Azerbaijan Journal of High Performance Computing, Vol 3, Issue 2, 2020, pp. 164-180
https://doi.org/10.32010/26166127.2020.3.2.164.180

# Challenges of Influence Dynamic and Interactive Events on Resource Discovery Functionality outside of Distributed Exascale Systems

Shakiba Rezaei[1], Ehsan Mousavi Khaneghah[1], Araz R. Aliev[2]

[1]Department of Computer Engineering, Faculty Engineering, Shahed University, Tehran, Iran, Shakiba.Rezaei@Shahed.ac.ir, EMousavi@Shahed.ac.ir;
[2]High Performance Computing Research Advance Center, Department of General and Applied Mathematics, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, alievaraz@asoiu.edu.az

*Correspondence: Ehsan Mousavi Khaneghah, Department of Computer Engineering, Faculty Engineering, Shahed University, Tehran, Iran, EMousavi@Shahed.ac.ir

## Abstract

The resource discovery in Exascale systems should support the occurrence of dynamic nature in each stakeholder's elements in the resource discovery process. The occurrence of dynamic and interactive nature in the accountable computational element creates challenges in executing the activities related to resource discovery, such as the continuation of the response to the request, granting access rights, and the resource allocation to the process. In the case of a lack of management and dynamic and interactive event control in the accountable computational element, the process of activities related to the resource discovery will fail. In this paper, we first examine the concept function of resource discovery in the accountable computational element. Then, to analyze the effects of occurrence, dynamic, and interactive event effects on resource discovery function in the accountable computational element are discussed. The purpose of this paper is to analyze the use of the traditional resource discovery in the Exascale distributed systems and investigate the factors that should be considered in the function of resource discovery management to have the possibility of application in the Exascale distributed system.

Keyword: Resource Discovery, Dynamic and Interactive Events, Response Computing Unit, Functionality, Out of Systems.

## 1. Introduction

Unlike other system management elements, the resource discovery, as part of the execution of activities, operates outside of the computational system's limits and bounds (Alzboon, M. S., Arif, A. S., & Mahmuddin, M., 2016; Kaur, M., & Kadam, S. S., 2017). In the traditional computing systems, the resource discovery is invoked after a request that the load balancing cannot respond to. Recall of the resource discovery allows us to receive a request process to search for a resource answered by process application (Jamal, A. A., & Teahan, W. J., 2017; Zarrin, J., Aguiar, R. L., & Barraca, J. P., 2018). The resource discovery is based on one of the two policies using pre-established response structures or lack of structure, action search resources where request (Jamal, A. A., & Teahan, W. J., 2017; Nickbakhsh, N., & Aghaei, M. R. S., 2017). The resource discovery in traditional computing systems makes the resource discovery process either based on the Request Nature Set (RNS) or a Request Attribute Set (RAS) or both (Adibi, E., & Khaneghah, E. M., 2020). In traditional computing systems, both sets of application nature and request attributes are fixed during the implementation process of resource discovery-related activities. This makes it clear from the point of view of the resource discovery if the resource discovery focuses on the RNS, a resource, and if it is based on RAS, is more than one source capable of responding to the process request (Mengistu, T. M., & Che, D., 2019; Alzboon, M. S., Mahmuddin, M., & Arif, S., 2019).

In traditional computing systems, no change in the state of the computational element of the resource discovery, and precisely the absence of process request, causes the situation to fail to visualize the source changes' resource discovery activities. On the other hand, during the activities related to resource discovery, which is done outside the computational system, the lack of time constraints and, in some cases of location constraints may cause the activity of the resource discovery to fail. The computational element containing the resource found by the resource discovery can be part of another system or out of any system. If the computational element contains the resource outside any system, then the computational element contains constraints in the context of priority or delay of local activities. If the computational element contains the system member resource, a set of rules is defined in responding to requests outside the system and the constraints of responding to the non-local request (Jamal, A. A., & Teahan, W. J., 2017; Bhattacharyya, S., Conradie, L., & Arezki, R., 2017).

In traditional computing systems, the existence of a set of rules is possible, causing the resource discovery to be challenged after finding the computational element containing the source required, implementing activities related to achieving access rights and resource allocation, and the activities related to the resource discovery fail. The failure of resource discovery activities on each of the two activities to receive the source or resource allocation's rights can occur. In some of the mechanisms of management of resource discovery, two activity recognition activity and allocation in a sequence and some other mechanisms - the discovery of the source of these two activities occurs separately and at a time distance. A time gap can also provide the possibility of failure due to time in the computational element containing the source. On the other hand, the existence of delays due to the network and the return of messages due to the resource discovery and time in the computational element containing the source may also provide the possibility of failure (Singh, M., 2019, October; Luppi, E., 2020)

