Journal of Advanced Computing Research Vol. 2, Issue 1 (2016) 24-28

A New Efficient Algorithm based on Divide and Conquer Strategies Algorithm for Scheduling Concurrent Tasks on Grid Computing

Nur Farah Hanim binti Rosli [#1], Mohamed Faidz Mohamed Said [#2]

[#] *Universiti Teknologi MARA*
70300 Seremban Negeri Sembilan, MALAYSIA
[1] nurfarahrosli28@gmail.com
2 faidzms@ieee.org

***Abstract*—There are research ideas which are based on a game theoretic model of divide-and conquer in developing some pioneering work. First, leader will can extract surplus from the responders and improve his private payoff, even though it can decrease total surplus in society because some surplus is destroyed in the process and all can do when transgression was succeeded. In this study to analyzing complex simulation models the design of experiment techniques (DOE) was used because it can offer valuable technique such as solutions to overcome resource limitations caused by computational costs by decreasing the range of possible experimental settings to the most important combinations.**

***Keywords:* algorithm efficiency, divide and conquer, grid computing**

I. Introduction

Algorithm efficiency is a good algorithm that can work as long as it could. Two or more algorithms that solve the same problem can be very different and still satisfy these two criteria. Therefore, the step is to identify which algorithm is "best". These two criteria used to determine whether one algorithm is "good" than another by looking whether it can settle its task in reasonable amount of time and it must run.

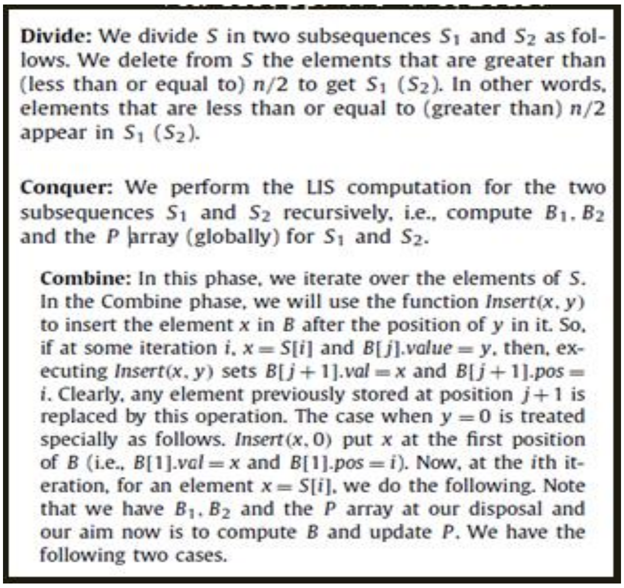
A. *Divide and conquer:* Divide is from big problem into a number of subproblems that are smaller instances of the same problem. Conquer is the subproblem by solving them recursively. If they are small enough, solve the subproblem as base cases.

B. *Grid computing:* Grid computing is distributed architecture of large numbers of computers connected to solve a difficult problem. In this system model, servers or individual computers run independent tasks are loosely linked by the Internet or low-speed network. Computers may be linked directly or via scheduling systems.

II. Review Papers

This paper is all about a new efficient algorithm based on divide and conquer strategies algorithm for scheduling concurrent tasks on Grid computing.

A. *A divide and conquer approach and a work-optimal parallel algorithm for the LIS problem*



Divide: We divide S in two subsequences S_1 and S_2 as follows. We delete from S the elements that are greater than (less than or equal to) $n/2$ to get S_1 (S_2). In other words, elements that are less than or equal to (greater than) $n/2$ appear in S_1 (S_2).

Conquer: We perform the LIS computation for the two subsequences S_1 and S_2 recursively, i.e., compute B_1, B_2 and the P array (globally) for S_1 and S_2.

