Azerbaijan Journal of High Performance Computing, Vol 6, Issue 1, 2023, pp. 113-120
https://doi.org/10.32010/26166127.2023.6.1.113.120

Resource Discovery in Distributed Exascale Systems using a Multi-Agent Model: Categorization of Agents Based on Their Characteristics

Fakhraddin Abdullayev

Azerbaijan State Oil and Industry University, Baku, Azerbaijan, fexreddinabdullayev98@gmail.com

*Correspondence: Fakhraddin Abdullayev, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, fexreddinabdullayev98@gmail.com

Abstract

Resource discovery is a crucial component in high-performance computing (HPC) systems. This paper presents a multi-agent model for resource discovery in distributed exascale systems. Agents are categorized based on resource types and behavior-specific characteristics. The model enables efficient identification and acquisition of memory, process, file, and IO resources. Through a comprehensive exploration, we highlight the potential of our approach in addressing resource discovery challenges in exascale computing systems, paving the way for optimized resource utilization and enhanced system performance.

Keyword: HPC, Resource Discovery, Agents, Dynamic and Interactive Event, Exascale Systems.

1. Introduction

Unlike traditional systems, processes in distributed exascale computing systems exhibit dynamic and interactive nature. The dynamic and interactive nature of processes manifests itself in the following three ways:

a) A process in the system creates an undefined process during the design of the computing system (Adibi, E., & Khaneghah, E. M., 2018).

b) There is an inter-process communication and interaction between other processes within the system, which is not defined during the design of the system (Adibi, E., & Khaneghah, E. M., 2018).

c) There is an inter-process communication and interaction between other processes outside the system, which is not defined during the design of the system (Adibi, E., & Khaneghah, E. M., 2018).

The situations described in cases a) and c) are related to the unavailability of the required resources to execute the assigned process in the system. An efficient resource discovery mechanism should be implemented in a distributed exascale system to address this. Efficient resource discovery is essential for achieving optimal utilization of computational resources and enabling high-performance computing in

distributed exascale systems. As the scale and complexity of these systems continue to grow, conventional centralized approaches to resource discovery need to be revised to meet the requirements of such environments. Consequently, researchers have shifted towards decentralized decision-making and harnessing the capabilities of autonomous agents to effectively tackle resource discovery challenges by employing agent-based models.

This article introduces a novel approach to resource discovery in distributed exascale systems by utilizing a multi-agent model that categorizes agents based on their specific characteristics. By combining agent-based modeling with resource classification, our proposed system offers an efficient and scalable solution for identifying and acquiring various types of resources, such as memory, processes, and I/O, while accounting for agent behaviors specific to resource search and utilization.

The main idea of our approach is to decentralize the resource discovery process by enabling autonomous agents capable of gathering information about their respective environments, establishing communication with neighboring peers, and making intelligent decisions to acquire resources. By categorizing agents based on their characteristics, we enhance the efficiency and effectiveness of the resource discovery process.

The first aspect of agent categorization revolves around resource types. Various resources possess unique characteristics and demands, requiring specialized agents' involvement for effective exploration and acquisition. For instance, agents specialized in memory resource discovery are equipped with specific expertise and strategies for memory allocation and management. Likewise, agents focused on process or I/O resource discovery are endowed with specialized capabilities tailored to those particular resource types. This categorization enhances system performance, facilitating purposeful and efficient resource acquisition.

The second aspect of categorization delves into the behavior of agents. Agents demonstrate distinct behaviors when exploring resources in neighboring peers, which are essential for adapting to the dynamic nature of exascale systems. By categorizing agents according to their behavioral attributes, we empower them to effectively handle such situations by making informed decisions rooted in maintaining system stability while achieving efficient resource acquisition.

2. Related Works

Adibi, E., & Mousavi Khaneghah, E. (2020) presents a framework to empower resource discovery units in distributed Exascale computing systems. The framework enables efficient resource discovery and effective management of dynamic and interactive events in these systems. Through analysis, the impacts of the dynamic and interactive nature on the functionality of the resource discovery unit (ExaRD) are examined, leading to decisions regarding framework elements and functionality.