Outside the computational system, the set of constraints and constraints governing the request can differ from the pattern of constraints defined on the computational system's request. This may result in some differences in request and response constraints in the context of request and response indicators. Outside the computational system, the response pattern rules can differ from the existing laws in the computational system. The resource discovery is activated within the computational system, so the constraints governing the response pattern considered by this element are equivalent to the constraints of responding to requests in the computational system. The accountable element may comply with the constraints on the local computational system or constraints defined by itself. This issue is particularly relevant for accountability. The computational element responsible for the request has no response to the request, and responding to the computational system member's process request is voluntary. This makes the computational element responsive to the request by two different policies in the context of accountability. There is a response constraint in the first model, and if the request is accepted, the computational element is responding to an acceptable request. In the second model, there is no response constraint, and at any moment of the response process, there is a lack of accountability (Mengistu, T. M., & Che, D., 2019; Mengistu, T. M., & Che, D., 2019).

In the Exascale computational distributed systems, dynamic and interactive events in the computational system or outside the computational system on resource discovery function. Occurring, dynamic and interactive events may influence each factor that influences the resource discovery activities and cause failure in the resource discovery. The dynamic and interactive event in the system or system-level will cause the system or system environment to change so that the resource discovery mechanisms cannot manage the situation. This is significantly more tangible in resource discovery management elements outside the system (Gharb, H., Khaneghah, E. M., et al., 2019; Adibi, E., & Khaneghah, E. M., 2018; Bidhendi, Z.E., Pouria F., Khaneghah E.M., 2019).

A dynamic and interactive event within the computational system will cause the system state and process calling to change accordingly. The system management element can be determined by defining measurable indicators of the dynamic and interactive event on system functionality. The manager of the computing system enabling the resource discovery system, due to having accurate information about the process of the resource discovery, and the factors affecting the way of resource discovery management, can be used to determine the effect of occurring indices dynamic and interactive event on the computational element of the resource discovery and also the computational elements that play a role in the implementation of resource discovery activities.

Outside the system, there is no possibility of defining indicators mentioned by the resource discovery. When resource discovery is implemented outside the computational system's limits, it may be the resource discovery that lacks information about a dynamic and interactive event. Outside of the system's limits, other computational systems' mechanisms and patterns to analyze a dynamic and interactive event may vary with the mechanisms used in the computational system of the resource discovery. This difference may cause resource discovery to be unable to the resource discovery. This difference may cause resource discovery to be unable to detect the dynamic and interactive effects outside of the system's boundaries (Khaneghah, E. M., Aliev, A. R., Bakhishoff, U., & Adibi, E., 2018).

In this paper, while analyzing the concept of resource discovery failure outside of the computational system, investigating and analyzing the effect of dynamic and interactive effects on resource discovery performance and the concept of failure to implement resource discovery outside of the computational system will be discussed. This analysis makes it possible to examine the challenges of performance failure in resource discovery management outside the Exascale computational system.

## *2. Related Work*

In computational systems, the load balancer has the responsibility to create a response structure to requests. If a process invokes a resource that does not exist in the system and the load balancer cannot respond, the resource discovery is called. The resource discovery is attempting to find the resource of process interest outside the computational system's limits. Unlike other system manager elements, it operates out of the system and rules other than rules governing the computational system.

In (Adibi, E., & Khaneghah, E. M., 2018), the concept of the dynamic and interactive nature of distributed Exascale systems and describes the effects of the dynamic and interactive event on the resource request unit and resource discovery. Besides, it describes the challenges of resource discovery in Exascale systems. Resource discovery unit in Exascale systems, the need for management to occur is a dynamic and interactive event in the request process and when operating the resource discovery. The occurrence of dynamic and interactive events in the request process causes challenges in implementing resource discovery. If there is no management on the dynamic and interactive event, executive activities related to resource discovery will fail.

The dynamic and interactive nature of the computational modules occurs due to the cases mentioned above. (a) When a process in the system creates a new process that has not been defined in the system design time. This is equivalent to the formation of a new concept or a new variable in the scientific program implemented by the computational system, (b) two processes require communication during the initial design of an unexpected system, (c) The process is in process with the process out of the system, and this phenomenon is not defined when designing the system. This occurs due to the incorrect determination of the system (Khaneghah, E. M., & Sharifi, M., 2014).

The occurrence of a dynamic and interactive event in the system may cause to change the process status, the resource discovery, and the system in such a way that the conditions governing the request will comply with the request for a different pattern or change the current constraints; as a result, the resource discovery has the responsibility to manage the situation in a way that the existing resources are not stopped. Dynamic and interactive events in the Exascale systems distributed on each of the factors affecting the resource discovery function can be effective. On the other hand, outside the computational system, there is also a dynamic and interactive event efficient on the resource discovery function (Adibi, E., & Khaneghah, E. M., 2020).

