Computer Fraud and Security
ISSN (online): 1873-7056

Resource Oriented Architecture for Cloud and big data with Resource Workflow Management to Support Secure by Design Operation

Mardhani Riasetiawan[1], Ahmad Ashari[2], Bambang Nurcahyo Prastowo[3]
[1,2,3]Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia

Abstract:
Introduction: The advent of cloud computing and big data technologies has revolutionized how data is stored, processed, and analyzed. However, the rapid expansion of these domains has introduced significant challenges in resource management, particularly in achieving high-performance computation

Objectives: Resource-Oriented Architecture (ROA), a paradigm rooted in efficient allocation, monitoring, and utilization of distributed resources, has emerged as a promising solution for addressing these challenges. This study delves into the application of ROA for cloud and big data management and computation, focusing on optimizing resource allocation, ensuring scalability, and enhancing computational efficiency.

Methods: The research develops the methods of task/job identification by utilizing the metadata which can use the cloud and big data workflow components to determine resource allocation. The identification process carried out on a task/job. The cloud and big data workflow components consist of task management, resource capacity management, communication and transition, and interaction.

Results: This paper underscores the potential of Resource-Oriented Architecture as a transformative approach for cloud and big data management. By enabling precise, scalable, and efficient resource management, ROA addresses the key challenges associated with high-performance computation in these domains.

Conclusions: By focusing on dynamic resource allocation, task management, and efficient data processing, ROA enables organizations to optimize their use of cloud resources while handling the immense volume, variety, and velocity of big data.

Keywords: resource oriented architecture, cloud, big data, workflow, secure by design.

INTRODUCTION

The exponential growth of data and the increasing demand for scalable, efficient computational infrastructures have placed cloud computing and big data analytics at the forefront of modern technology. Cloud computing offers on-demand access to vast pools of computing resources, while big data analytics enables organizations to derive meaningful insights from massive datasets. Together, these technologies power applications ranging from real-time decision-making to complex simulations and artificial intelligence (AI). However, the dynamic and distributed nature of cloud environments [24], coupled with the complexity of big data workflows [25], presents significant challenges in resource management, performance optimization, and system reliability [26].

Resource-Oriented Architecture (ROA) [28] has emerged as a paradigm that redefines how resources are conceptualized, provisioned, and utilized in cloud and big data ecosystems. Unlike traditional monolithic architectures, ROA emphasizes the decomposition of functionalities into modular resource components, enabling fine-grained control and enhanced adaptability. While the promise of ROA is compelling, its implementation in cloud and big data systems is fraught with critical issues that must be addressed to fully realize its potential for high-performance computation.

One of the primary challenges is the dynamic nature of workloads in cloud environments. Applications often experience fluctuating demands, with some requiring high computational power and others needing significant memory or storage. Traditional resource allocation methods struggle to adapt to these changes in real time, leading to issues such as resource underutilization, over-provisioning, or performance degradation. ROA introduces the possibility of dynamic resource provisioning based on granular resource allocation policies [29], but this demands advanced predictive models and real-time decision-making algorithms, which are yet to reach maturity. Another pressing issue lies in resource heterogeneity. Modern cloud and big data platforms host a diverse range of hardware and software resources, including CPUs, GPUs, memory, storage systems, and specialized accelerators. Managing and orchestrating these heterogeneous resources to maximize performance and efficiency requires an architecture that can abstract and unify resource control. ROA proposes a modular approach, but implementing it at scale involves overcoming compatibility issues, ensuring interoperability, and minimizing latency in resource orchestration. Scalability and elasticity are also key concerns. While cloud platforms are inherently designed for scalability, the efficiency of scaling operations heavily depends on the underlying resource management framework. ROA, by design, supports modular scaling, but this scalability can be hindered by network bottlenecks, inefficient load balancing, or unoptimized resource allocation algorithms. Similarly, balancing workloads across distributed environments is a complex challenge, especially in big data applications that demand high throughput and low latency. ROA-based systems must integrate advanced scheduling techniques and real-time monitoring to maintain optimal workload distribution.

Fault tolerance and reliability [30] remain critical issues, particularly in distributed systems where failures are inevitable. The decoupled nature of ROA facilitates failover mechanisms, but ensuring seamless recovery without compromising performance requires robust checkpointing, redundancy strategies, and error detection mechanisms. Furthermore, security and data privacy concerns, exacerbated by the distributed nature of cloud and big data platforms, present additional challenges. Resource-level security policies, encryption, and secure access control mechanisms must be embedded within ROA to protect sensitive data and computational integrity.