Combine: In this phase, we iterate over the elements of S. In the Combine phase, we will use the function *Insert*(x, y) to insert the element x in B after the position of y in it. So, if at some iteration i, $x = S[i]$ and $B[j].value = y$, then, executing *Insert*(x, y) sets $B[j+1].val = x$ and $B[j+1].pos = i$. Clearly, any element previously stored at position $j+1$ is replaced by this operation. The case when $y = 0$ is treated specially as follows. *Insert*$(x, 0)$ put x at the first position of B (i.e., $B[1].val = x$ and $B[1].pos = i$). Now, at the ith iteration, for an element $x = S[i]$, we do the following. Note that we have B_1, B_2 and the P array at our disposal and our aim now is to compute B and update P. We have the following two cases.

Figure 1

In this paper, it presents a divide and conquer method to solve the problem of computing a longest increasing subsequence. The main idea is how we can achieve work optimally using parallel algorithm D&C method.

B. *A Design of Automatic Visualizations for Divide-and-Conquer Algorithms*

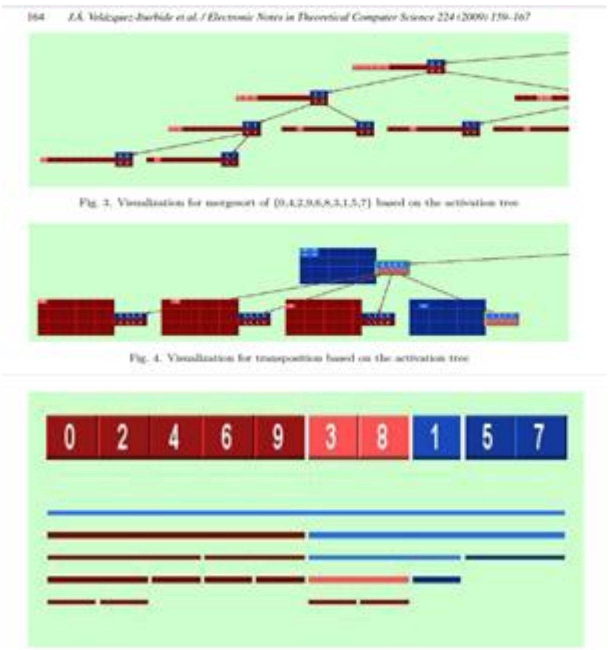

164 J.Á. Velázquez-Iturbide et al. / Electronic Notes in Theoretical Computer Science 224 (2009) 159–167
Fig. 3. Visualization for mergesort of {0,4,2,9,6,8,3,1,5,7} based on the activation tree
Fig. 4. Visualization for transposition based on the activation tree
0
2
4
6
9
3
8
1
5
7

Figure 2

This paper is focused on the design of program visualizations adequate to represent divide-and-conquer technique and algorithm.

C. DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed

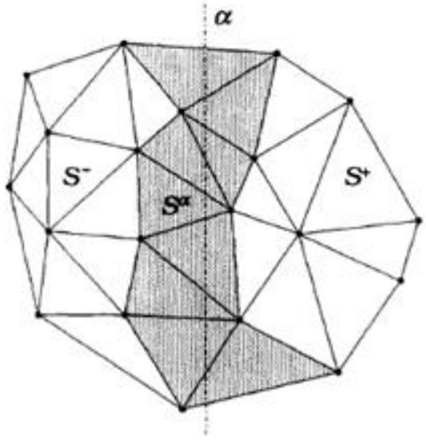


Figure 3

This is paper about divide and conquers deals with Delaunay Triangulations (DT) in Ed space. This classic computational geometry problem, so divide and conquer paradigm as a new solution. The main reason it can be simply extended to triangulate point sets in any dimension. Generating synchronization statements is done in divide-and-conquer programs.

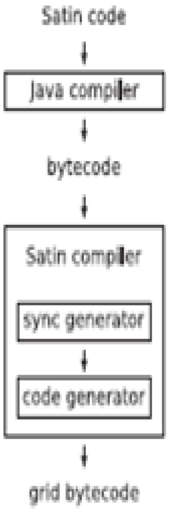


Figure 4

This paper shows generated synchronization divide and-conquer systems such as Satin, and recursive calls are automatically transformed into jobs that execute asynchronously. The main is Satin provides a divide-and-conquer programming model to execute applications efficiently on clouds and Grids computing.