The effect of the dynamic and interactive nature on the resource discovery is that View influence where due to the occurrence of dynamic and interactive nature may be the nature of the resource found by the resource discovery is different from the actual nature of the process request. Backward influence where The nature of the resource discovery's resource differs from the dynamic nature of the requesting process by the requestor's nature. Time influence: After finding the source by the source of the source discovery and its allocation, the process is driven out because of the response procedure's dynamic and interactive nature. In this case, the resource discovery failed to complete its respective activity, and all of its activities are unacceptable. In Global Activity, influence, dynamic, and interactive nature affects the overall activity's performance and changes it. By changing the characteristics of global activity, resource discovery should consider the new conditions and support them. The resource discovery in traditional computing systems cannot manage the four mentioned effects (Khaneghah, E. M., & Sharifi, M., 2014).

In ((Khaneghah, E. M., Aliev, A. R., Bakhishoff, U., & Adibi, E., 2018) introduces a mathematical model of events resulting in resource discovery failure in this type of computing systems based on descriptive spaces of resource discovery activities. This mathematical model helps to analyze this issue that the dynamic and interactive nature would lead to resource discovery and what capabilities resource discovery should have to prevent failure. Resource discovery in distributed Exascale computing systems, in addition to managing events resulting in failures due to not finding the resource, as well as failure to perform resource discovery activities at the acceptable time, needs to be able to manage events resulting in failures due to dynamic and interactive nature as well.

Resource discovery can be used in distributed Exascale systems when: (a) it can analyze the global activity process's functional and behavioral state. This means that resource discovery should not consider the computing process as an abstract element; instead, it should consider it as part of global activity. (b) It has mechanisms and methods to analyze changes in the process request nature during the resource discovery process. This implies that it should either be directly connected with requesting and activating resource discovery or can analyze the effects of other elements that change the request nature. It should also get information based on checking the process behavior about the frequency of request nature changes. It means having information on the requests' nature and their changing time. (c) It has mechanisms and methods for analyzing the requests' type. Thus, resource discovery should have a pattern for classifying resources and answerable resources and making decisions on changing the request type based on this pattern. To make such, resource discovery should either be directly connected with the process requesting and activating resource discovery or make decisions on changing the request type by examining the impacts of factors affecting the process. It should also decide on the frequency of changing the process request type by time check of the process.

In Exascale systems, multiple factors, including the lack of resources in an acceptable time, lack of accurate understanding of elements depending on the resource discovery activities, and occurring, can lead to resource discovery activity .failure

In (Adibi, E., & Khaneghah, E. M., 2020), a two-step mechanism is proposed for resource discovery. This method can be used in peer to peer computing systems and considers the scalability and dynamic characteristics of such systems. Some resources exist in peer - to - peer computing systems whose features change over time. Therefore, to discover a suitable source and consider static properties, mechanisms should investigate the system's changes. The method introduced in this paper is to determine the weight of each static and dynamic resource in the system. If a user requests a resource, computing finds resources as a super-peer to respond to the request. Resources are ranked based on defined mechanisms, and the most appropriate resource is assigned to the applicant.

One of the effects of a dynamic and interactive event of a change in RNS (Request Nature Set) or RAS (Request Attribute Set) can lead to resource discovery activities' failure. Resource discovery in Exascale systems based on policies assigned to it, the maximum resource (in traditional systems), or the lowest resource (in distributed systems) is similar to the request and then obtain access to that resource to have full or partial access.

After obtaining complete access to a resource, the resource discovery assigns it to the requesting process as the local source, and the system administrator is obliged to create transparency. Specific activities can be assigned to the requesting process, and the system administrator has a responsibility to create and manage a particular activity in remote resources. The dynamic and interactive event, leading to increased resource discovery and change in constraints RAC, affects RNS and RI (Request Image).

The difference between traditional systems and Exascale systems is the method of creating RNS. When resource management is active in traditional systems, the resource discovery provides RNS concerning time constraints, location, and RAS based on the request type. The Exascale distributed system is one of the challenges of resource discovery, type, and nature of the request due to dynamic and interactive events. If the resource discovery mechanism is based only on the request type and is not considered dynamic and interactive, it can cause or increase the response time. Therefore, in the Exascale distribution of information based on the nature of the request that arises from a resource request and global activity, the response structure is created, and the most appropriate mechanism for discovering the resource is selected.

In (Kaur, M., & Kadam, S. S., 2017), we have proposed a two-phased multi-attribute decision making (MADM) approach for the discovery of grid resources by using P2P formalism and provides the best suitable resource (s) to grid users. The majority of the resource discovery techniques rely on static resource attributes during resource selection. The matching resources based on the static resource attributes may not be the most appropriate resources for user applications' execution because they may have heavy job loads, less storage space, or less working memory (RAM).