Dimakopoulos, V. V., & Pitoura, E. (2003, August) propose a distributed approach

for resource discovery in multi-agent systems. Agents maintain a local cache with information about available resources, and neighboring agents are contacted if the requested resource is not found in the cache. Variations of the flooding-based search method are explored and evaluated through simulations.

Kambayashi, Y., & Harada, Y. (2007) present an efficient resource-locating method using a multi-agent approach in a pure P2P system. The system model is based on a distributed hash table (DHT) P2P system, consisting of high-performance nodes with DHT and regular nodes without DHT. Cooperative multi-agents manage and organize the resources and their information. To reduce communication traffic, migrating multi-agents are strategically clustered based on logical similarity.

Ding, S., Yuan, J., & Hu, L. (2005, March) present an agent-based resource management model, focusing on the structure and function of the agents. The model adopts a "First Come, First Served" strategy based on local resources for task assignments. A heuristic algorithm is proposed to facilitate resource management, utilizing resource advertisement and discovery techniques. The agents are organized in a graph, and the algorithm emphasizes multi-agent cooperation to schedule tasks effectively. This methodology aims to optimize task scheduling within the resource management framework.

Tan, Y., Han, J., & Wu, Y. (2010) present a survey of research that integrates multi-agent systems and peer-to-peer resource discovery services in Grid systems. The proposed discovery services aim to possess desirable features such as scalability, reliability, self-organization, and self-healing. These features enable efficient resource search by leveraging replication mechanisms and facilitating query operations.

In (Tan, Y., & Zheng, Z., 2009, May) proposed multi-agent approach for resource discovery in P2P systems addresses the challenges of locating shared resources in decentralized networks. Agents maintain local resource information, and mobile agents migrate to uncover resources, reducing resource discovery time and improving scalability and robustness.

In (Yamasaki, J., & Kambayashi, Y., 2010, November), an efficient resource locating method is proposed for peer-to-peer (P2P) systems using a multi-agent system. The method utilizes the ant colony optimization (ACO) algorithm to assist mobile agents in finding resource-rich nodes. However, alternative algorithms, such as the honey bee algorithm, have been suggested to be more efficient for resource discovery. This paper explores the integration of bee-like agents into the resource discovery method to optimize the behavior of mobile agents. The goal is to achieve efficient migration through direct communication between bee agents on the dance floor rather than indirect pheromone-based communication.

Zarrin, J., Aguiar, R. L., & Barraca, J. P. (2018) address the challenge of resource discovery in large-scale distributed computing environments that offer diverse and heterogeneous computing resources for sharing and distributed computing. The authors investigate the current state of resource discovery protocols, mechanisms, and

platforms, specifically focusing on design aspects. The paper classifies the relevant aspects, general steps, and requirements for constructing a novel resource discovery solution into three categories: structures, methods, and issues.

2. Categorization of Agents

In contrast to traditional computing systems, our proposed approach in distributed exascale computing systems adopts a multi-agent framework. In the proposed approach, agents are programs responsible for conducting resource discovery. These agents are specialized programs with autonomous functionality that automatically perform resource search operations. Agents in this framework are categorized based on two concepts: 1. Resource type and 2—search behavior.

2.1. Resource-specific Agents

These agents are specifically assigned to search for designated resources. Their classification is determined by the types of resources they are intended to explore. The resource types considered include processes, memory, files, and IO. Consequently, the resource-specific agents are further classified into four distinct groups:

1. Process agents: These specialized agents are responsible for discovering resources capable of executing processes efficiently and effectively. The process can be executed on CPU, GPU, and TPU resources. Each of these devices is specialized for specific types of operations (Raj, P., & Sekhar, C., 2020). Process agents store their operation pool and input arguments, enabling them to determine whether the required operation is assigned to them during resource discovery.