In this context, this study explores the potential of Resource-Oriented Architecture as a framework for addressing these challenges in cloud and big data environments. By focusing on resource management strategies that prioritize scalability, flexibility, and performance, the research aims to establish ROA as a foundational paradigm for high-performance computation. The remainder of this paper discusses existing approaches, analyzes the integration of ROA with modern technologies, and presents a roadmap for overcoming current limitations. Ultimately, the study seeks to provide a comprehensive understanding of how ROA can reshape the landscape of cloud and big data resource management to meet the demands of the future.

CHALLENGES

Cloud and big data a critical components in delivering internet-based services, acting as the provider of infrastructure. It offers physical resources such as servers, networks, storage, power support, and operational management to ensure functionality. Cloud and big data facilitate infrastructure and operational systems that simplify management, hardware maintenance, and operational sustainability. It supports dynamic capacity expansion, enabling adaptability to user demands [1]. Operating a cloud and big data entails challenges in providing reliable and high-performance services within specific timeframes. It requires maintaining uninterrupted operations and managing resources optimally [2]. Reliability refers to the cloud and big data's ability to offer services, manage them effectively, and maintain execution time consistency [3]. High performance is essential for allocating resources accurately to meet data processing requirements [1]. The resources in cloud and big data include processing capacity (processors), memory space, bandwidth, and the ability to execute instructions within a given time frame [1]. The cloud and big data critical aspect lies in the availability of infrastructure capacities—processing, networking, and storage—impacted by resource capacity and allocation methods [1]. The processing, networking, and storage capacities represent vulnerabilities due to limitations in component specifications and capacities. These limitations prevent the cloud and big data from regularly scaling, responding to on-demand requirements, and achieving automation due to the need for adaptive methodologies and approaches. Resource capacity management becomes another vulnerability when allocation patterns and resource management are misaligned with the workload characteristics of the cloud and big data. Performance degradation can occur when the cloud and big data are configured based on default operating system setups without customized tuning or configuration suited to service characteristics. Performance decline can result in unfulfilled jobs/tasks requested by users [4]. The cloud and big data utilize information on jobs/tasks and resource availability to maximize service delivery. The research emphasized that configuration plays a crucial role in managing resource capacities, allocation mechanisms, and service strategies in cloud and big data [5]. Resource capacity serves as key information for managing jobs/tasks, including service provision scheduling, and ensuring adequate resource support for execution [6]. Resource allocation, based on task identification, requires mechanisms that can be effectively implemented within a cloud and big data [7]. In cloud and big data, adaptive resource management aligns with user service demands and component availability. Resource capacities and user needs dynamically change depending on service characteristics [8]. These characteristics include traffic information, loads, and user access patterns [9]. Resource management and service identification are foundational factors in determining resource allocation strategies in a cloud and big data. Resource allocation strategies are classified based on objectives identified in the research, including service-oriented, energy usage, business-oriented (cost-benefit), and performance-oriented approaches. The focus is on enhancing resource productivity and operational security in a cloud and big data to deliver requested services. Strategies prioritize comparing service levels with economic considerations (cost-benefit) using decision-support models [10],[5]. Implementations often involve "pay-as-you-use" principles, emphasizing resource utilization within specified durations. The Energy Usage Approach Aims to deliver resource efficiency without excessive power consumption. Studies [11] and [1] have explored green energy approaches using job submission analysis, energy-saving evaluation, and virtual machine placement strategies. The Performance-Oriented Approach: Maximizes available capacity to enhance service delivery. Techniques include load balancing, predictive elasticity, auto-scaling, and metadata management for job/task allocation. Each strategy requires task/job identification to understand workload characteristics and resource needs. They leverage task/job information to configure workflows and components within the cloud and Big data, ensuring adaptive responses to processing demands. Developing task/job identification methods is critical for enhancing resource allocation strategies, and enabling cloud and big data to sustain service continuity effectively. Figure 1 shows the landscape of resource management that is explained in this section.