The first phase describes a mechanism to discover all matching resources and applies the SAW method to shortlist the top-ranked resources communicated to the requesting super-peer. The second phase of our proposed methodology applies an integrated MADM approach on the list of selected resources received from different super peers. The pairwise comparison of the resources concerning their attributes is made, and each resource's rank is determined. The top-ranked resource is then communicated to the grid used by the grid scheduler. The proposed method methodology enables the grid scheduler to allocate the most appropriate resource to the user application and reduces the search complexity by filtering out the less relevant resources during resource discovery.

The factors that contribute to resource discovery failure are different. The resource discovery's failure factors outside the boundaries of the computational system are usually considered in the traditional computing systems due to the high cost of discovering and increasing the execution time of the resource discovery. If the execution time constraint is violated, the resource discovery process is faced with failure. The Exascale distributed system of the management mechanism and the way of solving the failure operations are expressed. In other words, the resource discovery management component's failure operations correspond not to enable restart operation.

### *3. The Function of the Resource Discovery Outside the System in Traditional Computing Systems*

The resource discovery in traditional computing systems involves acquiring RAS, or RNS, or (RAS, RNS) from the load balancer, and will start activities related to resource discovery. The resource discovery, based on the constraints governing each of the two space, RNS, and RAS, or combination constraints of two spaces, initiates actions related to the resource's discovery (Adibi, E., & Khaneghah, E. M., 2020). If the resource discovery, regardless of the mechanism used, tries to find the source of the request, in the Alpha, from the point of view of the resource discovery, is described in the form expressed in formula 1.

$$Alpha_{RD} \overset{Establishes}{\because} \left[ Request\ (T, L, S, C) \overset{\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \end{bmatrix}}{\leftrightarrows} Answer\ (St, R, Rem) \right] \quad Eq. 1$$

As observed in formula 1, the local manager of Alpha, from the point of view of the resource discovery, requests mapping the source to the source in the Alpha leading to the request. In traditional computing systems, regardless of the resource discovery mechanism, the resource discovery process enables the request to form $Request\ (T, L, S, C)$. The resource discovery seeks to find the computational unit capable of responding to the request expressed in the form $Request\ (T, L, S,)$ C.

The process request of the resource discovery is a variable of time constraints, spatial constraints, the constraints governing the computational requesting system, and the communication constraints between requests and replies. The resource discovery sends a process request to the local manager of any external computational resource outside the system based on the mechanism used to discover the source. Suppose the Alpha's local manager can establish formula 1 and accountability to the request. In that case, the resource discovery will terminate the resource discovery process if the Alpha's computational unit can establish formula 1 and respond to the request to form $Answer\ (St, R, Rem)$, then sent to the resource discovery.

In the form, $Answer\ (St, R, Rem)$ respond to a variable request of the status of the accountable computational element, the constraints governing the computational unit, and how Alpha was present in the response process. The St sign indicates the description of the Alpha's state, and R indicates a constraint-based on the Alpha. The St can implicitly include the resource status requested. Let R denote the constraints of usage time, access rights constraint, and how to allocate it. In traditional computing systems, rem indicates that whether the presence of Alpha in response to the request is deterministic or not? If Alpha's presence is deterministic, then the value $(St, R)$ cannot be changed during the response to the request in the traditional computing systems. If the procedure of Alpha's presence is uncertain and possible to exist in response to the request of the absence or change of Alpha's status, then the value $(St, R)$ is a function of time-independent variables.

The local manager of Alpha, under $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \end{bmatrix}$ constraints, moves mapping request to the source. Response constraints are the set of constraints on how the source is used. The operations constraints are defined as the constraint set that is defined in the activities applicable to the resource. Time constraints apply the set of constraints governing the resource utilization time. System constraints govern the resource arising from a resource's existence in a given computational system. The Alpha can be part of a computational system or is itself an independent computational element. The constraints defined in the matrix $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \end{bmatrix}$ may lack the value that is not considered by the Alpha's local manager. In the computational system, the Alpha specifies the constraints and constraints specified by the Alpha and, in some cases.

In formula 1 the defining variables of the request are determined by the process requestor.

In traditional computing systems, because the resource request's process state is unaltered in executing discovery activities, the variables $(T, L, S, C)$are remain unchanged in the process. In the traditional computing systems, after Alpha's computational element determine the activation of the resource discovery and finding the Alpha, the values for $(St, R, Rem)$. in traditional computing systems, only if rem is a non-deterministic status descriptor for the process of the presence of Alpha, in the process of responding to the process request of the resource discovery, it is possible to not respond to the request by the Alpha. The impossibility of answering the request by Alpha results in the failure of the resource discovery process outside the computational system of the resource discovery activation

In traditional computing systems such as peer-to-peer and grid computing systems, if the Alpha uses the volunteer pattern to respond to the resource discovery management requests' requests in other computational systems, the Rem variable is a non-deterministic presence in the response process to requests. In such a situation, the resource discovery function's failure can be due to the lack of resources or the change in the state of Alpha in response to requests. In this type of computational systems, if the Rem variable represents a non-deterministic presence of Alpha in the response process, and the variables St and R are a function of time, in this case, the variation of any constraints expressed in the matrix $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \end{bmatrix}$ can fail the function of the resource discovery.