D. A divide-and-conquer approach to compute collision cross sections in the projection approximation method

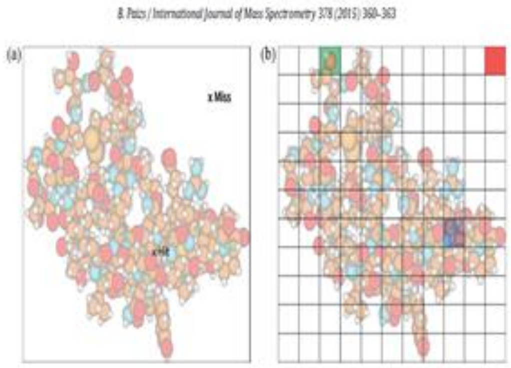

B. Pais / International Journal of Mass Spectrometry 378 (2015) 360-363
(a)
(b)
x Miss
x Hit

Figure 5

This paper is a new strategy to compute these projections based on a divide-and-conquer (DC) strategy. The main work of a new divide-and-conquer algorithm is to evaluate collision cross sections in the PA framework and efficiently calculate molecular projections of big biomolecules.

E. An efficient divide-and-conquer approach for big data analytics in machine-to-machine communication

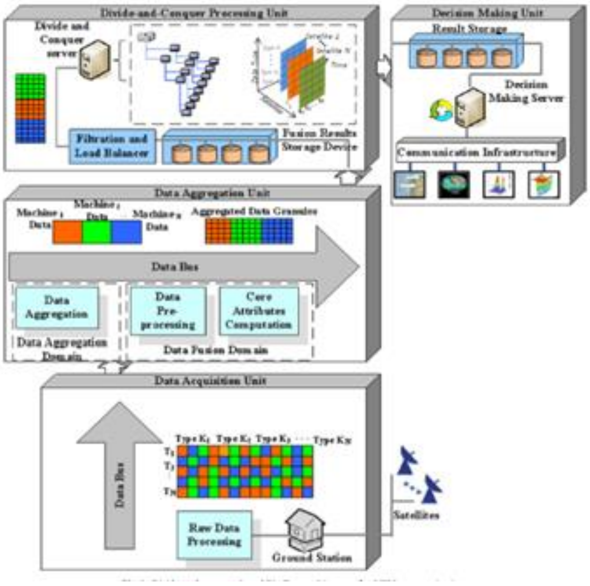


Figure 6

This paper is machine-to-machine communication (M2M). The main content is processing server follows the technique of the divide-and-conquer mechanism, when each server analyzes the data and then conquers the data to produce desired results.

F. *Divide-and-conquer based summarization framework for extracting affective video content*

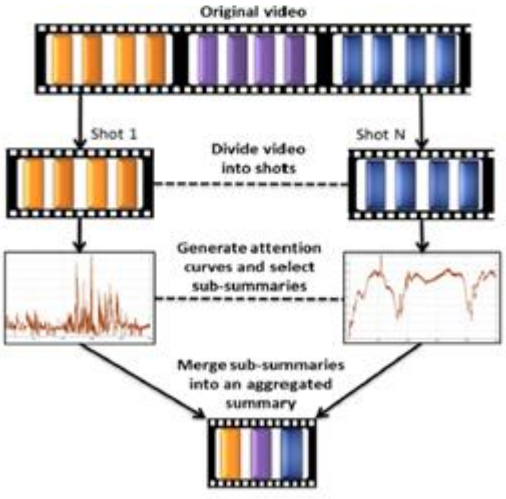

Original video
Shot 1
Divide video into shots
Shot N
Generate attention curves and select sub-summaries
Merge sub-summaries into an aggregated summary

Figure 7

This paper is video summarization to be more interesting. It is proposed that shot-detection based divide-and-conquer strategy alleviate the time and computational complicated problem. It can also compute aural, visual, and EEG attention features to extract the affective key frames from video sequences.

