Tender of computational works in heterogeneous distributed environment

A G Feoktistov[1]

[1]Matrosov Institute for System Dynamics and Control Theory of SB RAS,
Lermontov St. 134, Irkutsk, Russia, 664033

agf@icc.ru

Abstract. Nowadays, applying various market-based methods for regulating supply and demand of resources for distributed computing is becoming increasingly relevant. In particular, different forms of standard auctions are actively used. However, their basic capabilities often do not enable to fully solve the complicated problems of resource allocation in a heterogeneous distributed computing environment. In this regard, a tender of computational works based on a combinatorial Vickrey auction has been designed. It is applied within multi-agent computing management. For the tender, new models are proposed to rank the criteria of resources owners and users. The tender use advantages are shown in comparison with traditional meta-schedulers.

1. Introduction

The current trend of turning computing resources into goods actualizes designing and applying economic mechanisms for regulating supply and demand of such resources. This is especially pronounced within cloud computing.

Traditionally, various auctions and tenders are widely used to regulate supply and demand for goods and services in practice.

An auction is a form of public sale of any objects (lots), for example, goods and services, according to predefined rules. Owners of lots are sellers. Bidding is based on competition between potential purchasers of these goods and services. A person that has received the right to purchase a lot in accordance with the auction rules becomes its winner.

A tender is a competitive form for selecting proposals for the supply of goods, provision of services, or fulfilment of work in accordance with predefined conditions. Herein, bidders are participants wished to perform services that are in demand by customers. Unlike auctions, where the only criterion is the lot cost, the tender enables us to determine additional conditions for the fulfilment of work. The winner of the tender is its bidder, whose bid contains the best proposals that meet the predefined conditions.

A tender may be designed in the form of an auction or a contest. In both cases, its bidders make bids for work.

In the first case, the bid distribution is carried out according to the results of bidding (comparison of aggregated parameters of bids, the process of which is easily formalized and automated). There is a large spectrum of solutions for holding standard auctions.

In the second case, additional complex stages are required, including the condition development for the contest, contest committee organization, and expert evaluation of bids.

Obviously, an auction is the simplest form of bidding. This is due to less complexity and lower overheads in its design compared to a contest.

In the paper, a tender in which the end-user of a heterogeneous distributed computing environment acts as a customer is proposed. A customer presents computational work (a job). Service sellers (agents) are representatives of computational resources of the environment. Sellers claim to perform a submitted work.

The remainder of the paper is organized as follows. A brief overview of approaches related to the paper topic is given in the next section. Section 3 describes the computing environment under consideration. A problem of multi-agent computing management is formulated in Section 4. A combinatorial Vickrey auction is described in Section 5. Section 6 provides new models for matching owner preferences of the resource use and user criteria in solving their problems. The results of the computational experiments are shown in Section 7. They have demonstrated the advantages of the proposed multi-agent management. The last section concludes the paper.

2. Related work

The results of comparative analysis [1-5] of various auction models show that standard auctions are the most effective, discussed, and widely applicable in practice [6]. Among them are English (forward or reverse) auction [7], first-price sealed-bid auction [8], Dutch auction [9], and second-price sealed-bid auction [10]. The principles of their operation are presented in detail in [11, 12].

Each of the four above-listed standard auctions gives the auctioned lot to the participant who appreciates it the most. At all four auctions, the bidding result is selected from a variety of Pareto-optimal solutions. However, Vickrey and English auctions, which implement the dominant strategies, are more effective. This means that no new proposals from other bidders can change the final situation [13].

In the real world, an English auction is one of the most popular in practice. It is often used in computing systems for resource allocation (see, for example, [14]) when executing both independent jobs and pre-formed job packages. Those and others are considered as indivisible lots.

However, additional overheads occur in resource allocating for problem-solving schemes (workflows) in an English auction. These overheads are associated with the organization of individual bidding for each subjob of the scheme and need in multi-criteria selecting optimal deadlines for carrying out rounds of bidding.

Such deadlines are determined in order to ensure equal conditions in bids waiting and responses sending for all bidders. Participants may be at different distances from the auctioneer. Moreover, their interconnects may have different bandwidth and load.

In addition, these deadlines should lead to quick decision-making for resource allocation.

Another problem is taking into account the relations between the subjobs. Sometimes, a situation arises when the cost in executing the subjobs set may differ from the cost in executing these subjobs separately. Often, this problem is unsolvable in fully within an English auction.

Within a Vickrey auction, bidders have equal rights. They act according to uniform rules and make sealed (unknown to other bidders) bids for the lot. All bids are reviewed by the auctioneer. The winner is the one who made the best bid. The winner receives the lot on conditions of the bid proposed by the participant, who turned out to be the second, according to the auction results.

