Algorithm of Performance Prediction by Resource Sharing in Distributed Database

S. Jagannatha
M.S.Ramaiah Institute of Technology
Dept of Computer Applications
Bangalore-54, India

T.V.Suresh Kumar
M.S.Ramaiah Institute of Technology
Dept of Computer Applications
Bangalore-54, India

Rajanikanth
M.S.Ramaiah Institute of Technology
Former Principal
Bangalore-54, India

ABSTRACT
Resource allocation is one of the main issues in solving database applications where resources and data fragments are distributed geographically. The query of each use case assigned into resources to solve parallel computing problems and avoid remote data access. Hence system resources have to be allocated to handle workload and minimize the cost of computing and maximize the utility of resources. In this paper, it is propose an algorithm for optimal allocation strategy that minimizes the cost of computation by predict the performance. The overall goal is to minimize the cost of allocated resources usage in distributed database system during early stages. We propose game theoretic approach for finding the optimum allocation strategy which determines the performance during the early stages of life cycle.

General Terms
Resource Utilization, UML model

Keywords
Resource allocations, Distributed Database, Performance Engineering

1. INTRODUCTION
A distributed database (DDS) consists of set of fragments which are distributed across the sites of a computer network. A distributed database management system (DDBMS) is software that manages the e distributed database system. Distributed processing is an effective way to improve the performance of a database system. Improving the performance is one of the key research issues by proper design of efficient distributed database and proper usage of resource in information technology areas now days. Distribution of data is a collection of fragmentation, allocation and replication processes [1]. Developing distributed database systems by fragment design, fragment allocation, and resource allocation based on query requirement become more challenging. Performance can be improved by localization of data fragment, and avoid remote data accessed and by reducing the data exchange among sites. Database design involves making decisions on the placement of fragments across the sites of a computer network. Data fragments are stored in different data centers which are connected in communication network. Collections of fragment data can be distributed across multiple physical locations of different data centers.

Distributed Design involves making decisions on the fragmentation based on the requirement and placement of data across the sites of a computer network [2]. Data locality of a query minimizes the data transfer by different nodes. The higher the data locality by improve the performance of the system. Query based resource allocation, fragment design, dynamic allocation algorithm is feasible to performance goals in early design stages. Non functional requirement are to be predict at early stages is important consideration.

Unified Modeling Language (UML) is modeling for object oriented system. It helps you specify, visualize, and document models object oriented software systems. Performance models can be generated from use cases during requirements phase, sequence diagrams during analysis and design phase and from state chart and activity diagrams during design phase. UML 2.0 supports for stereotype which supports for performance modeling

Game theory is used to solve the problem of resource allocation for executing query in distributed database system. The majority of requests received by system used for data retrieving [3]. In this paper it is propose an algorithm for optimal allocation of resources in order to reduce the cost of data transfer in distributed environment. Estimate the cost of each use case which is modeled in UML.

The rest of the paper is organized as follows. In Section 2, we describe related work in resource allocation and distributed database design and in Section 3, we propose the methodology and algorithm of resource allocation scheme in distributed system. In section 4, we have illustrated the problem model with a case study for performance studies using simulation, and analyze the performance of the proposed resource allocation strategy for various schemes under different resource usage. Finally concluding remarks and future studies in Section 5.

2. RELATED WORK
Various researchers made significant contributions towards the area of resource allocation in distributed database system. In [4] author proposes model that fragment allocation based on transaction behavior in distributed database system. The model uses near-optimal allocation algorithm for allocation of fragments, transaction information, two heuristic algorithms is for reduces communication cost.. In [5] the author describes a distributed solution which integrates workload prediction and distributed non-linear optimization techniques. In [6] authors describe on-demand access from computing resources, which enable application providers scaling their services. This work attempts to establish formal measurements for under and over provisioning of virtualized resources. In [7] proposed an algorithm of job scheduling based on Berger model. The algorithm establishes dual fairness constraint. The general expectation functions in accordance with the classification of tasks. The second constraint is resource fairness justice function to judge the fairness of the resources allocation. In [8] this paper authors attempts to establish formal

measurements for provisioning of virtualized resources under and over in infrastructures, specifically for Software as Service platform deployments. The author proposes a resource allocation model to deploy applications over computing platforms by taking into account their multi tenancy for creating a cost-effective scalable environment.