G. *Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory*

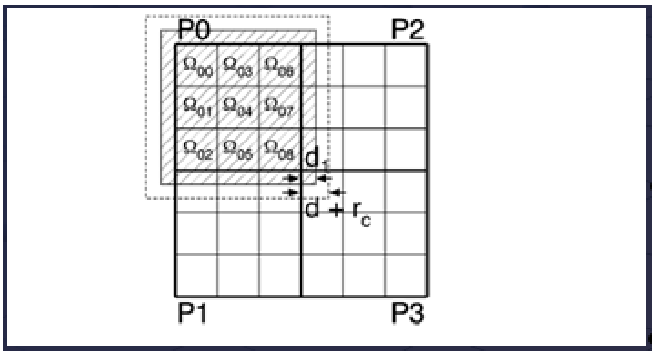


Figure 8

This paper is a linear-scaling algorithm to compute the electronic structure and a divide-and-conquer algorithm was used where non-additive offering to the kinetic energy is included with an implant cluster scheme.

H. *Divide And Conquer Strategies For Parallel TSP Heuristics*

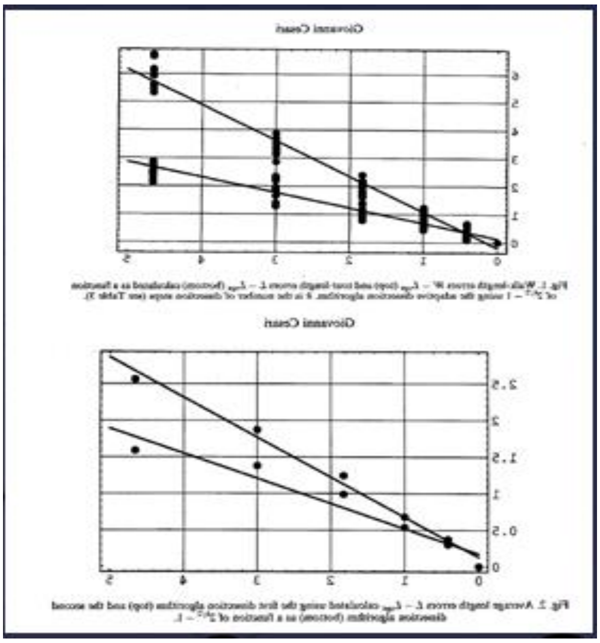


Figure 9

This paper is about geometric Traveling Salesman Problem (TSP). The aim of this paper is to study experimentally divide and conquer strategies, used to implement parallel heuristics for the geometric Traveling Salesman Problem (TSP).

I. *Adaptive Workflow Scheduling In Grid Computing Based On Dynamic Resource Availability*

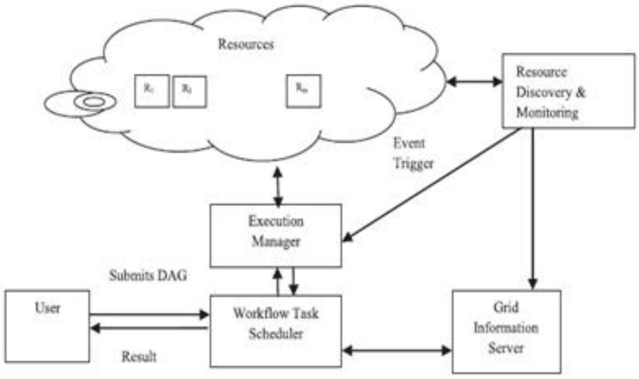


Figure 10

This paper is related to Grid computing. The solution to address the dynamic nature of Grid environment and to efficiently utilize the resources based on QoS information like availability along with the accessibility as indicated by SLA.