Each bidder makes a bid, reflecting the true value of the lot and maximum usefulness for this participant. This is a feature of a Vickrey auction. The consensual steady state of auction participants is achieved at the end of the auction. The achieved state is an analogue of the Nash equilibrium in game-theoretic models [15]. This was shown in [16]. The example in applying the Vickrey auction model in a heterogeneous distributed computing environment is given in [17].

Algorithms for multi-objective resource selection in heterogeneous cloud computing environments are presented in [18]. They are based on the double auction conception [19]. Wherein, cloud resources (virtual machine instances) are characterized by the performance, sizes of RAM and disk storage, network bandwidth, and use cost. Users specify requirements to different sets of virtual machines. Usually, in such cases, a weighted price of the computational work execution is determined.

3. Environment

Cloud computing is a very important direction in preparing and carrying out large-scale scientific experiments. This is achieved by elastic providing computing resources on demand. Clouds provide processors, storage, interconnect, etc.

Large-scale scientific applications (distributed applied software packages) require applying computing environments that provide maximum use of heterogeneous resources to increase the computation speedup. Often, the resources of the cloud platforms themselves are not enough. Therefore, additional resources of computing clusters, grid-systems, and public access supercomputer centers are included in clouds.

Such resources belong to different owners. They are used by end-users of various categories with diverse purposes. In this regard, we have faced the need for a large trade-off between the efficient resource use and makespan minimization for the applications executed on these resources.

In the paper, abstract workflows (problem-solving schemes) are considered. They can be executed on different target resources.

To execute a problem-solving scheme, an end-user has to create a job for the run environment. A job specifies requirements to the environment (number of nodes and cores, sizes of RAM and disk memory, OS, supposed job makespan, etc.) that are necessary to solve the problem. Usually, jobs are formed automatically at packages in formats used by environment meta-schedulers (GridWay [20], Condor DAGMan [21], etc.).

The systems supported the package design and use, for example, Orlando Tools [22], provide mapping jobs to target resources based on methods for computation planning and resource allocation. Wherein, the selection of resources is a key issue to ensure the efficiency of their use and decrease job makespan. The proposed tender of computational works is designed within the Orlando Tools development.

4. Problem formulation

A computational model of the environment is presented by the following structure:

$$MOD =< MOD_a, MOD_s, O >,$$

$$MOD_a =< ASP, Z_a, F_a, M_a, S_a, J_a, Q_a >,$$

$$MOD_s =< SSP, Z_s, F_s, M_s, S_s, J_s, Q_s, R, A, P, H >,$$

where MOD_a and MOD_s are the structures that describe applied and system knowledge, respectively. ASP, Z_a, F_a, M_a, S_a, J_a, and Q_a are the sets of applied software packages, parameters, operations, program modules, problem-solving schemes, jobs, and constraints to their execution.

Operations from F_a determine computational actions on the set Z_a of parameters.

Parameters can be represented by scalars, vectors, and matrices of various basic data types or arbitrary data structures.

Each operation $f_i \in F_a$ is implemented by the module $m_j \in M_a$, where $i \in \overline{1, n_f}$, $j \in \overline{1, n_m}$, n_f is the number of operations, and n_m is the number of modules. One module can implement several operations.

Each operation f_i has two subsets $Z_i^{in}, Z_i^{out} \subset Z_a$ of parameters. The subset Z_i^{in} includes input parameters. Their values must be known in order to calculate values of parameters from Z_i^{out}. Parameters of the subsets Z_i^{in} and Z_i^{out} reflect the purpose and semantics of formal parameters of the module m_j that implements the operation f_i. Parameters are transferred between modules in the form of data files.

Schemes from S_a represent problems-solving processes in packages. They correspond workflows in workflow management systems.

A description of applied knowledge is represented in detail in [23].

The sets SSP, Z_s, F_s, M_s, S_s, and J_s from MOD_s have the similar purpose for system objects as the corresponding sets of applied objects from MOD_a. Q_s is the set of resource use criteria.

R, A, and P are the sets of resources, agents, and administrative policies determined for resources.

The data structure H represents the computational history that reflects the execution statistics for modules from M_a.

O determines relations between the above-listed objects from MOD_a and MOD_s.

The model MOD enables us to define the following multi-criteria problem formulation of the distributed computing management: Calculate $Y = \{y_1, y_2, \dots, y_{n\_out}\}$ knowing $X = \{x_1, x_2, \dots, x_{n\_in}\}$ by executing $F' = \{f_1, f_2, \dots, f_{n\_f}\}$ on resources $R' = \{r_1, r_2, \dots, r_{n\_r}\}$ taking into account $Q' = \{q'_1, q'_2, \dots, q'_{n\_s}\}$ and $Q'' = \{q''_1, q''_2, \dots q''_{n\_a}\}$, where $F' \subseteq F_a$, $R' \subseteq R$, $Q' \subseteq Q_s$, and $Q'' \subseteq Q_a$. The presence of various uncertainties in the problem conditions (in Y, X, F', C^u, C^o, R', and Q) leads to a variety of its additional specialized formulations.

