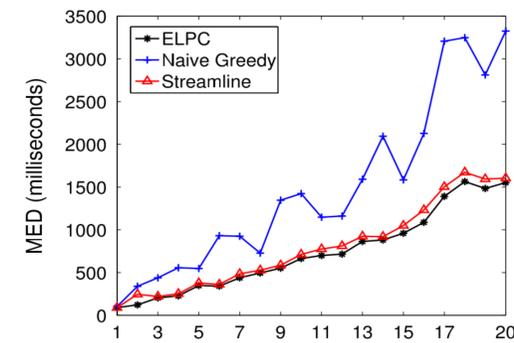


Time series analysis of resource sharing dynamics
Taken From Y. Gu, Q. Wu, and N. S. Rao, 2011 [8]

3. RESULTS

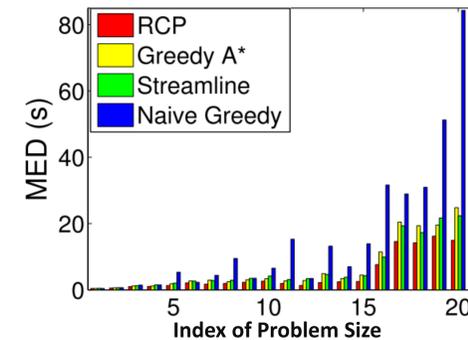
- We anticipate that the RCPFS algorithm will provide an improved value of MED over the RCP algorithm for cases when resource sharing occurs. (Fig. 6)
- RCPFS will provide a substantial performance speedup over the RCP algorithm. (Fig. 7)
- Small standard deviation values in the MED demonstrates the performance robustness and optimization stability of RCPFS in achieving MED in various task graphs and computer networks of disparate topologies and different scales. (Fig. 8)

Figure 4



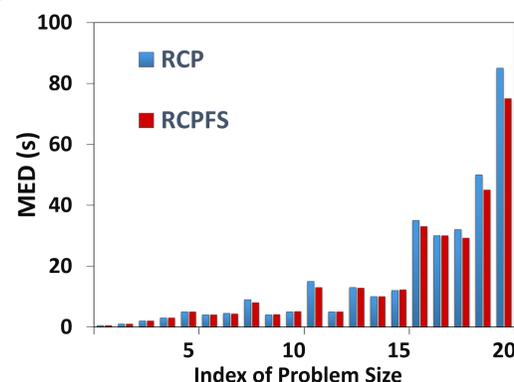
MED value comparison of ELPC, Naive Greedy and Streamline
Taken from Q. Wu, Y. Gu and N. S. Rao, 2011 [6]

Figure 5



MED comparison-RCP, Greedy A*, Streamline and Naive Greedy
Taken From Q. Wu and Y. Gu, 2008 [7]

Figure 6

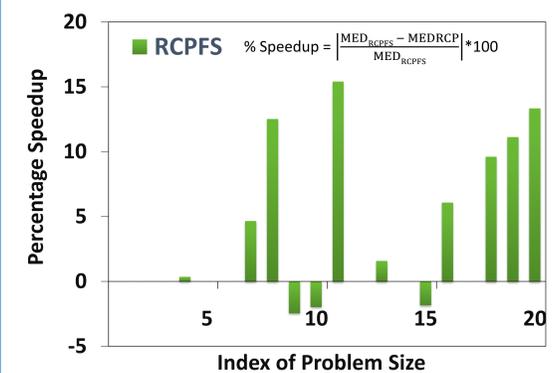


Predicted MED comparison-RCP and RCPFS
Figure is generated using Mock Data

4. CONCLUSIONS AND FUTURE WORK

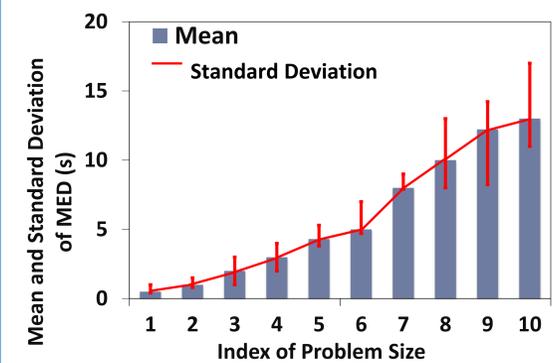
- Performance analysis results will prove that RCPFS algorithm provides an improved MED value over RCP algorithm for resource sharing scenarios. (Fig. 6, Fig. 7)
- The future work will focus on designing a priority based dynamic resource sharing algorithm that will give more resources to critical modules over branch modules. [9]

Figure 7



Predicted percentage speedup in MED value of RCPFS over RCP
Figure is generated using Mock Data

Figure 8



Predicted Mean and Standard Deviation of MED in sec
Figure is generated using Mock Data

REFERENCES

- [1] D. Jewell, R. D. Barros, S. Diederichs, L. M. Duijvestijn, M. Hammersley, A. Hazra, C. Holban, Y. Li, O. Osaigbovo, A. Plach, I. Portilla, M. Saptarshi, H. P. Seena, E. Stahl, and C. Zolotarev, "Performance and Capacity Implications for Big Data," *International Business Machines Corporation*, pp. 1–36, 2014.
- [2] H. Han, W. Yonggang, C. Tat-Seng, and L. Xuelong, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," *Access, IEEE*, vol. 2, pp. 652–687, 2014.
- [3] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. Su, K. Vahi and M. Livny, "Pegasus: mapping scientific workflows onto the grid," *Proc. of the European Across Grids Conference*, pp. 11–20, 2004.
- [4] K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *CSUR ACM Comput. Surv. ACM Computing Surveys*, vol. 31, no. 4, pp. 406–471, Jan. 1999.
- [5] F. Afrati, C. Papadimitriou, and G. Papageorgiou, "Scheduling dags to minimize time and communication," *Proc. Of the 3rd Aegean Workshop on Computing: VLSI Algorithms and Architectures*, Springer-Verlag, pp. 134–138, 1988.
- [6] Q. Wu, Y. Gu, M. Zhu, and N. S. Rao, "Optimizing network performance of computing pipelines in distributed environments," *2008 IEEE International Symposium on Parallel and Distributed Processing*, 2008.
- [7] Q. Wu and Y. Gu, "Supporting Distributed Application Workflows in Heterogeneous Computing Environments," *2008 14th IEEE International Conference on Parallel and Distributed Systems*, 2008.
- [8] Y. Gu, Q. Wu, and N. S. Rao, "Analyzing Execution Dynamics of Scientific Workflows for Latency Minimization in Resource Sharing Environments," *2011 IEEE World Congress on Services*, 2011.
- [9] Q. Wu, D. Yun, X. Lin, Y. Gu, W. Lin, and Y. Liu, "On Workflow Scheduling for End-to-End Performance Optimization in Distributed Network Environments" *Job Scheduling Strategies for Parallel Processing Lecture Notes in Computer Science*, pp. 76–95, 2013.

ACKNOWLEDGEMENTS

This research is sponsored by National Science Foundation under Grant No. 536958 with Middle Tennessee State University.