MarketNet [12] was introduced as a market-based job distribution system using a cost-offering model for resource allocation. MarketNet facilitates interaction between resource demand and cost offerings for job queues. The system executes tasks based on resource demand while considering traffic loads and managed resources, minimizing internal risks. Nimrod-G [13] was introduced as a modeling and parameterization tool for automatic resource allocation in grids. It employs a broker model where users utilize varying costs and capacities of available resources. Performance depends on the quality of service (QoS) requested by users, with Nimrod-G striving to maximize QoS. Stanford Peers [14] is a peer-to-peer framework enabling resource allocation negotiation through data exchange models between tasks and available capacity. The exchange model provides information about resource capacity and offers it to users. Users can submit initial requests for their required capacity. The barter function facilitates resource trading, aligning requested capacity with availability, thereby minimizing costs, particularly for data exchange between remote sites. OpenPEX [15] is a resource allocation system that uses a reservation-based approach for virtual resources. OpenPEX allows users to reserve virtual machines that can be activated at specified times and operated over different periods. It negotiates resources with other users to ensure task continuity and resource access.

Cluster on Demand [16] is a service-oriented architecture designed for computational needs. It features independent virtual clusters that distribute workloads across different groups. Each virtual cluster job is managed by a cluster broker, utilizing tender and economic models. Users only need to assign tasks to the cluster agent, and each broker manages multiple tasks with specified execution times. MOSIX [17] is a distributed operating system designed for high-performance clusters. It employs opportunity cost to minimize overall cluster costs, using a commodity-based model to calculate processor and memory costs for each process. Cluster nodes work collaboratively to reduce costs and execute assigned tasks efficiently. CloudBus [18] is a toolkit that provides market-based resource allocation strategies, enabling both physical and virtual distributed implementations. CloudBus includes a cloud broker offering a general framework for various cloud platforms. It integrates economic models, such as commodity, tender, and auction strategies, and functions as a plug-in for middleware technologies like Unicore and Globus.

D'Agen [19] is a mobile agent system for task management in distributed computing. It implements a proportional tendering system where agents negotiate with resources. Resources are allocated proportionally if there are multiple requests, as shown in Table 1. From the above discussion, resource allocation models can be identified as tendering, commodity, auction, bartering, proportion-shared auction, and posted price. These models are implemented across various architectures, including clusters, peer-to-peer networks, mobile agents, distributed information systems, grids, and clouds. Cloud architecture is the most widely adopted and incorporates multiple resource allocation models. This literature review provides a strong foundation for the resource allocation architecture and models used in this study, further supported by the growing trend of cloud-oriented data centers.

Table 1 highlights the focus on resource issues, challenges, and models that form the framework of this research. Based on the review of resource allocation architectures, this study focuses on the commodity model within a cloud architecture**.** Resource allocation strategies relate to the methods and processes used to allocate resources to meet processing needs based on tasks/jobs. The research [20] researched QoS-based Scheduling of Workflows on Global Grids. Yu explained that Grid computing has emerged as a global cyber infrastructure, integrating large-scale, distributed, and heterogeneous resources. Managing workflows in Grid or Data Center environments involves handling complex processing and distributed resources such as computing devices, data, applications, and scientific instruments. Yu's research presented workflows and algorithms for mapping resource allocation in Grids based on specific QoS (Quality of Service) parameters. The approach proposed a taxonomy of workflow management systems for Grid computing, developing workflows utilizing tuple spaces to manage execution activities. Genetic algorithms were used to schedule workflows with QoS considerations. Yu emphasized the importance of resource allocation in Grid and Cluster environments within Data Centers, highlighting the need for workflow management systems among Data Center components to implement strategies effectively. The study leveraged multi-objective evolutionary algorithms, offering alternative scheduling through trade-offs between tasks and resource capacities.

TABLE I
RESOURCE MANAGEMENT MODEL

Model	Cluster	Peer to Peer	Mobile Agents	Distributed information	Grid	Cloud
Tendering	V				V	V
Commodity	V				V	V
Auction/tendering		V			V	V
Bartering		V				V
Proportion shared auction			V			
Posted Price				V		V