In general, the criteria $q'_1, q'_2, \dots, q'_{n\_s}$ and $q''_1, q''_2, \dots q''_{n\_a}$ are the subjective preferences of resource owners and their end-users, respectively. They can impose conflicting restrictions on a problem-solving scheme.

In the paper, the cost, makespan, and reliability of a problem-solving scheme are considered in the role of end-user's quality criteria. The preferences of resource owners include the resource use efficiency, average processor load, and its balancing.

Let a problem-solving scheme be formed. For modules $m_1, m_2, \dots, m_{n\_m}$, which implement scheme operations, evaluations of module runtimes are determined. In addition, evaluations of data (parameters) size transferred between the modules are obtained.

Agents need to allocate their resources to execute the modules.

Computing process must satisfy the criteria $c_1, c_2, \dots, c_k \in \{q'_1, q'_2, \dots, q'_{n_s}, q''_1, q''_2, \dots q''_{n\_a}\}$. For each ith criterion, its upper and lower bounds $c_i^{min} \geq 0$ and $c_i^{max} \geq 0$ are determined, $c_i^{min} \leq c_i \leq c_i^{max}$, $i = \overline{1, k}$. Moreover, the following criteria optimality are defined: $c_i \rightarrow c_i^{min}(c_i^{max})$.

The job that is generated to execute the problem-solving scheme includes a set of subjobs. Subjobs can relate to different classes of jobs.

5. Combinatorial Vickrey auction

A seller income within an English auction may slightly exceed a corresponding income within a Vickrey auction.

Proposition 1. The difference in seller income within a Vickrey and forward English auction with the fixed value δ of the bid increase does not exceed this value.

Proof. Let v_1 and v_2 $(v_1 < v_2)$ are true lot worths for the agents a_1 and a_2, δ is the fixed value of the bid increase in bidding.

Then the agent a_2 becomes a winner with the bid $v_2^* = v_1^* + \delta$ within an English auction, where $v_2^* \leq v_2$, $v_1^* \leq v_1$ is the bid of the agent a_1 at which it will complete bidding. The final lot cost (seller income) is equal to v_2^*.

Within a Vickrey auction, agents truthfully reflect their true worths. The agent a_1 will complete bidding with the bid v_1. The agent a_2 becomes a winner with the bid v_2. In accordance with the rule of the second price, the final lot cost (seller income) is equal to v_1.

According to the bidding conditions, $v_1^* \leq v_1$. Then

$$v_2^* - v_1 \leq v_2^* {-} v_1^*. \tag{1}$$

Since $v_2^* = v_1^* + \delta$ then, in the result of the substitution of this expression into the right part of (1), we obtain $v_2^* - v_1 \leq \delta$■

The same is true for a reverse English auction in providing, for example, of services.

Proposition 2. The difference in the income of agents from the service provision within a Vickrey and forward English auction with the fixed value δ of the bid decrease does not exceed this value.

Proof. Let v_1 and v_2 $(v_2 < v_1)$ are true service worths for the agents a_1 and a_2, δ is the fixed value of the bid decrease in bidding.

Then the agent a_2 becomes a winner with the bid $v_2^* = v_1^* - \delta$, where $v_2^* \geq v_2$, $v_1^* \geq v_1$ is the bid of the agent a_1 at which it will complete bidding. The final service cost (income of the agent a_2) is equal to v_2^*.

Within Vickrey auction, agents truthfully reflect their true worths. The agent a_1 will complete bidding with the bid v_1. The agent a_2 becomes a winner with the bid v_2. In accordance with the rule of the second price, the final service cost (income of the agent a_2) is equal to v_1.

According to the bidding conditions, $v_1 \leq v_1^*$. Then

$$v_1 - v_2^* \leq v_1^* - v_2^*. \tag{2}$$

Since $v_2^* = v_1^* - \delta$, then, in the result of the substitution of this expression into the right part of (2), we obtain $v_1 - v_2^* \leq \delta$■

Within a combinatorial Vickrey auction, several lots can be simultaneously put up for bidding. A combinatorial Vickrey auction provides an opportunity to make offers (bids) for combinations of such lots [24]. An extensive bibliography confirms the effectiveness of applying various forms of such an auction in relation to distributed computing [25-33].