In [9], author proposes policy that allocates the resources if available, otherwise the request is rejected and request is placed in a FIFO queue. It uses as resource leases as resource allocation abstraction and implements these leases by allocating Virtual Machines. Proposed dynamic planning based scheduling algorithm is implemented in that can admit new leases and prepare the schedule whenever a new lease can be accommodated and it maximizes resource utilization and acceptance of leases. In this [10] paper author proposed an approach that combines proprietary cloud based load balancing techniques and density-based partitioning query processing across relational database as a service in cloud computing environments conducted over a real-world data. The proposed approach implemented and tested as a multi layer web application and database layers. The author focuses in [11] constructing non dominated local coteries to solve the problem in a distributed way and it reduces communication overheads. The algorithm achieves the highest degree of fault-tolerance in resource allocation problem. In [12] resource allocation mechanism based on reverse auction and proposes a market model based on reverse auction and presents the bidding process of reverse auction and it provides optimal resource provider for each consumer.

In [13] automatic resource allocation strategy based on market mechanism is proposed. Market model of ARAS-M are constructed, in requirements of Client. A Genetic Algorithm based automatic price adjusting algorithm is introduced. Experiments results show that approximately achieve the equilibrium state and are capable of achieving resource balance in resource usage. In [14] this paper, study the resource allocation at the application for better resource utilization and author proposed a threshold-based dynamic resource allocation scheme that dynamically allocate the virtual resources among the load changes and can use the threshold method to optimize the decision of resource reallocation. Allocation and reallocation techniques for resources in a distributed database (DDB) system, which partitions data among a set of nodes in a network. It is proposes multiple token borrowing polices, which anticipate future need and keep the system balanced.

In all the approaches, the researchers use different techniques and methodologies are used in allocation of data fragments in distributed system. Some authors address the allocation and reallocation based on transaction behavior both static and dynamic technique was discussed. Effective utilization of resources in computing and allocation of resources in parallel computing applications were addressed by some authors. Authors do not consider the performance predictions by optimum allocation strategy of resources based on query requirements where data fragment exist in distributed system in early stages of life cycle. Keeping these in view it is propose an algorithm to optimum allocation strategy in effective utilization of resources by predict the performance of distributed database system.

3. PROPOSED METHODOLOGY AND ALGORITHM

3.1 Methodology

The performance of Distributed database is strongly related to the fragment allocation in the nodes of computer network. A query requires data to be accessed from local or remote sites. The cost of computing the query depends on the fragment allocation and the query allocation of the resources. The goal is to minimize the total data transfer, resource usage cost for transaction processing. The cost is to minimize by proper allocation of resources, data availability in distributed environment [15][16][17]

3.2 Proposed Methodology.

Suppose that n be the number of use cases share m resources. Each use case U_i consisting k parallel and dependent sub queries, for example: use case with-draw has withdraw, check balance, overdraw, and are parallel sub queries. These tasks are sharing R_j resources and share the R_j capacity and expense P_j according to its processing capacity. Each sub query may have to share R_j and may be waiting to share R_j. The assigning these queries into these resources based on the data fragments available. A solution of the scheduling problem is a non-negative matrix i.e. query allocation matrix a of n rows one for each use case and m columns one for each resource. The entry aij is the allocation of query in the Ui use case to the R_j resources. Using allocation matrix another two matrices are obtained: Completion time matrix T, and final execution matrix E. Let tij of T be the turnaround time it takes resource R_j to complete aij queries of the use caseUi. Ui queries are parallel and dependent, the completion time of use case Ui is given by $max\,\{tij \mid tij \in ti\}$, where ti denotes the vector of the i^{th} row of matrix T. The entry Eij of the matrix E is the network cost for using resource Rj to complete aij queries. So the total cost of usage of U_i is $\sum_{j=1}^{m} eij$

In general there is a tradeoff between completion time and cost of usage of resources of each use case U_i. Let w_t and w_e be the weight of completion time and network cost respectively. Therefore the Total cost of

$$U = \text{Ui} = wt \cdot maxtij * ti\,\{tij\} + we * \textstyle\sum j\, eij.$$

$1/Ui$ denote the utility of use case Ui .