J. *Divide-and-conquer Mapping of Parallel Programs onto Hypercube Computers*

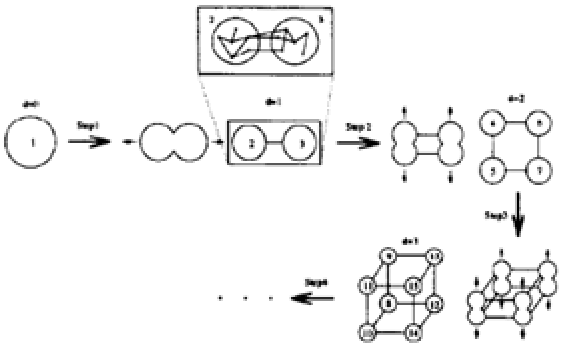


Figure 11

This paper shows plotting of parallel programs into parallel computers to execute with efficiency. Using the divide and conquer clustering by repeatedly splitting the task graph until the number of clusters achieved the number processors of a target hypercube computer.

K. *Automation Of Clock Distribution Network Design For Digital Integrated Circuits Using Divide And Conquer Technique*

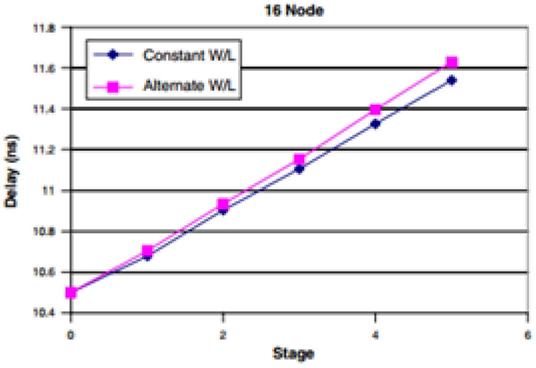


Figure 12

This paper presents automation of clock distribution network design. The new algorithm developed uses the divide and conquer technique on Grid files in solving the problem. Grid files are data structures used in database systems for storing data with multiple attributes.

L. *Reverse Scheduling-an Effective Method For Scheduling Tasks Of Parallel Programs Employing A Divide-and-conquer Strategy Onto Multiprocessors*

This paper involves the use of a parallel program. Consider the problem of scheduling tree structured task graphs onto many processor systems. Such graphs represent parallel programs which employ a divide and-conquer plan.

M. Optimality of HLF for scheduling divide-and-conquer UET task graphs on identical parallel processors

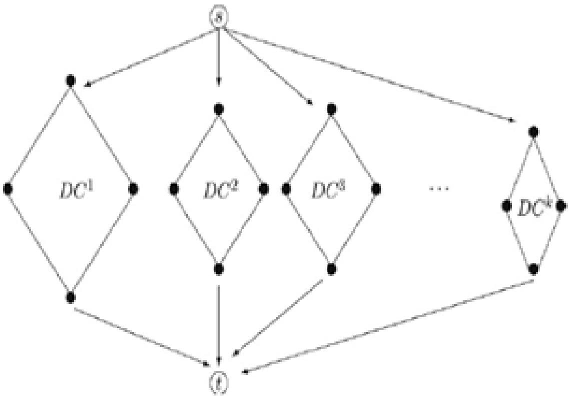


Figure 13

This paper considers the divide-and-conquer class of graphs. This shows that divide-and-conquer graphs naturally model the execution of the recursive control concept of divide-and-conquer algorithms. It is well known as several problems advanced them naturally to the divide-and-conquer solution strategy.

III. METHODOLOGY

Simulation methods can be seen as a difficult experimental environment that can give some advantages from a preliminary phase. In line with lean management's philosophy of doing the things right the first time and both of the sub and full models are supporting a thorough planning and preparation of simulation experiments.

IV. CONCLUSION

This study has explained the a new efficient algorithm based on divide and conquer strategies algorithm for scheduling concurrent tasks on Grid computing. Furthermore, divide and conquer (D&C) is an algorithm design paradigm based on multi-branched recursion in computer science that can simply solve the difficult problem to become the ease one. The problem can be minimized by using D&C method and can be proved by mathematical induction.