### *4. Proposition*

In traditional computing systems, the resource discovery activation means accepting the representation process-related request on behalf of the source discovery. In traditional computing systems, when the Alpha is adapted to accept the resource discovery request, the request will be applied as a member of the request for resource access requests set. The resource discovery provides the acquisition rights to the resource required and the resource allocation required by enabling the resource discovery.

In traditional computing systems, the stakeholder's elements related to the resource discovery manager, including the source, the destination, and the governing system on the source computing system and the governing system on the target computational element, are unchanged. It causes that in formula 1, the constraint defining the request, the constraints governing the response to the request, and the constraints governing the mapping of the request to the resource, are stable. In this type of computational system, the resource discovery's failure occurs only if the Alpha uses a deterministic presence pattern to the requesting process request. In an uncertain present, as Alpha's computational element may attempt to leave the response process to a process request, the resource allocation process that is part of resource discovery may fail.

The function of the resource discovery in the Alpha can be considered as the form of space <Answer, Answer Condition, Mapping Condition, <Permissions, Allocation > >. In this form, the axial element of the definition of resource discovery is based on the concept of response. The concept of response from a resource discovery contains a resource that can respond to the request by considering request constraints. The resource concept from the point of view of resource discovery management includes process constraints that require access to the resource. If the Alpha can satisfy the constraints governing the process of requesting, the process of finding the resource is terminated and the process of access and allocation starts.

In traditional computing systems, regardless of the mechanism used to discover resources, the answer is unique. In traditional computing systems, resource discovery management is based on the mechanism used, to investigate each candidate's computational element to find the resource. If the specified computing element can meet formula 1, the computational element contains the resource from the resource discovery point of view. This makes it possible to implement the resource discovery process only once in traditional computing systems. The result of the resource discovery process's implementation, or leading to the discovery of the computational element that applies to formula 1, has a source that can be termed as an answer to the resource discovery management process demand response element. From the point of view of resource discovery management, the computational element contains the response to the process request of the agent discovery only once. In traditional computing systems, the resource detection process follows a Bernoulli distribution pattern. If a process leads to success, then the process terminates because it is found in the computational element. If a resource discovery process fails, then because no computational element can respond to the source detection mechanism, it terminates. The computational element's nature contains the source so that the computational element contains a process resource concerning all constraints of response and request or does not contain the desired resource. In traditional computing systems, no other state can be considered for accountable elements.

In traditional computing systems, the concept of resource discovery is unique, and the process of resource discovery cannot lead to another resource discovery process.

In traditional computing systems, provided that the accountable computational element's presence is deterministic, the constraints governing the response during all the resource discovery process and resource allocation are fixed. In traditional computing systems, the acceptance of a request by the accountable computational element occurs when (a) the resource is proportional to the process request (b) the constraints governing the request are consistent with the constraints dominating the response.

In traditional computing systems, as the computational element at the time of the accept request, the constraints on the request are accepted, and the action is defined as constraints that lead to the completion sub-activity of the resource discovery from the resource discovery activity set; therefore, the process of discovering the resource and allocating resource to the requesting process does not respond to the constraints change. The solution's constraints are derived from the constraints governing the computational element containing the resource and the computational system that is the computational element of its membership. The lack of response constraints means not changing the useful parameters on the response to process requests in the computational element and the computational system.

The constancy of the constraints governing the response causes that the model used to grant access rights and the allocation to the process is constant. The constancy of the constraints dominating the answer makes it clear that the resource discovery, after allocating the source to the requesting process, does not need to review the constraints governing the response, as well as the access rights pattern as well as how the resource is allocated to the requesting process. This, from the point of view of the resource discovery, means that after each step of the set of discovery activities, access rights, and the way of allocation, there is no need to review the accountable computational element responsible for the change. considering that in case of violation of the constraints governing the mapping, the activities related to the resource discovery will fail and also due to the constant state of the computational element containing the resource, the constraints governing the response and request will cause that (a) failure due to violation of System and Operation constraints is not defined in the traditional computing systems, and it cannot be done. (b) In case of failure due to violation of Answer constraint, only if it is possible to occur in the hardware or network of the accountable computational element gets into trouble, and it is not impossible to continue to respond to the request. (c) About the failure due to Time constraints, even if factors affecting the process of responding to requests in the computational element and the computational system may be stable, it may occur due to the dominant pattern of process management. Model acceptance of process lieutenant general functions by the accountable computational element is effective on the possibility of a time type failure.

Traditionally, in traditional computing systems, one of the two models in terms of the process taking into account is considered a local processor considered a global process as the requestor's process management patterns. Changing process execution time, change of process executive priority or any change in process execution process in the accountable computational element can lead to the Time type failure. If a failure of the Time type occurs and this failure due to the hardware function and the computational element network is not accountable, then this failure is considered in the judgment of taking a Time Out for the process lieutenant execution.