The procedure to estimate the cost in early design stages.

1. The key scenarios of the software system are identified
2. The use cases representing the key scenarios are identified to develop the use case model using UML.
3. Let m be the no of resources(R) are used in the proposed architecture.
4. Identify the subtask in each use case share m resources(nodes)
5. Each resource R_j has fixed price P_j according to the capacity.
6. Identify the vector $p = (p1, p2, .., pm)$ satisfies $p1 < p2 < \cdot \cdot \cdot < pm$, and the corresponding execution time of any subtask of an arbitrary task S_i satisfies $\hat{}ti1 > \hat{}ti2 > \cdot \cdot > \hat{}tim$.
7. Identify the multiple sub queries in each use case, they share R_j capacity and expense.

8. Propose the non-negative matrix a(allocation matrix) of n rows for a set of use case and m columns, one for each resource.
9. Let $\hat{t}ij$ denote the execution time it takes for resource R_j solely to complete one subtask of a use case U_i without consideration of communication time.
10. Let a_i represents the amount of sub query allocated to the resource R_j.
11. Derived from a another two $n\, x\, m$ matrix are obtained
    i. Compute complétion time matrix T. The entry tij of T is the turnaround time it takes for Resource R_j to Complete aij sub queries of a use case Ui i.e. $tij = \sum j\ aij * \hat{t}ij$ where $\hat{t}ij$ execution matrix
    ii. Expense matrix E. The entry eij of E is the expense Ui pays for Resource R_j to Complete ai_j sub queries, i.e. $eij = aij * \hat{t}ij * pj$ when only one sub task assigns to one resource
12. Let w_t and w_e denote the weights of completion time and expense, respectively.
    It is assumed $wt = we = 0.5;$
13. Therefore, the total "cost" of a query is U_i is
    $wt\ \cdot maxtij * \ ti\,\{tij\} + we * \sum j\ eij.$
14. Compute utility of task
    $ui\,(ai) = 1/\,(wt * maxtij * ti\,\{tij\} + we * \sum j\ eij.)$
15. Repeat step 9 through 12 for a different allocation strategy until it reaches the optimum allocation i.e. least cost by predicts the performance.

3.3 Algorithm

The algorithm for the proposed methodology.

Identify key performance scenarios of a given applications

Let M be the number of resources (nodes) of a given architecture.
Each resource R_j has uncertainty waiting time, Each resource has fixed price P according to the capacity.
for all scenarios S do
  Develop use case model
  Identify the set of use case N
  Identify the multiple sub queries, they share R_j capacity.
  Compute probable execution time matrix $\hat{t}ij\ (N\ x\ M)$
  Identify all subtasks of each use case.
  for each allocation a do
    Propose allocation matrix $a(N\ x\ M)$ where the queries are allocated to the M resources
    Compute $tij = \sum j\ aij * \hat{t}ij$ (completion time matrix)
    Compute Expense Matrix E i.e. $Eij = aij * \hat{t}ij * pj$
    Compute total cost of a use case
    $Ui = wt * maxtij * ti\,\{tij\} + we * \sum j\ eij.$
    Compute utility of use case
    $ui(ai) = 1/\,(wt * maxtij * ti\,\{tij\} + we * \sum j\ eij.)$
  end for (Until performance goal)
end for.

4. ILLUSTRATED THE PROBLEM MODEL WITH CASE STUDY

In this section we proposed a case study, Airline reservation application [18][19] that are highly distributed in nature for apply the proposed methodology, to illustrate and validate. The prediction of performance for this problem is discussed using different approach in [18][19]. The required data are fragmented and distributed in various nodes of the system. The objective is to reduce the cost of computation. Thus compare the cost of execution of various allocation strategies with probable execution time matrix. It also obtained the system utility of each use case from the proposed methodology. The performance metrics taken from [3] .

4.1 Description of the Case Study.

Nachtfliegen airline system has major functions: Flight Booking, Login, Abandon, Get Flight planning, Find Flights, Select Seat, Get Fare, Purchase Itinerary, and Store Itinerary, The database is fragmented and deployed in a given architecture of 8 nodes. All servers connected by a LAN. The details can be found in [18] the typical requests from the users for the application may be authenticating the user, getting the page of the application, searching for the appropriate flights, selecting the desired seat, purchase the ticket, store the details about the flight for later reference, and abandon the Itinerary. The use cases have taken in[18], actors identified for the scenarios are presented in the use case model in Figure 1.