The research [21] proposed a market-oriented meta-scheduler called Meta-Broker. This component coordinates resource requests and allocates the best resources to users based on monetary and performance costs. The findings showed significant cost reduction and increased throughput by adopting the Meta-Broker approach. The Meta-Broker uses a semi-decentralized architecture, where scheduling decisions are made by the Meta-Broker, while job submission, execution, and monitoring are delegated to users and middleware providers. Garg's research introduced a market-oriented meta-scheduling algorithm designed to maximize service utilization. The market-oriented algorithm considers QoS requirements for multiple users by mapping tasks to heterogeneous resources. Garg's work contributed significantly by addressing QoS requirements for all users and maximizing resource utilization. However, coordinated scheduling led to overloading tasks from heterogeneous sources, highlighting the need to exploit the heterogeneity of resources and Data Center capacity to meet user needs. The research [22] focused on workflow architecture in Data Center components by proposing improved scheduling algorithms tailored for managing data-intensive applications. Pandey's study provided a comprehensive mapping of scheduling techniques, integrating data management components into the architecture. Implementation was tested on Functional Magnetic Resonance Imaging (fMRI) applications and Evolutionary Multi-objective Optimization Algorithms within a distributed Grid architecture. The study proposed heuristic algorithms considering data transfer time and costs. Contributions include workflow architecture enhancements by handling multisource parallel data transfers, tested using real-world testbeds in static and dynamic environments. Pandey's findings complement Garg's research by validating testbeds in real Data Center environments.

The research [1] reviewed and explored cloud computing approaches for Data Centers, emphasizing resource management challenges. Teng's study combined scheduling theory with Data Center hierarchy to address diverse cloud service needs, proposing a game-theoretical algorithm to predict job/task bidding and auction pricing (capacity). Using Bayesian Learning, resource allocation achieved Nash Equilibrium even among non-cooperative users. Teng's research addressed task scheduling at the system level with an online schedulability test implemented on MapReduce. It highlighted the relationship between cluster utilization and map-reduction ratios in MapReduce, while also introducing an online evaluation model for probabilistic tests to assess overall system utilization. However, Teng's approach is limited to specific MapReduce and cluster environments, requiring further implementation in heterogeneous and real-world Data Center environments. The research [23] has proposed a resource allocation strategy in virtual machines using a re-packaging approach for trade-offs in cloud scalability, both horizontally and vertically. The study analyzed performance outcomes of different VMS strategies, combining re-packaging with auto-scaling for virtual machines. It identified optimal sets, transition policies, reconfiguration costs, and decision-making mechanisms. The approach integrated vertical and horizontal elasticity for Virtual Management Systems, emphasizing scalability in service-oriented Data Center models. Xu et al. (2015) introduced a load-balancing approach for scalable, adaptive metadata server clusters in cloud file systems. They proposed Cloud Cache as an adaptive and scalable load-balancing mechanism using adaptive diffusion and replication to collect load characteristics. Cloud Cache implemented adaptive cache diffusion and replication schemes to enhance load-balancing performance efficiently. Resource allocation in cloud computing environments can be based on user preferences [5]. Cloud computing operates on a pay-as-you-go model, where users pay based on usage behavior over a specified period. This study introduced a market-driven tender mechanism to allocate resources based on user capacity preferences. A system model consisting of resource allocation units, auction subunits, and payment subunits was tested in a CloudSim environment with single and multiple users. The study highlighted that demand-driven preferences in resource allocation determine commodities. Research on resource allocation strategies underscores their aim to meet service needs through tasks/jobs. Data Centers and their resources must be responsive, adopting service-oriented approaches. Resource allocation methods based on demand-driven and commodity models align with this study's focus on developing resource allocation strategies based on service characteristics.