2. Memory agents: Agents in this category are designed to identify and acquire memory resources, ensuring optimized memory allocation and management. When referring to memory, it can encompass RAM and swap memory. Memory agents are involved in memory discovery based on a given memory interval or a specific memory value. In this case, the concept of resource matching is of great importance. Resource matching determines the degree of compatibility between the requested and available resources. Allocating resources in nodes with an excessive amount beyond the requested limit is inefficient because the distributed system has the requested resource in a distributed form, allowing the allocation of the resource on multiple nodes. This practice can lead to a degradation in system performance. To mitigate this, a semantic matching approach can be utilized (Castano, S., Ferrara, A., Montanelli, S., & Racca, G., 2004, April).

3. File agents: These agents specialize in the discovery and management of file resources, facilitating efficient access, storage, and manipulation of data files. To conduct file searches, the agents employ file hashing methods, comparing the hash of the searched file with the hashes of the files available on the queried node. If the hashes match, the file is considered found (Selvaraj, C., & Anand, S., 2012).

4. IO agents: IO agents specialize in exploring and utilizing input/output resources,

enabling effective data input/output operations within the distributed exascale computing system. These agents handle resource discovery and management, specifically concerning resources such as storage devices (HDD, SSD, etc.) and network adapters within the distributed system. By focusing on resource exploration and utilization, IO agents contribute to optimizing data input and output operations, enhancing the overall performance of the distributed exascale computing system.

2.2. Behavior-specific Agents

In high-performance computing systems, resource discovery occurs outside the system (Ding, S., Yuan, J., & Hu, L., 2005, March). In traditional systems, the computation element initiating the resource search sends queries to its neighboring nodes, storing information about the resources in its local memory. The resource search is completed within that computation element if the required resource is available in those nodes. Suppose that A- is the node initiating the resource discovery process, B – is the neighboring node of A to which the request is sent, $C_1, C_2, \ldots, C_n$ – the neighboring nodes of B to which resource discovery requests can be made, r – the requested resource. In distributed exascale computing systems, the possible scenarios related to resource availability during off-system resource search can be demonstrated as follows:

Node B can provide resources r on its own.

Node B can provide resource r with the assistance of neighboring nodes $C_1, C_2, \ldots, and\ C_n$, to which it sends requests for resource discovery.

Node B cannot provide resource r, but its neighboring nodes $C_1, C_2, \ldots, C_n$ can. In this case, if node C_i is not accessible to node A, then node B acts as the resource provider.

Node B cannot provide resource r, but its neighboring nodes $C_1, C_2, \ldots, and\ C_n$ can. If node C_i is accessible to node A, node C_i becomes the resource provider.

Both node B and its neighboring nodes $C_1, C_2, \ldots, C_n$ can provide resource r.

Neither node B nor its neighboring nodes can provide the resource.

The first five of the six cases shown above are success cases and allow for resource provision. We will assign behavior-specific agents for each of these five cases. These agents are as follows:

Self-Provisioning Agents. These agents only search for the requested resource among the neighbors of the initiating node. If the resource is not found, a failed response is returned. For these agents to function properly, the node initiating the resource discovery process must store a list of its neighbor nodes in its local memory. This list should also include the timestamp indicating the last time information about each neighbor node was obtained. This is crucial because distributed exascale computing systems are dynamic in nature, and nodes can join or leave the system over time. By maintaining the list of neighbor nodes and their corresponding timestamp, the resource discovery agents can adapt to the changing network topology and ensure the accuracy of resource availability information.

Collaborative Provisioning Agents. Distributed systems sometimes encounter situations where the node initiating the resource discovery can only provide a certain

portion of the requested resource. For example, if 8GB of RAM is requested, node X can only provide 4GB of RAM. In such cases, Collaboration Provisioning agents create new resource discovery request for the unfulfilled portion of the resource. The unfulfilled portion of the resource is then searched among the neighbors of the requested node. During this process, the value of the resource to be searched in neighbor node C_k is determined by the following guidelines:

$$v_{req}^{k} = V - v_{pro}^{0} - \sum_{i \in N_v} v_{pro}^{i}$$

v_{req}^{k}: Represents the requested value of the resource from node k, which is the node where the resource is being searched.