4.2 Estimation of the cost matrix of a given Use case

In this section, it is proposed to illustrate the methodology for the case study. The objective is to select optimum allocation strategy by reducing the cost of computation of use case by maximum utilization of server. It is proposed to estimate (i) find allocation strategy (ii) compute turnaround time (iii) compute Expense matrix E (iv) compute total cost of each query (v) Compute utility of resources by each query. Let w_t and w_e denote the weights of completion time and expense, respectively. It has been assumed that $w_t = w_e = 0.5$. Optimum allocation strategy is obtained by different runs and using probable execution time matrix.

The use case model given in Figure 1 consists of use cases namely, Login, GetFlightPlanningPage, Plan Itinerary, Get Fares, Select Segment, Select Seat, Purchase Itinerary, Store Itinerary and Abandon(U1 - U7) the actors namely, customer, user and frequent flyer. The terminals of the nodes are connected to their respective computers

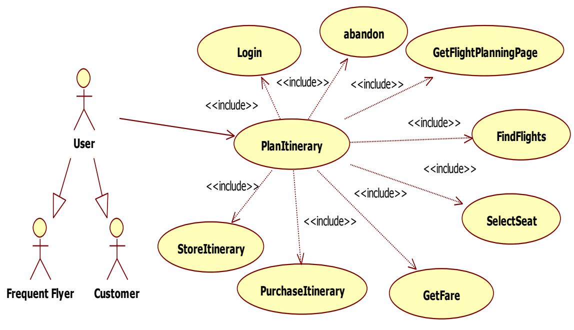

Fig 1: Use case model of the case study

5. RESULTS AND GRAPHS

The price vector of each resources(R1 - R8) is ranging from 1 to 10 by taking low level is 1 middle level is 5 and high level is 10, then the execution matrix i.e. final execution time matrix, completion time matrix and expenses matrix is obtained in table 1, table 3 and table 4 respectively. The price vector, the available resources (R1 – R8) and their configuration as Intel Core 2 Duo, 3.0 GHz are considered for computation. The description of use cases (U1 – U8) as described above.

Table 1: The execution time matrix

9	8	7	6	5	4	3	2
8	7	6	5	4	3	2	1
7	6.5	5	5.5	5	4.5	4.0	3.5
6	5.5	5	4.5	4	3.5	3	2.5
5	4.5	4	3.5	3.0	2.5	2.0	1.5
4.5	4	3.5	3.0	2.5	2.0	1.5	1.0
4.2	4.0	3.8	3.5	3.2	2.0	1.5	1.0
3.5	3.2	3.0	2.5	2.2	2.0	1.8	1.5

Table 2 Allocation matrix

1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	1
1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

The use case U1 chooses { R1, R2, R3, R4,} and U2 chooses {R3, R4, R5, R6, R7, R8} U3 chooses (R3, R5, R6, R7, R8) U4 chooses (R1, R2, R4, R7, R8) U5 chooses (R1, R2, R3, R4, R5, R6, R7, R8) U6 chooses (R1, R2, R5, R6, R7, R8) U7 chooses (, R3, R4, R5, R6, R7, R8) and U8 chooses (R3, R4, R5, R6, R7, R8) as mentioned in the allocation matrix is in table 2. The use cases contains sub tasks share the capacity and expense of allocated resources. The task of each use case share the resources based on the data availability. The data is distributed based on the requirement of the applications. The resources are sharing by parallel tasks. The completion and expense matrix are obtained and the results are shown in table 3 and table 4 respectively.