Moreover, in some cases, the combinatorial auction use gives an advantage over English auction. Let two agents a_1 and a_2 participate in bidding for the jobs j_1 and j_2 execution. The jobs process the same data set d. The job j_1 (j_2) execution cost summarizes the module execution cost v_1 (v_2) and data transfer cost v_d. True worths v_1, v_2, and v_d of executing the jobs j_1 and j_2 and transferring data d by the agent a_1 and a_2 are shown in Table 1.

Table 1. Resource allocation results based on the reverse English auction.

Agent	v_1^j	v^d	v_2^j	v^d	w_1^j	w_2^j	Final cost
a_1	1.7	1.0	1.2	1.0	+	–	4.8
a_2	1.8	1.0	1.1	1.0	–	+	

A reverse English is more consistent with the tender of computational works in comparison with a forward English auction. Based on this auction, agents make their bids decreasing them by the fixed value of 0.1. They stop betting, having reached their true worths.

In this example, the agent a_1 becomes the winner for the job j_1 execution with the final service cost equalled 2.7. The second agent a_2 becomes the winner for the job j_2 with the final service cost equalled 2.1. The total cost of two jobs execution is equal to 4.8. The bid offers order does not affect the final bidding result.

A combinatorial auction within which we can make a bid for the joint execution of several jobs enables us to improve the bidding result. It provides the reduction in overhead costs for data transfer when the jobs are executed by one agent.

Table 2 shows the resource allocation results based on the combinatorial auction, where additional bids appeared for the joint execution of several jobs by one agent. In this case, the final cost is 3.9. This cost has been proposed by both agents. Therefore, the winner will be determined by comparing the bids receipt time or taking into account additional criteria (problem-solving makespan, computing reliability, computation speedup, etc.).

Table 2. Resource allocation results based on the combinatorial auction.

Agent	v_1^j	v^d	v_2^j	v^d	$v_1^j + v_2^j + v^d$	w_1^j	w_2^j	$w_{1,2}^j$	Final cost
a_1	1.7	1.0	1.2	1.0	3.9	–	–	+	3.9
a_2	1.8	1.0	1.1	1.0	3.9	–	–	–	

Thus, a combinatorial Vickrey auction and English auction have advantages and drawbacks. However, the combinatorial nature of the multi-agent management problem leads to the selection of the first auction. At the same time, the main drawback of this auction, like all other types of combinatorial auctions, is the computational complexity of decision-making.

The following methods are used to mitigate this complexity:

- Preliminary classification of jobs [34]. It ensures the formation of a virtual community of agents (tender participants), each of which is sure in the presence of necessary resources and interested in executing jobs corresponding to them.
- Decomposition of a scheme for problem-solving that represented by a tiered-parallel form into subschemes, each including only one tier [35]. The tender for each subscheme is held separately. This enables agents to take into account the resource allocation for the previous tiers when processing the current tier.
- The use of parallel operations, providing the opportunity in proportional distributing of the computational load related to the execution of impersonal subjob instances. Thus, it eliminates the need in hard searching combinations of subjob instances when agents are betting.

Given the above-listed methods, the maximum number of possible combinations of subjobs at each tender round varies from one combination to several tens of combinations. This is the inherent property for both the scientific workflows (Montage, CyberShake, Epigenomics, etc. [36]) and schemes for solving many practical problems [35].

6. Models

To conduct and evaluate the bidding results, a virtual resource cost is used. In addition, resource owner preferences $q_1'', q_2'', ... q_{n\_a}''$ and user criteria $q_1'', q_2'', ... q_{n\_a}''$ of the problem-solving efficiency are applied. Resource cost characteristics are selected in such a way that the cost in computational work performing on a more performance resource cannot be higher than the cost in performing the same work on a less performance resource.

The tender is based on three models for matching owner preferences and user criteria:

1) Model oriented to user criteria, when the following ranking by an importance decrease is applying in bids reviewing: $q_1'', q_2'', ... q_{n\_a}'', q_1', q_2', ..., q_{n\_s}',$
2) Model oriented to owner preferences, when the following ranking by an importance decrease is applying in bids reviewing: $q_1', q_2', ..., q_{n_s}', q_1'', q_2'', ... q_{n\_a}'',$
3) Model, in which user criteria and owner preferences are not ranked.

Each agent makes bids for executing individual subjobs, the class of which corresponds to the resources of this agent. Elements of one bid reflect $q_1', q_2', ..., q_{n_s}', q_1'', q_2'', ... q_{n\_a}''$ for one subjob.

In addition, any agent can make bids on executing sets of jobs. Elements of such a bid reflect $q_1', q_2', ..., q_{n_s}', q_1'', q_2'', ... q_{n\_a}''$ for each job in the set.

To determine the best bids in Models 1 and Model 2, the lexicographic rule of multi-criteria selection is applied. Model 3 uses the Pareto-optimal selection of the best bids. The aspects of the aforementioned selection methods may be found in [37].