A dynamic resource allocation algorithm with a hierarchical approach was implemented in a Cloud computing environment by Wang (2015). This research utilized multiple cloud nodes to represent a big data environment. The algorithm was developed using fuzzy pattern recognition to assign tasks and nodes based on computational power and storage factors. Tasks and nodes were evenly distributed, aligning with the developed method. Resource management, such as data storage and processing, was presented by Kin et al. (2015). Their research focused on clouds integrated with the Internet of Things (IoT). Resource management with high availability, scalability, and high-speed processing was achieved using the Efficient Resource Management Schema (ERMS). ERMS employed XML-based standards for data storage and classification techniques to interact with data generated by IoT. A resource allocation approach through modeling in cloud computing centers was presented by Vakilinia et al. (2015). This research developed a performance model for cloud computing systems to handle multiple incoming jobs and distribute them for execution in Virtual Machine Systems (VMS). Each job consisted of several tasks with varying characteristics. The study dynamically determined and distributed tasks to VMs based on job variables, which were foundational for distributing workloads to VMS and resources. Research on dynamic resource allocation strategies for servers/VMS, enabling allocation and deallocation without downtime, was conducted by Wolke et al. (2015) using simple bin-packing heuristics. The study combined server/VMS allocation controllers with periodic reallocation, achieving high efficiency in energy benchmarks compared to other environments. A service-oriented approach to virtual machine placement in support of data centers was presented by Tseng et al. (2015). The research focused on VMS placement mechanisms using integer linear programming. A tree algorithm was used to minimize communication costs and achieve economical construction costs. A Forest Algorithm formulated using graph theory and a best-fit algorithm, was also employed. Cloud Data Centers require scalable and flexible resources. Virtualization offers examples of operating systems interacting with existing resources in virtual services. Virtual machine placement and positioning are crucial to resource management. Moschakis and Karatza (2016) introduced a meta-heuristic optimization approach. Their research dealt with scheduling processes in task-to-task applications within heterogeneous and multi-level clouds. This approach addressed characteristics of current big data challenges. The study modeled multi-cloud environments with a mid-cloud operator acting as a global service. The service distributed tasks to cloud operators as entry points, followed by cloud sizing to simulate dynamic cloud environments. Cloud sizing enabled VM performance modeling based on the created model, VM performance, and cost. Other parameters included application models utilizing parallel task-to-task jobs. Pop and Potop-Butucaru (2016) examined resource management for cloud computing. An adaptive perspective emphasizes resource management for stability, organization, and system independence in cloud environments. Resource management addressed fault tolerance, reliability, and availability in distributed systems. The business perspective focused on maximizing service value while minimizing costs. The technical perspective included virtual machine configuration, job scheduling, resource allocation strategies for cloud federation, dynamic data distribution, performance, and energy efficiency. Sedaghat et al. (2016) studied cloud data center resource allocation with reconciliation and decentralization using emergent behavior and topology. The research introduced the consolidation of multiple applications on a single physical machine (PM) in cloud data centers to enhance utilization, reduce energy consumption, and minimize costs. A resource management framework was proposed, consisting of a placement scheduler that enabled durable, cost-effective, energy-aware, and risk-oriented allocation. The studies focused on developing resource allocation strategies for Data Centers. Resource allocation models and architectures employed cloud-clustered, commodity-oriented architectures. The research focused on service provision scheduling by developing mechanisms to identify job/task characteristics based on metadata from submitted files. This identification yielded metadata classification information based on volume, variety, veracity, and velocity. The approach implemented workflow resource allocation for task identification and management components, communication mechanisms, transitions between components, and their interactions. Performance testing was conducted using real-world Data Center testbeds under peak load conditions during data migration. The research contributed significantly by introducing the Self-Assignment Data Management method for identifying job/task characteristics and adding task identification components to Data Center workflows. Performance testing compared configurations for dedicated and dynamic resource allocation, topologies, and peak loads. The evaluation was conducted by comparing metadata grouping using match-aggregate-pipeline with MapReduce, performance across Cluster, Grid, and High-Performance Computing architectures, and comparisons with other studies. Table 2 highlights comparisons between the aforementioned studies.

TABLE II
SELECTED RESOURCE MANAGEMENT RESEARCH

Methods	Metadata	Scheduling	Workflow	VM	XML based	Testing		Evaluation
						Real Case/testbed	Simulation	
QoS and Genetic Algorithm	-	V	V	-	-	V	-	Quality of services
Meta-scheduler and QoS	V	-	V	-	-	V	-	Quality of service
Scheduling algorithm and workflow architecture	-	-	V	-	-	V	-	Schedulability test
Scheduling	-	-	V	-	-	-	V	Schedulability test
Re-packaging for cloud-scale trade-off	-	V	V	-	-	-	V	Performance Index
Load balancing & metadata	-	V	-	-	-	-	V	Performance Index
Tender and capacity	-	V	V	-	-	-	V	Quality of services
Node cloud and fuzzy	V	-	-	-	-	-	V	-
Efficient resource management system (ERMS)	-	-	-	-	V	V	-	Quality of service
Dynamic VM distribution	-	-	-	V	-	V	-	Performance Index
Dynamic VMS with simple bin packing heuristic	-	V	-	-	V	-	V	Performance Index
tree algorithm	-	-	-	-	V	-	V	Performance Index
meta-heuristic optimization	-	-	-	-	V	-	V	Schedulability test
adaptive resource allocation	-	V	-	-	V	V	-	Performance Index
reconciliation and integration of resource allocation	-	V	-	-	-	V	-	Performance Index

A RESOURCE ORIENTED ARCHITECTURE

The workflow in Figure 2 for resource allocation in a Cloud and Big Data requires several components to function effectively, including entities, execution management, entity communication, transition processes, and algorithms. The workflow for resource allocation involves the following components Task Identification**.** This component identifies each data or file as a task by accessing information provided by the XML metadata from the SADM process. Task Management and Prioritization; This component manages task queues and prioritization based on information from task identification. It also negotiates resource availability with the resource capacity management component. Resource Capacity Management; This component provides the necessary resources to execute tasks that have been managed by task management. Task identification utilizes this information for task submission, functioning like job registration, which will then be processed by task identification.