Table 3 The completion time matrix

36	32	36	36	0	0	0	0
0	0	0	30	24	18	14	7
0	0	25	0	30	27	28	24.5
24	22	0	27	0	0	21	17.5
20	18	20	21	18	15	14	10.5
18	16	0	0	15	12	10.5	7.00
0	0	19	21	19.2	18	19.6	7.5
0	0	15	15	13.2	12	12.6	10.5

Table 4: The expense matrix/network usage matrix

9	16	21	30	0	0	0	0
0	0	0	25	8	3	14	8
0	0	15	0	10	4.5	28	28
6	11	0	22	0	0	21	20
5	9	12	17	6	25	14	12
4.5	8	0	0	5	20	10.5	8
0	0	11	17.5	6.4	3.0	19.6	20
0	0	9	12.5	4.4	2.0	12.6	12

The total cost of each use case and its utilities mentioned in table5. We observed from the shaded part in table 5 that the cost of using resources by the use case U3 is highest compared to others use case and the corresponding utility of resources is least. Hence we conclude that more and more utility of resources effectively the cost of paying is least and vice versa. In use case U6 least cost but the utility of resources is highest not properly utilized. Hence we deduce from the results that more and more usage of resources the expense of network usage is least and we obtain optimum allocation strategy.

Table 5: Utility of a given matrix

Use case	Cost of each Use case	Utility of Resource
U1	56.00045	0.017857
U2	44.00052	0.022727
U3	57.75006	0.017316
U4	53.74899	0.018605
U5	49.50005	0.020202
U6	28.00022	0.035714
U7	49.45109	0.020222
U8	33.74958	0.029630

For illustrative purpose we have taken seven price vector P in Table 5. Each resources has taken different values which are uniformly distributed with the minimum value 1 average 5 and the maximum range is 9, the corresponding utility of resources and the cost of each use case mentioned in table 7 and table 8 respectively. The shaded portion of these table shows that higher the utility of resources, performance in terms of cost is less cost and vice versa. The results shows the optimum allocation strategy can choose by initial design stages by predicting the performance

Table 6: Price vector of seven different inputs of Resources

	P1	P2	P3	P4	P5	P6	P7	P8
Case 1	1	2	3	3.5	4	4.5	5	6
Case 2	2	2.2	3	3.5	4	4.2	5	6
Case 3	4	5	3	6	8	9	1	2
Case 4	8	6	5	4	7	2	3	8
Case 5	6	5	4	2	8	9	4	1
Case 6	6	9	4	3	1	7	9	5
Case 7	5	6	3	4	9	8	5	4

Table 7: Utility of Resources

Use case	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
U1	0.01379	0.017204	0.011429	0.008889	0.008097	0.009091	0.008333
U2	0.019560	0.018198	0.009390	0.007722	0.018018	0.007782	0.008163
U3	0.017660	0.013741	0.022472	0.007576	0.02667	0.008475	0.011696
U4	0.040000	0.039024	0.011940	0.009662	0.006780	0.010204	0.019048
U5	0.025907	0.027548	0.021482	0.014493	0.012195	0.013072	0.006135
U6	0.034843	0.026059	0.028902	0.020000	0.022727	0.009390	0.015625
U7	0.029369	0.043764	0.022198	0.008230	0.011696	0.009852	0.013333
U8	0.026846	0.041017	0.028736	0.011429	0.014815	0.00950	0.010363

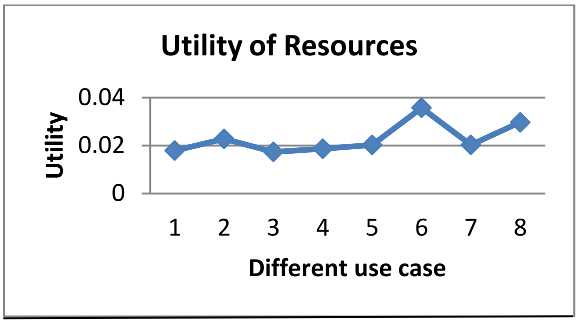

Fig 2 : Utility of Resources

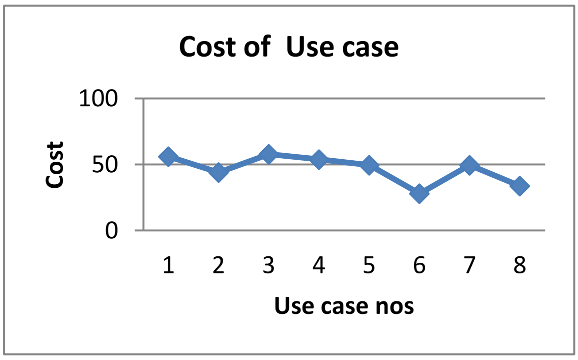

Fig 3: Cost of Use case

In the fig 2 shows the Utility of resources of eight use cases. The cost of Use case 1 utilizes 0.017857 Use case 2 utilizes 0.022727 etc. as shown in table 5. The utility of resources in use case 6 is high and least cost with allocation matrix as in fig 3. The results show that the utility of