In the case of non-uniqueness of the decision, an additional bid evaluation is carried out. The selected bids are compared with the ideal bid using the Cartesian metric.

7. Experiments

The evaluation in operating a multi-agent system (MAS) in processing job flows through semi-natural modeling [38] for the three proposed models was carried out in an experimental environment that includes two resource pools:

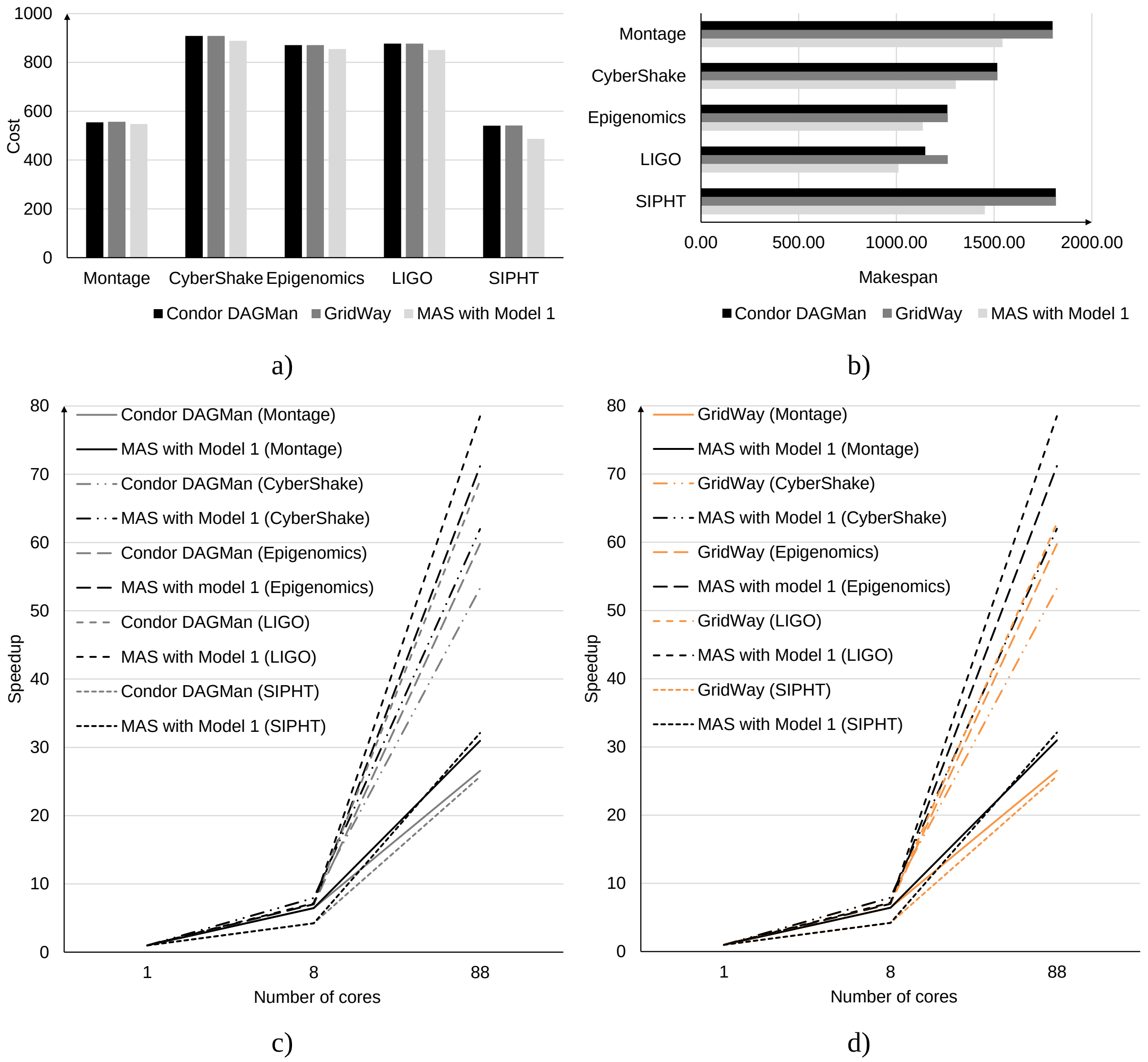
- Pool 1: 8 virtual machines with the following characteristics: 1 Intel Xeon processor E5506 (1 core, 2.13 GHz, 4 MB L3 cache, 2 GB RAM DDR3-800, 4 FLOP/cycle),
- Pool 2: 10 virtual machines with the following characteristics: 1 AMD Opteron 6276 processor (8 core, 2.3 GHz, 16 MB L3 cache, 32 GB RAM DDR3-1600, 4 FLOP/cycle).