Task identification determines the tasks, parameters, and dependencies. Tasks refer to jobs that must be completed by the cloud and big data. The 4V Parameters represent commodity information, while dependency provides information about tasks with special characteristics. Task identification performs several key functions, including accessing and receiving data or file information in XML and content formats. It identifies each data or file by describing them in two XML formats that Provide content and attribute information for each data or file to be processed. The XML Big Data offers information about the characteristics of each data or file based on classification structures, including volume, variety, veracity, and velocity.

The Task Management (TM), task, parameter, and dependency information is received and passed on to the task manager. The task manager organizes tasks using the following components Task will be responsible for managing the task queue. Monitor and oversee the task registration process, execution, and completion. Resource Group will retrieve resource allocation information from resource capacity management. In Resource Capacity Management (RCM)**,** the process begins with the initialization of resource availability in the cloud and big data. This initialization is followed by resource registration, which declares resources as available and ready for use. Subsequently, resource capacity management waits for incoming tasks from the resource group via the queue and requirements of each task. Once registration and requirements are approved, the Service Provision component is responsible for executing the tasks and sending the task results back to task management.

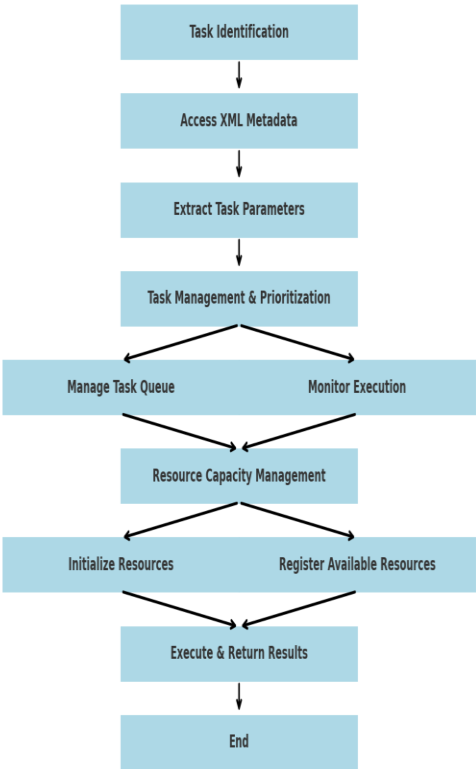

Fig. 2. The Workflows

In execution management managed by task management in Figure 3, tasks are executed in parallel. Execution management is carried out based on the input from received tasks. In this case, execution management can be implemented using **a** one-to-one**,** one-to-many**, or** many-to-many model. This mechanism employs an event-driven approach based on a subscription-notification system, which simplifies control and execution management.

The task manager responsible for execution management relies on task information provided by the status of task processing in the monitor component. Event management serves as a central component, replacing the function of the event service. This allows the component to communicate directly with tasks in a head-to-head manner, bypassing the task manager. This approach enables the resolution of simultaneous tasks and supports a more flexible communication pattern. In task management transitions as shown in Figure 4, the initiation phase starts from the initial state, followed by a registration event. After the registration event, which provides event status information, the first task activation begins. The initially activated task enters monitoring, which performs several functions. Each task executed is monitored by task management. If an event failure occurs, task management proceeds to failure processing and terminates the task. Task management also checks resource availability through workflow execution checks until completion. Once no more tasks are running, the transition concludes. The transition in tasks begins with a registration event that follows a waiting mechanism. This is followed by event processing, during which resource matching is carried out based on the 4V parameters. If suitable resource allocation is achieved, job submission takes place. Job submission is then monitored up to the output process.