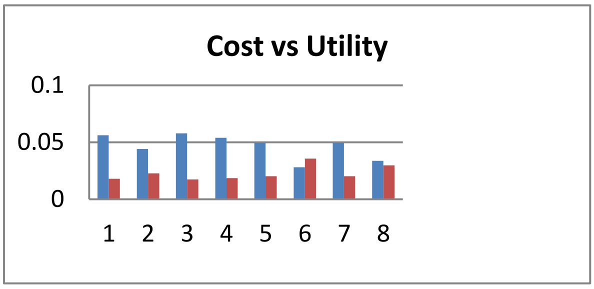

Fig 4 : Cost Vs Utility of resources

resources increases the cost is proportionally decreases and vice versa as shown in the fig 2 and fig 3. Cost vs. utility, price vector of resources are represented in fig 4. It is observed that in the third case the utility of resources increases the cost of using resources is least. Hence user has to select optimum allocation strategy for better performance. In the fig 5 the price vector of seven different case represented the correspond utility of resources is depicted in the table 7.

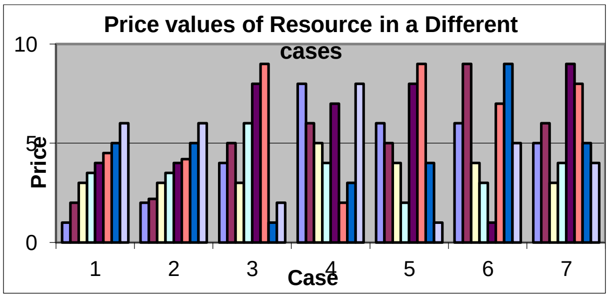

Fig 5: Shows the price values of resources in seven different cases

6. CONCLUSION

In this paper we use resource allocation strategy in minimizing the cost of resources in computing early design stages. We propose an algorithm for optimal allocation strategy that maximum utilization of resources and minimize the cost computation. Data fragments and allocated into these set resources in the proposed architecture. The fragment allocation and resource allocation strategy is considered in minimize the cost computation.

FUTURE SCOPE

Further system can be enhance using software execution model and system execution model of performance engineering technique for predict the performance in distributed database system during early design stages.

7. ACKNOWLEDGMENTS

Our thanks to the T .V Suresh Kumar, E Geetha, and Rajanikanth for guidance and contributed towards development of the paper.

8. REFERENCES

[1] Bente Anda, Hege Dreiem, Dag, I. K., Sjobergand Magne Jorgensen: 'Estimating Software development Effort based on Use Cases – Experiences from Industry',www.idi.ntnu.no/emner/tdt4290/docs/faglig/uml2001-anda.pdf.Accessed January 2007.

[2] Michele Mazzuccoa,b, Dmytro DyachukcA Optimizing Cloud providers revenues via energy efficient server allocation 2210-5379/$ – see front matter © 2011 Published by Elsevier Inc. doi:10.1016/j.suscom.2011.11.001Ss

[3] GuiyiWei · Athanasius V. Vasilakos · Yao Zheng · Naixue Xiong A game-theoretic method of fair resource allocation for cloud computing services Published online: 29 July 2009 © Springer Science+Business Media, LLC 2009

[4] Shahidul Islam Khan Dr. A. S. M. Latiful Hoque " A New Technique for database Fragmentation in Distributed Systems International Journal of Computer Applications (0975 – 8887) Volume 5– No.9, August 2010

[5] Danilo Ardagnaa, Sara Casolari b, Michele Colajanni b, Barbara Panicucci a,cDual time-scale distributed capacity allocation and load redirect algorithms for cloud systems J. Parallel Distrib. Comput. 72 (2012) 796–808

[6] Javier David Conchaa A tenant-based resource allocation model for scaling Software-as Espadas a, Arturo Molina b, Guillermo Jiménez a, Martín Molina b, Raúl Ramírez a, -a-Service applications over cloud computing infrastructures 0167-739X/$ – see front matter © 2011 Elsevier B.V. All *rights reserved. doi:10.1016/j.future.2011.10.013A.*