V: Denotes the total value of the requested resource.

v_{pro}^{0}: Represents the value of the resource that the current node can provide or supply.

N_v: Refers to the number of visited nodes by the current node.

v_{pro}^{i}: Represents the value of the resource that the i-th neighbor node can provide or supply, where i is a specific neighbor node.

Neighbor-Aided Provisioning Agents. These agents perform resource discovery in the neighboring nodes of the requested node, which are present in the requested node's adjacency list but are not directly accessible. In this case, the requested node is the output node that provides the resources.

Proxy Neighbor-Aided Provisioning Agents. These agents perform resource discovery in the neighboring nodes of the requested node, accessible from outside the requested node. In this scenario, unlike Neighbor-Aided Provisioning Agents, the node where the resource is found acts as the resource-providing node.

Multiple-Provisioning Agents. These agents perform resource discovery in the requested node and its neighboring nodes. As a result of the resource discovery process, several nodes and the resources they provide are identified. To limit the number of neighbor nodes to be visited in this approach, the parameter c_v (number of visited nodes) is included in the resource discovery request. The value of this parameter is incremented for each visited node until it reaches the value of the static parameter c_{max} (maximal number of nodes to be visited). If the values of c_v and c_{max} are equal, then the resource discovery process is considered complete for the requested node.

Conclusion

In conclusion, this paper examined a multi-agent-based approach for resource discovery in distributed exascale computing systems. The agents were categorized based on two fundamental concepts, allowing each agent to specialize in the search for specific types of resources. Furthermore, agent types have been categorized to provide resource provisioning based on the behaviors of agents in neighboring nodes. This approach thoroughly investigated the challenges associated with each agent type. The findings of this study contribute to the understanding of resource discovery

mechanisms in distributed computing systems and provide insights for improving the efficiency and effectiveness of resource allocation.

References

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

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

Castano, S., Ferrara, A., Montanelli, S., & Racca, G. (2004, April). Matching techniques for resource discovery in distributed systems using heterogeneous ontology descriptions. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.* (Vol. 1, pp. 360-366). IEEE.

Dimakopoulos, V. V., & Pitoura, E. (2003, August). A peer-to-peer approach to resource discovery in multi-agent systems. In *International Workshop on Cooperative Information Agents* (pp. 62-77). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ding, S., Yuan, J., & Hu, L. (2005, March). A heuristic algorithm for agent-based grid resource discovery. In *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service* (pp. 222-225). IEEE.

Kambayashi, Y., & Harada, Y. (2007). A resource discovery method based on multi-agents in P2P systems. In *Agent and Multi-Agent Systems: Technologies and Applications: First KES International Symposium, KES-AMSTA 2007, Wroclaw, Poland, May 31–June 1, 2007. Proceedings 1* (pp. 364-374). Springer Berlin Heidelberg.

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

Raj, P., & Sekhar, C. (2020). Comparative Study on CPU GPU and TPU. *Int. Journal of Computer Science and Information Technology for Education, 5,* 31-38.

Selvaraj, C., & Anand, S. (2012). A survey on security issues of reputation management systems for peer-to-peer networks. *Computer science review, 6*(4), 145-160.

Tan, Y., & Zheng, Z. (2009, May). A multi-agent based resource discovery scheme for p2p systems. In *2009 International Workshop on Intelligent Systems and Applications* (pp. 1-4). IEEE.

Tan, Y., Han, J., & Wu, Y. (2010). A multi-agent based efficient resource discovery mechanism for grid systems. *Journal of Computational Information Systems, 6*(11), 3623-3631.

Yamasaki, J., & Kambayashi, Y. (2010, November). Design an Implementation of Bee Hive in a Mult-agent Based Resource Discovery Method in P2P Systems. In *2010 First International Conference on Networking and Computing* (pp. 292-293). 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, 11*3, 127-166.

Submitted: 16.02.2023
Accepted: 26.05.2023