The environment is based on resources of public access Irkutsk Supercomputer Center of the Siberian Branch of the Russian Academy of Sciences [39].

Jobs specify scientific workflows. Among them are Montage, CyberShake, Epigenomics, LIGO, and SIPHT. Each flow included 30 workflows of the same type.

Analysis of the MAS operation was fulfilled in comparison with the meta-schedulers Condor DAGMan and GridWay. The following criteria were selected for a comparative analysis of the quality in job flow processing: cost and makespan of job execution, computation speedup, resource use efficiency, average processor load, and standard deviation from it. The first three criteria represent user criteria. The other three criteria reflect the resource owner preferences. A standard deviation characterizes the resource load balancing.

Figure 1 shows that the MAS with Model 1 provides advantages in all user criteria for each job flow in comparison with both Condor DAGMan and GridWay. The management results with the Condor DAGMan and GridWay use are very close. At the same time, MAS reduces the cost and makespan of executing jobs and increases computation speedup. In particular, Figure 1(c) and Figure 1(d) show the speedup of computations under the MAS management in comparison with Condor DAGMan and GridWay, respectively. In contrast to Condor DAGMan and GridWay, the transfer of data directly between agents significantly influenced the achieved cost, makespan, and speedup. The aforementioned meta-schedulers transfer data through centralized storages. This leads to more overheads.



a)

b)

c)

d)

Figure 1. User criteria: cost (a) and makespan (b) of executing jobs and computation speedup (c).

Figure 2 shows that MAS with Model 2 provides the advantages in all criteria that reflect the resource owner preferences. The benefits are demonstrated for each job flow in comparison with both Condor DAGMan and GridWay. MAS increases the resource use efficiency and average processor load. In addition, it decreases the standard deviation σ from average processor load.

Using Model 3 with unranked criteria, MAS shows an improvement in more criteria than Condor DAGMan and GridWay.

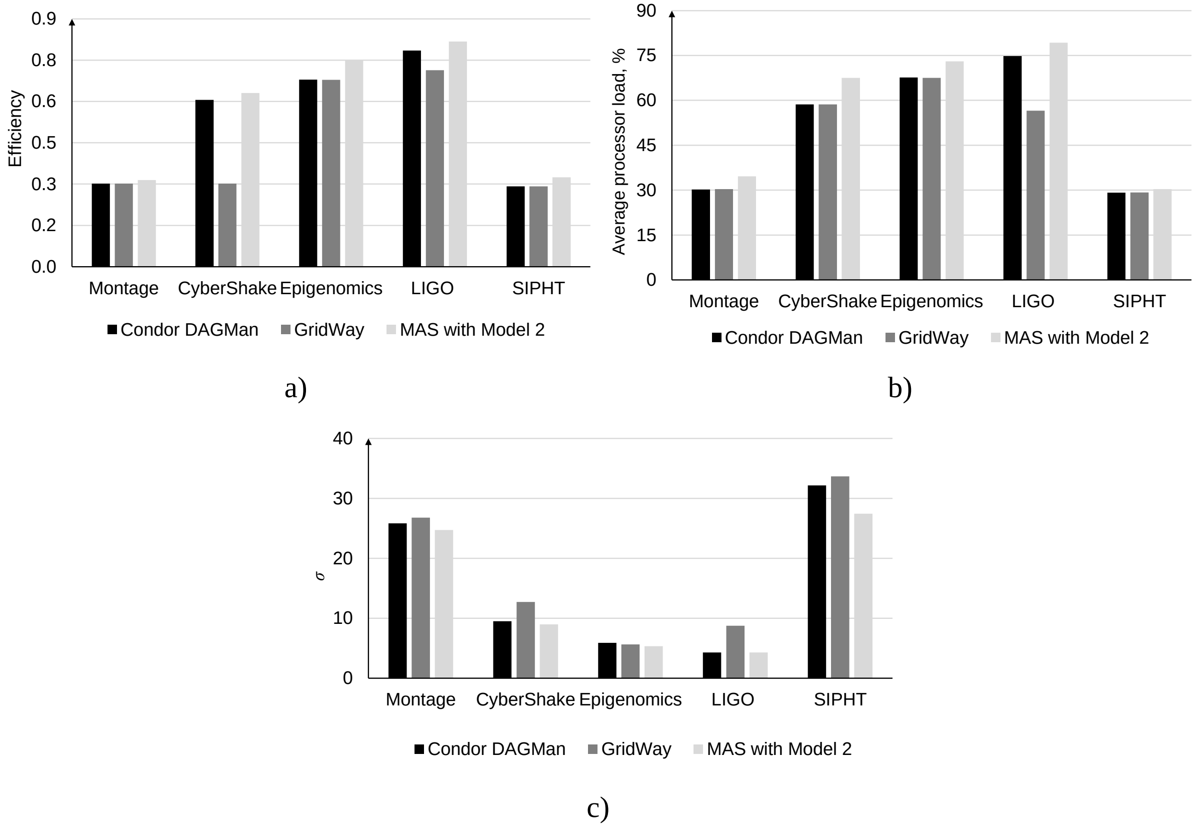


Figure 2. Owners preference in resource use: efficiency (a), average processor load (b), standard deviation σ from average processor load (c).