[7] Baomin Xu a,⇑, Chunyan Zhao b, Enzhao Hua, Bin Hu c,dJob scheduling algorithm based on Berger model in cloud environment 0965-9978/$ - see front matter 2011 Elsevier Ltd. All rights reserved. doi:10.1016/j.advengsoft.2011.03.007

[8] Javier Espadas a, Arturo Molina b, Guillermo Jiménez a, Martín Molina b, Raúl Ramírez a, David Conchaa A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved. doi:10.1016/j.future.2011.10.013

[9] Amit Nathani a,1, Sanjay Chaudharya,2, Gaurav Somanib,3 Policy based resource allocation in IaaS cloud 0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved. doi:10.1016/j.future.2011.05.016

[10] Adnene Guabtni · Rajiv Ranjan · Fethi A. RabhiA workload-driven approach to database query processing in the cloud Springer Science+Business Media, LLC 2011

[11] Jehn-Ruey Jiang Nondominated local coteries for resource allocation in grids and clouds 0020-0190/$ – see front matter © 2011 Elsevier B.V. All rights reserved. doi:10.1016/j.ipl.2011.01.008 S

[12] Xiaohong Wu∗, Yonggen Gu, Jie Tao Cloud computing resource allocation mechanism research based on reverse auction ESEP 2011: 9-10 December 2011, Singapore

[13] Xindong You, Jian Wan, Xianghua Xu, Congfeng Jiang, Wei Zhang, Jilin Zhang

[14] ARAS-M: Automatic Resource Allocation Strategy based on Market Mechanism in Cloud Computing JOURNAL OF COMPUTERS, VOL. 6, NO. 7, JULY 2011

[15] Bente Anda, Hege Dreiem, Dag, I. K., Sjobergand Magne Jorgensen: 'Estimating Software development Effort based on Use Cases – Experiences from Industry', www.idi.ntnu.no/emner/tdt4290/docs/faglig/uml2001-anda.pdf.Accessed January 2007.

[16] Michele Mazzuccoa,b, Dmytro DyachukcA Optimizing Cloud providers revenues via energy efficient server allocation 2210-5379/$ – see front matter © 2011 Published by Elsevier Inc. doi:10.1016/j.suscom.2011.11.001Ss

[17] GuiyiWei · Athanasios V. Vasilakos · Yao Zheng · Naixue Xiong A game-theoretic method of fair resource allocation for cloud computing services Published online: 29 July 2009 © Springer Science+Business Media, LLC 2009

[18] Evangelin Geetha, D., Suresh Kumar, T. V., and Rajani Kanth, K.: 'Predicting the Software Performance during Feasibility Study

[19] Connie, U. Smith, and Lloyd G. Williams: 'Performance Solutions' (Addison-Wesley, 2000).

[20] ARAS-M: Automatic Resource Allocation Strategy based on Market Mechanism in Cloud Computing JOURNAL OF COMPUTERS, VOL. 6, NO. 7, JULY 2011

[21] Weiwei Lina, James Z. Wangb, Chen Liangc, Deyu Qia A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing 1877-7058 © 2011 Published by Elsevier Ltd.doi:10.1016/j.proeng.2011.11.2568

[22] Yin-Fu Huang and Juh-her Chen Fragment Allocation in Distributed Database Design Journal of Information Science and Engineering 17, 491-506 (2001)

[23] S Jagannatha, D Evangelin Geetha, TV Suresh Kumar, Rajani Kanth Fragmentation of Distributed Database in

Healthcare System Using UML 2.0, 2009/2/1, Advances In Data Management49Macmilla

[24] S Jagannatha, M Mrunalini, TV Suresh Kumar, Rajani Kanth Modeling of Horizontal Fragmentation in Distributed Database using Health Care System Proceedings Second International Conference On Information Processing,284, 2008IK International Pvt Ltd.

[25] S Jagannatha, TV Suresh Kumar, DE Geetha, K Rajani Kanth, Assessment of Workload Using Shapely Value in Distributed Database, Proceedings of International Conference on Advances in Computing,31-40,Publisher, Springer India, 2012.