In traditional computing systems, the function of resource in the accountable computational element is described by two variables (Resource, Time). The stability of the effective elements states on the accountable source in the accountable computational element and the system in which the computational element is part of it makes it possible to define the function of resource discovery management in traditional computing systems based on two resource spaces and time constraints. Resource space includes constraints governing the request as well as constraints governing the response. Time-space also includes time constraints of resource usage. This causes that in traditional computing systems, the performance of the resource discovery in the accountable computational element is the probability that there are two types of failure due to the no impossibility of obtaining resource and failure due to the violation of time constraints due to accountability.

## *5. Argument*

In Exascale distributed systems, at each moment of executing the activities related to resource discovery, it is possible to occur dynamic and interactive nature. Due to the limitation of the resource discovery performance, which is part of it in the computational system and the other part of it outside the computational system, a dynamic and interactive nature may occur in each of the resource discovery domains. The dynamic and interactive nature of the functional space within the computational system may affect the source and process computational element of the resource discovery and the resource discovery function in the computational system. The dynamic and interactive event within the computational system may affect the concept of request and the constraints governing the request. A dynamic and interactive event occurring within the functional space outside the computational system may cause the accountable computational element, the resource requested, and the resource discovery management element's function outside the computational system. A dynamic and interactive event occurring outside the computational system may affect the concept of response and mapping constraints. A dynamic and interactive event occurs in each of the two activities of resource discovery, affecting the resource discovery management element's function in another activity space.

In large scale systems, resource finding is a relative activity, and Matching's concept is also considered an acceptable activity in the resource discovery performance. This may cause the resource discovery process to be considered a global activity. In this situation, a set of Alphai elements responds to each part of the requestor requirement, and, as the degree of Matching is established between the source in Alphai and the process request activating of the resource discovery. Therefore, Alphai is part of the global activity of accountability to process requests. This makes it possible in the Exascale distributed system to examine the function in the $[Alpha_1 \quad \cdots \quad Alpha_n]$ matrix to examining the function of the resource discovery in the individual Alpha. In this situation, the concept of request from the point of view of resource discovery management, the concept of responses in Exascale distributed systems, is considered a $[Alpha_1 \quad \cdots \quad Alpha_n]$ matrix. The definition of the concept of response in a matrix form means considering one or more resources to respond to an activating request for resource discovery. From the point of view of resource discovery in Exascale systems, each $Answer_i$ response to the quaternary form (R, St, Rem, G) is defined that four defining space $Answer_i$ and $Answer_i$ constraints are the sources of resource discovery.

In the Exascale distributed system, the definition of constraints governing the resource to form R(Resource Limitation(t), Local System Limitation(t), Global Activity Limitation(t), Time Usage(t), Permission (t), Allocation(t), Similarity (t)) is considered . In the Exascale distributed system, the accountable resource can be a member of the computational system that consists of one or more of a computational element, so the accountable resource to request can be present in more than one global activity. Constraints govern the accountable resource to process requests derived from resource constraints, resource definition in the computational element, and the presence of a computational element containing the resource in global activity (or activities). On the other hand, according to the nature of the Exascale system's distribution and the probability of occurrence of dynamic and interactive nature, variables of R space are variable independent of time. In the Exascale distributed system, the concept of resource and considering resource usage constraints, access rights, and allocation constraints that are variable of time-independent variable, the concept of similarity to process request resource are considered a of a time-independent variable.

In the Exascale distributed system, the responsive source's concept to request's state follows a similar pattern to what is shown in figure 1.

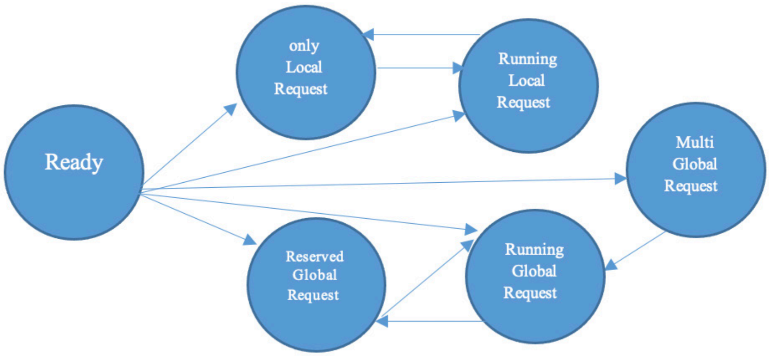


*Fig. 1. The well-defined situations for accountable resource in Exascale distributed system*

As shown in figure 1, in addition to the two running to respond and ready to Local Request accountability, situations in the traditional computing systems, the situation is considered only to respond to the local request, reserved to global request, and also running to respond to a global request and the existence of accountability in more than one global request. This causes the concept of accountability in more than one global request that needs to consider the global activities or G space in which the resource is responding to one or more of a process belonging to the global activities. A responsive resource to request can have constraints on the number of global activities used to mention resources. G space is a function of time-independent variables, resource capability, membership capacity of a computational element in global activities, and time.