Thus, the advantages of the proposed multi-agent management over the traditional meta-schedulers are obvious. Each of the three models makes it possible to achieve the required level of service, depending on the way of ranking the criteria.

8. Conclusions

The paper addresses a relevant issue related to resource allocation in a heterogeneous distributed computing environment with multi-agent management. The environment can integrate resources of public access supercomputer centers, grid-systems, and cloud platforms. Agents represent environment resources and implement their allocation in solving problems of scientific applications (distributed applied software packages).

A new bidding mechanism is proposed to solve the resource allocation issue. The mechanism is implemented in the form of a tender of computational works. The tender is based on applying a combinatorial Vickrey auction. Similar mechanisms enable resource providers to effectively utilize their available resources and obtain higher income. In the paper, such mechanisms are expanded. Within the tender, resource owner preference and user criteria of problem-solving complement the only criterion (cost of computational works) of a Vickrey auction.

Using the tender, agents truthfully make their bids and quickly provide decision-making. The provided bidding mechanism is highly scalable, efficient and validated by comprehensive semi-natural modeling. The experiments were carried out through processing job flows for the well-known scientific workflows. The advantages of the proposed multi-agent management are demonstrated in comparison with the well-known traditional meta-schedulers for distributed computing environments.

The full theoretical rationale for the achievement of an agreed state by agents within the tender is the subject of further study.

Acknowledgments

The study is supported by the Basic Research Program of SB RAS, project no. IV.38.1.1.

References

[1] Wellman M P, Walsh W E, Wurman P R and MacKie-Mason J K 2001 Auction protocols for decentralized scheduling *Games Econ. Behav.* **35(1)** 271–303

[2] Ausubel L M and Milgrom P 2006 The lovely but lonely Vickrey auction *Combinatorial auctions* Cramton P, Shoham Y and Steinberg R (Cambridge: MIT Press) pp 22–26

[3] Buyya R, Abramson D, Giddy J and Stockinger H 2002 Economic Models for Resource Management and Scheduling in Grid Computing *Concurr. Comput.* **14(13-15)** 1507–1542

[4] Cramton P, Shoham Y and Steinberg R 2007 An overview of combinatorial auctions *ACM SIGecom Exchanges* **7(1)** 3–14

[5] Grosu D and Das A 2004 Auction–based resource allocation protocols in grids *Proc. of the 16th IASTED Intern. Conf. on Parallel and Distributed Computing and Systems* (Calgary: ACTA Press) pp 20–27.

[6] Coppinger V M, Smith V L and Titus J 1980 Incentives and behavior in English, Dutch and sealed-bid auctions *Econ. Inq.* **18(1)** 1–22

[7] Neeman Z 2003 The effectiveness of English auctions *Games Econ. Behav.* **43(2)** 214–238

[8] Harrison G W 1989 Theory and misbehavior of first-price auctions *Am. Econ. Rev.* **79(4)** 749–762

[9] Bagwell L S 1992 Dutch auction repurchases: An analysis of shareholder heterogeneity *J. Finance* **47(1)** 71–105

[10] Vickrey W 1961 Counterspeculation, Auctions, and Competitive Sealed Tenders *J. Finance* **16(1)** 8–37

[11] Cassady R JR 1967 *Auctions and Auctioneering* (Berkley and Los Angeles, California: University of California Press) p 327

[12] McAfee R P and McMillan J 1987 Auctions and bidding *J. Econ. Lit.* **25(2)** 699–738

[13] Sandholm T 2000 Issues in computational Vickrey auctions *Int. J. Electron. Comm.* **4(3)** 107–129

[14] Badica C, Ilie S, Muscar A, Badica A, Sandu L, Sbora R, Ganzha M and Paprzycki M 2015 Distributed agent-based online auction system Computing and Informatics. **33(3)** 518–552

[15] Blume A and Heidhues P 2004 All equilibria of the Vickrey auction *J. econ. Theory* **114(1)** 170–177

[16] Nash J 1950 Equilibrium points in n–person games *P. Natl. Acad. Sci. USA* **36(1)** 48–49