REFERENCES

[1] A. AHMAD, A. PAUL, AND M. M. RATHORE, "AN EFFICIENT DIVIDE-AND-CONQUER APPROACH FOR BIG DATA ANALYTICS IN MACHINE-TO-MACHINE COMMUNICATION," *NEUROCOMPUTING,* VOL. 174, PP. 439-453, 2016.

[2] K. N. B. M. t. a. C. S. R. M. A Sreenivas, "Reverse scheduling- an effective method for scheduling tasks of parallel programs employing a divide-andconquer strategy onto multiprocessors," *Microprocessors and Microsystems,* vol. 18.

[3] M. R. Alam and M. S. Rahman, "A divide and conquer approach and a work-optimal parallel algorithm for the LIS problem," *Information Processing Letters,* vol. 113, pp. 470-476, 2013.

[4] T. N. Cason and V.-L. Mui, "Rich communication, social motivations, and coordinated resistance against divide-and-conquer: A laboratory investigation," *European Journal of Political Economy,* vol. 37, pp. 146-159, 2015.

[5] G. Cesari, "Divide and Conquer Strategies for Parallel TSP Heuristics," *Computers Ops Res.,* vol. 23, pp. 681-694, 1996.

[6] R. Garg and A. K. Singh, "Adaptive workflow scheduling in grid computing based on dynamic resource availability," *Engineering Science and Technology, an International Journal,* vol. 18, pp. 256-269, 2015.

[7] A. Gupta and B. Kulkarni, "Automation of clock distribution network design for digital integrated circuits using divide and conquer technique," *Integration, the VLSI Journal,* vol. 39, pp. 407-419, 2006.

[8] P. Hijma, R. V. van Nieuwpoort, C. J. H. Jacobs, and H. E. Bal, "Generating synchronization statements in divide-and-conquer programs," *Parallel Computing,* vol. 38, pp. 75-89, 2012.

[9] W. Kubiak, D. Rebaine, and C. Potts, "Optimality of HLF for scheduling divide-and-conquer UET task graphs on identical parallel processors," *Discrete Optimization,* vol. 6, pp. 79-91, 2009.

[10] I. Lorscheid and M. Meyer, "Divide and conquer: Configuring submodels for valid and efficient analyses of complex simulation models," *Ecological Modelling,* vol. 326, pp. 152-161, 2016.

[11] I. Mehmood, M. Sajjad, S. Rho, and S. W. Baik, "Divide-and-conquer based summarization framework for extracting affective video content," *Neurocomputing,* vol. 174, pp. 393-403, 2016.

[12] C. Notes, "Algorithm Efficiency," ed, 2000.

[13] C. M. a. R. S. P Cignonit, "DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed," *Computer-Aided Design,* vol. 30, pp. 333-341, 1998.

[14] B. Paizs, "A divide-and-conquer approach to compute collision cross sections in the projection approximation method," *International Journal of Mass Spectrometry,* vol. 378, pp. 360-363, 2015.

[15] N. M. Reda, A. Tawfik, M. A. Marzok, and S. M. Khamis, "Sort-Mid tasks scheduling algorithm in grid computing," *J Adv Res,* vol. 6, pp. 987-93, Nov 2015.

[16] M. Rouse. (2016). *Grid Computing*. Available: http://searchdatacenter.techtarget.com/definition/grid-computing

[17] H. S. Sam Lor , Piyush Maheshwari, "Divide-and-conquer mapping of parallel programs onto hypercube computers " *Journal of Systems Architecture 43,* p. 373, 1997.

[18] F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta, "Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory," *Computer Physics Communications,* vol. 167, pp. 151-164, 2005.

[19] D. B. thomas Cormen. (2016). *divide and conquer algorithms*. Available: https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms

[20] J. Á. Velázquez-Iturbide, A. Pérez-Carrasco, and J. Urquiza-Fuentes, "A Design of Automatic Visualizations for Divide-and-Conquer Algorithms," *Electronic Notes in Theoretical Computer Science,* vol. 224, pp. 159-167, 2009.