In the Exascale distributed system, the matrix of constraints governing the mapping in the request to the responsive source to the request in the form $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \\ Global\ Activity_{condition} & Related\ Advantage_{condition} \end{bmatrix}$ is defined. In addition to the constraints defined in the traditional computing systems for mapping, the two constraints resulting from activity (or global activities) involve the computational element of Alpha and constraints and limitations due to the relative advantage of Alpha. The constraints governing the response process in Exascale distributed systems, unlike traditional computing systems, are a function of time-independent variables. In the Exascale distributed system, the function of resource discovery in the computational element is defined in the form (R, Mapping Matrix, St) because of two concepts of the possibility of not answering a resource in a computational element to process the request, the definition of resource usage state for global and local activities, as well as a function of time-independent constraints on resource and mapping constraints.

In the Exascale distributed system, unlike traditional computing systems, which are the axial element of system activity based on process, the axial element of system activity is based on the concept of global activity and process definition as part of the global activity. In the Exascale distributed system, to carry out the activities related to resource discovery, three global activities are responsive to the request, the global activities that the resource is a part of and the global activity that the process was requesting member. In the accountable computational unit, two first and second global activities will be affected if dynamic and interactive. In investigating the function of the resource discovery management system, the dynamic and interactive events on the computational element of the requesting, changes of the third global activity affect the resource discovery management system's performance in the accountable computational element.

In the Exascale distributed system, unlike traditional computing systems that a computational element responds to a process request, several computing elements may be responding to a process request. In such a situation, each computing element is based on the concept of its relative advantage in responding to a request (or part of a process request) in the global activity leading to a process request. The concept of comparative advantage indicates the computational element's ability to respond to the requests for resources, which the function of the computational element makes it capable of answering the requests for those resources. In the Exascale distributed system, the concept of comparative advantage is a variable of a time-independent variable. The relative advantage space may change due to the passage of time and the computational element's functional status or due to a dynamic and interactive event. This change may violate the definition of comparative advantage of the computational element.

Given the definition of the mapping constraints matrix and considering the function of the Alpha's resource discovery, we can consider the following failure to function the computational system's resource discovery.

A) RBroken, a dynamic and interactive nature may change the resource situation so that the source's new source may not involve process constraints. The function of resource discovery in conventional computational systems is in such a way that only when examining the feasibility of process application implementation in the accountable computational element, it is attempted to investigate whether the computational element can respond to a process request or not. The resource discovery during resource discovery processes does not provide information about the correspondence between constraints of request to the response's constraints. In a dynamic and interactive event and changing the R descriptor's parameters, the resource discovery has no information about the change. A dynamic and interactive event occurs from three approaches to impact resource constraints, the effect on the global activity, and how resource operation can affect the resource's parameters.

R (Resource Limitation(t), Local System Limitation(t), Global Activity Limitation(t), Time Usage(t), Permission (t), Allocation(t), Similarity (t))

B) MappingBroken, Changing each of the elements of the $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \\ Global\ Activity_{condition} & Related\ Advantage_{condition} \end{bmatrix}$, due to the occurrence of a dynamic and interactive event, which may cause the accountable element to lack the ability to respond to the request constraints. In this situation, the $\begin{bmatrix} Answer_{condition} & Time_{condition} \\ Operation_{condition} & System_{condition} \\ Global\ Activity_{condition} & Related\ Advantage_{condition} \end{bmatrix}$ the matrix cannot establish the mapping between the request space and the answer, or the matrix cannot establish a mapping based on request and response constraints. The resource discovery if it can review its function number1 during the execution time of its resource discovery activities, then the failure of the type MappingBroken will not occur. In the failure of the type of MappingBroken, any of the mapping conditions can fail to occur.

C) StBroken, a dynamic and interactive event, may cause the resource situation to change. The change of resource status due to the source's dynamic and interactive nature may result in the change of the resource status based on the process specified in figure 1 or may lead to an undefined state in figure 1. Changing the resource status makes the accountable element unable to respond to the request (or part of the request) of the process enabling resource discovery.

In the three cases mentioned, the requestor process is considered a fixed element in the failure analysis. In each of the three failures mentioned, the dynamic and interactive effects on the accountable element result in the change of the elements of the resource discovery's function in the mentioned element, and it leads to the failure of the activities of the resource discovery.