[17] Feoktistov A, Kostromin R, Sidorov I, Gorsky S and Oparin G 2019 Multi-Agent Algorithm for Re-Allocating Grid-Resources and Improving Fault-Tolerance of Problem-Solving Processes *Procedia Comput. Sci.* **150** 171–178

[18] Toporkov V, Tchernykh A and Yemelyanov D 2019 Budget and Cost-Aware Resources Selection Strategy in Cloud Computing Environments *Commun. Comput. Inf. Sci.* **1129** 667–677

[19] Li Q, Huang C, Bao H, Fu B and Jia X 2019 A Game-Based Combinatorial Double Auction Model for Cloud Resource Allocation *Proc. of the 28th Intern. Conf. on Computer Communication and Networks (ICCCN)* (IEEE) pp 1–8

[20] GridWay Metascheduler. Available at: http://www.gridway.org (accessed: 10.05.2020)

[21] Condor DAGMan. Available at: https://research.cs.wisc.edu/htcondor/dagman/dagman.html (accessed: 10.05.2020)

[22] Tchernykh A, Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Bychkov I, Basharina O, Alexandrov A and Rivera-Rodriguez R 2019 Orlando Tools: Development, Training, and Use of Scalable Applications in Heterogeneous Distributed Computing Environments *Comm. Com. Inf. Sc.* **979** 265–279

[23] Bychkov I, Oparin G, Tchernykh A, Feoktistov A, Bogdanova V and Gorsky S 2017 Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management *Procedia Comput. Sci*. **103** 162–167

[24] Pekec A and Rothkopf M H 2003 Combinatorial auction design *Manage. Sci.* **49(11)** 1485–1503

[25] Hershberger J and Suri S 2001 Vickrey prices and shortest paths: What is an edge worth? *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society) pp 252–259

[26] Jain R and Walrand J 2010 An efficient Nash–implementation mechanism for network resource allocation *Automatica (Oxf)* **46(8)** 1276–1283

[27] Lin W Y, Lin G Y and Wei H Y 2010 Dynamic auction mechanism for cloud resource allocation *Proc. of the10th IEEE / ACM Intern. Conf. on Cluster, Cloud and Grid Computing (CCGrid)* (IEEE Computer Society) pp 591–592

[28] Zhang Y, Niyato D, Wang P and Hossain E 2012 Auction–based resource allocation in cognitive radio systems *IEEE Commun. Mag.* **50(11)** 108–120

[29] Di S, Wang C L and Chen L 2013 Ex–post efficient resource allocation for self–organizing cloud Computers and Electrical Engineering **39(7)** 2342–2356

[30] Prasad A S and Rao S 2014 A mechanism design Approach to resource procurement in cloud computing *IEEE Trans. Comput.* **63(1)** 17–30

[31] Nejad M, Mashayekhy L and Grosu D 2015 Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds *IEEE T. Parall Distr.* **26(2)** 594–603

[32] Tanaka M and Murakami Y 2016 Strategy-proof pricing for cloud service composition *IEEE Trans. on Cloud Comput.* **4(3)** 363–375

[33] Landa R, Charalambides M, Clegg R G, Griffin D and Rio M 2016 Self–Tuning Service Provisioning for Decentralized Cloud Applications *IEEE Trans. Netw. Service Manag.* **13(2)** 197–211

[34] Feoktistov A G, Tchernych A, Kostromin R O and Gorsky S A 2017 Knowledge Elicitation in Multi-Agent System for Distributed Computing Management *Proc. of the 40th Inter. Convention on information and communication technology, electronics and microelectronics (MIPRO-2017)* (Riejka: IEEE) pp 1350–1355

[35] Feoktistov A G, Kostromin R O, Sidorov I A and Gorsky S A 2018 Development of Distributed Subject-Oriented Applications for Cloud Computing through the Integration of Conceptual and Modular Programming *Proceedings of the 41st Intern. Convention on information and communication technology, electronics and microelectronics (MIPRO-2018)* (Riejka: IEEE) pp 256–261

[36] Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G and Vahi K 2013. Characterizing and profiling scientific workflows *Future Gener. Comput. Syst.* **29(3)** 682–692

[37] Ho W, Xu X and Dey P K 2010 Multi-criteria decision making approaches for supplier evaluation and selection: A literature review *Eur. J. Oper. Res.* **202(1)** 16–24

[38] Feoktistov A G, Sidorov I A, Kostromin R O, Oparin G A and Basharina O Yu Methods and tools for evaluating the reliability of information and computation processes in grid and cloud computing systems *Proc. of the 1st Intern. Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE)* (CEUR-WS Proceedings) **2430** pp 60–69

[39] Irkutsk Supercomputer Center. Available at: http://hpc.icc.ru (accessed: 10.05.2020)