Once the job status is marked as completed, the task transition process concludes, as shown in Figure 5. In the resource allocation mechanism, interactions occur among tasks (task1, task2, and task n), task identification, task management, resource capacity management, and the cloud and big Data. The Cloud and Big Data initiates and

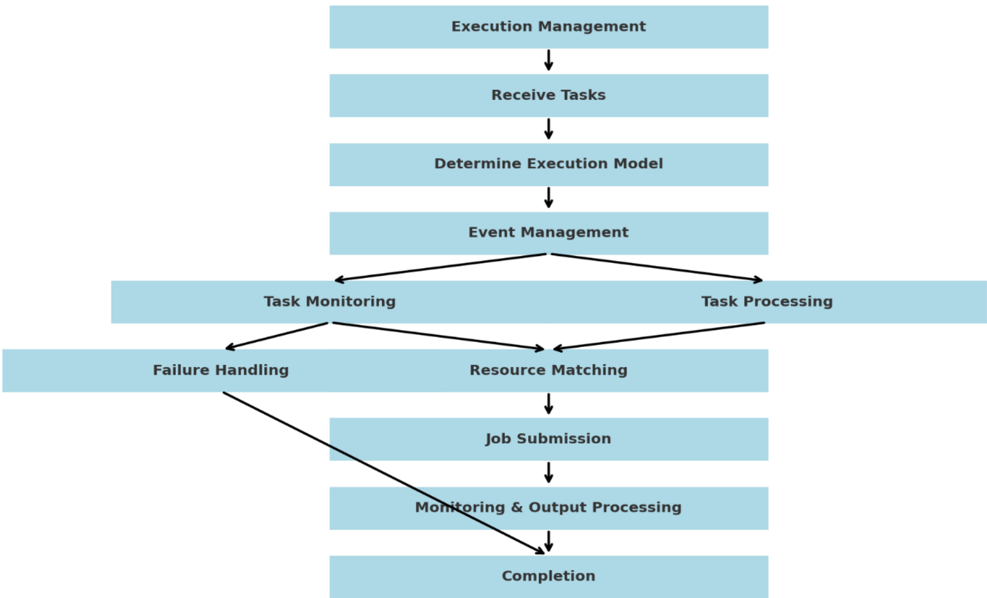

Fig. 3. Event Management

registers resources so they can be accepted by resource capacity management as ready-to-use services and resources. The first incoming task is received by task identification in a specific queue. Task identification gathers characteristic information for each task. Subsequent tasks are also received by task identification in their respective queues, and so on. Task identification then forwards the tasks to task management with specific requirements. These requirements are passed on to resource capacity management to acquire the necessary resource services. The Cloud and Big Data receiving the service request allocate resources and hand them over to resource capacity management and task management as part of the service provision. The service provision component informs task identification that the service is active and ready for use. This information is then relayed to the user, who proceeds to execute their respective tasks.