## *6. Conclusion*

The resource discovery in the Exascale system and finding a process resource are responsible for creating a response structure. The major part of the resource discovery occurs outside the request process-related computing system's limits and bounds. This causes that in case of occurrence, dynamic and interactive event processes outside the request process-related computational system are also affected and require the mechanisms and frameworks for management to occur, dynamic and interactive event out of the computational system. To manage and control the impact of the dynamic and interactive event on the accountable element, we need to analyze resource discovery's function in the accountable computational element and redefine the elements of this function with consideration of the effects of dynamic and interactive effects. To manage the dynamic and interactive event in the accountable computational element, the resource discovery requires the development of the concept of resource status, resource status, and mapping conditions of the function of the resource discovery. This development of resource discovery in the responsive element makes it possible to change the function from an unrelated function to time to an independent time and event function. This change function of resource discovery causes the resource to need to develop the concept of responsibility and consider accountability. The resource discovery's development function requires the development of the concept of the resource discovery from a one - dimensional structure to a multidimensional structure. This makes it clear that the resource discovery requires consideration of the status of the responsive element in the resource discovery process and other units to support the occurrence of the dynamic and interactive event in the accountable computational element.

*Reference*


Adibi, E., & Khaneghah, E. M. (2018). Challenges of resource discovery to support distributed exascale computing environment. *Azerbaijan J. High Pefrom. Comput., 1*(2), 168-178.

Adibi, E., & Khaneghah, E. M. (2020). ExaRD: introducing a framework for empowerment of resource discovery to support distributed exascale computing systems with high consistency. *Cluster Computing, 23,* 3349-3369.

Alzboon, M. S., Arif, A. S., & Mahmuddin, M. (2016). Towards self-resource discovery and selection models in grid computing. *ARPN J. Eng. Appl. Sci, 11*(10), 6269-6274.

Alzboon, M. S., Mahmuddin, M., & Arif, S. (2019, September). Resource Discovery Mechanisms in Shared Computing Infrastructure: A Survey. In *International Conference of Reliable Information and Communication Technology* (pp. 545-556). Springer, Cham.

Anderson, D. P. (2019). Boinc: A platform for volunteer computing. *Journal of Grid Computing,* 1-24.

Bharti, M., Kumar, R., & Saxena, S. (2018). Clustering-based resource discovery on Internet-of-Things. *International Journal of Communication Systems, 31*(5), e3501.

Bhattacharyya, S., Conradie, L., & Arezki, R. (2017). Resource discovery and the politics of fiscal decentralization. *Journal of Comparative Economics, 45*(2), 366-382.

Bidhendi, Z.E., Pouria F., Khaneghah E.M. (2019) *Challenges of Using Unstructured P2P Systems to Support Distributed Exascale Computing. Azerbaijan Journal of High Performance Computing. 2* (1), 3-6.

Gharb, H., Khaneghah, E. M., et al. (2019) *Challenges of Execution Trend in Distributed Exascale System. Journal of Distributed Computing and Systems, 2* (1), 140-151.

Jamal, A. A., & Teahan, W. J. (2017). *Alpha multipliers breadth-first search technique for resource discovery in unstructured peer-to-peer networks. Int. J. Adv. Sci. Eng. Inf. Technol,* 7(4), 1403-1412.

Kaur, M., & Kadam, S. S. (2017). Discovery of resources using MADM approaches for parallel and distributed computing. *Engineering Science and Technology, An International Journal, 20*(3), 1013-1024.

Kaur, M., & Kadam, S. S. (2017). Discovery of resources using MADM approaches for parallel and distributed computing. *Engineering Science and Technology, An International Journal, 20*(3), 1013-1024.

Khaneghah, E. M., & Sharifi, M. (2014). AMRC: an algebraic model for reconfiguration of high performance cluster computing systems at runtime. *The Journal of Supercomputing, 67*(1), 1-30.

Khaneghah, E. M., Aliev, A. R., Bakhishoff, U., & Adibi, E. (2018). The influence of exascale on resource discovery and defining an indicator. *Azerbaijan J. High Peform. Comput., 1*(1), 3-19.

Luppi, E. (2020). Introduction to Distributed Computing. *TORUS 1–Toward an Open Resource Using Services: Cloud Computing for Environmental Data,* 163-177.

Mengistu, T. M., & Che, D. (2019). *Survey and taxonomy of volunteer computing. ACM Computing Surveys (CSUR), 52*(3), 1-35.

Nickbakhsh, N., & Aghaei, M. R. S. (2017). Resource discovery algorithm based on hierarchical model and Conscious search in Grid computing system. *Journal of Soft Computing and Applications, 2017(1),* 24-43.

Prabhakaran, S., Doshi, K. A., & Bernat, F. G. (2019). *U.S. Patent Application No. 16/241,891.*

Singh, M. (2019, October). An Overview of Grid Computing. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 194-198). IEEE.

Zarrin, J., Aguiar, R. L., & Barraca, J. P. (2018). Resource discovery for distributed computing systems: A comprehensive survey. *Journal of Parallel and Distributed Computing, 113,* 127-166.

***Submitted: 12.08.2020***
***Accepted: 12.11.2020***