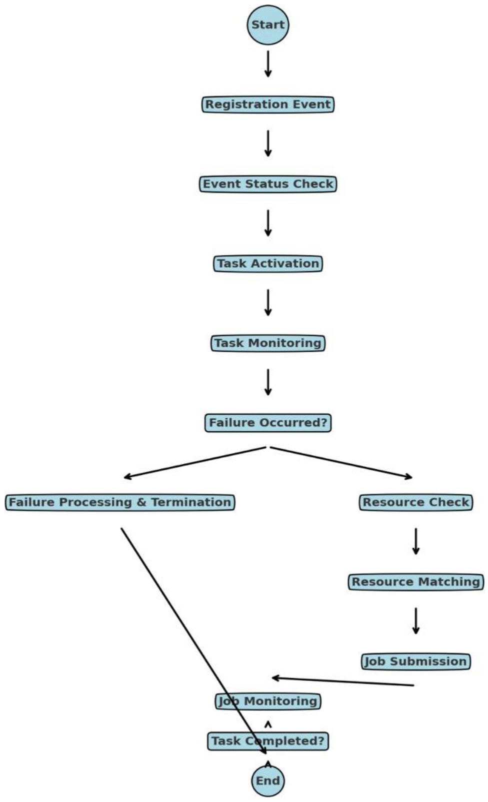

Fig. 4. The Task Management Transition

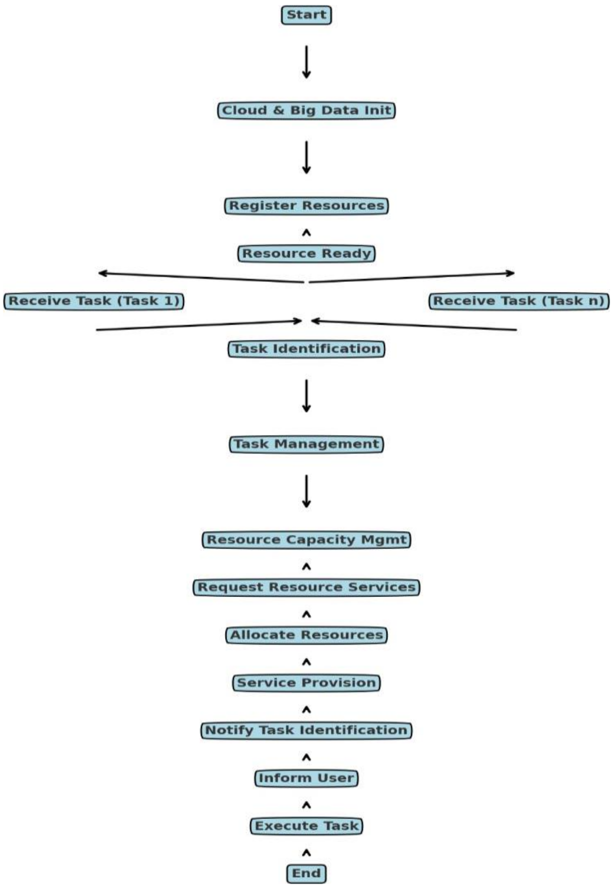

Fig. 5. The Task Transition Process

CONCLUSIONS

In conclusion, the study of Resource-Oriented Architecture (ROA) for cloud and big data management and computation plays a pivotal role in addressing the complexities of modern data-driven systems. By focusing on dynamic resource allocation, task management, and efficient data processing, ROA enables organizations to optimize their use of cloud resources while handling the immense volume, variety, and velocity of big data. However, significant challenges persist, including scalability, performance optimization, energy efficiency, and ensuring the reliability and security of distributed systems. Overcoming these challenges requires the development of advanced algorithms, decentralized architectures, and energy-aware mechanisms, alongside improved data management strategies. As the field evolves, future directions should focus on AI/ML-driven resource management, the integration of edge and fog computing, and the adoption of standardization to facilitate interoperability. Additionally, advancements in security, fault tolerance, and QoS management will be crucial to meet the demands of increasingly complex cloud environments. Ultimately, continued research and innovation in ROA will be essential to unlocking the full potential of cloud and big data technologies, providing scalable, efficient, and reliable solutions for the next generation of computing needs.

DISCUSSIONS AND FUTURE DIRECTIONS

The study of Resource-Oriented Architecture (ROA) for cloud and big data management faces several challenges and opportunities for future advancements. Scalability and performance bottlenecks remain significant issues, as managing the high volume, variety, velocity, and veracity of tasks and data require scalable architectures that avoid delays and inefficiencies. Resource utilization efficiency poses another challenge, with a constant need to balance between over-provisioning, which leads to resource wastage, and under-provisioning, which causes service delays. Dynamic resource allocation, particularly in multi-tenant environments, adds complexity, as it requires real-time adaptability to fluctuating workloads while ensuring Quality of Service (QoS). Additionally, energy efficiency in data centers is critical, as high energy consumption has environmental and operational cost implications, necessitating energy-aware resource allocation mechanisms. Data management complexities also present challenges, particularly in handling diverse formats, ensuring data integrity, and efficiently managing metadata for task prioritization. Fault tolerance and reliability in large-scale distributed systems remain essential to mitigate disruptions caused by hardware or software failures. Interoperability and standardization are ongoing challenges, as integrating diverse platforms and frameworks is hindered by a lack of universal standards while ensuring security and privacy in multi-tenant environments remains a pressing concern.

Future directions for ROA research include the development of advanced scheduling algorithms using AI/ML to predict workloads and optimize resource allocation. Decentralized architectures, such as blockchain-based resource management, can enhance transparency and reliability, while energy-aware mechanisms and the integration of renewable energy can address sustainability challenges. Enhanced data management strategies, including real-time analytics and improved metadata-driven task prioritization, are vital for handling large-scale, diverse data streams. The incorporation of edge and fog computing can reduce latency and enable localized processing, further extending the capabilities of ROA. Promoting standardization and interoperability through universal APIs and communication protocols will simplify integration across platforms. Security mechanisms like zero-trust architectures and homomorphic encryption will strengthen data protection, while autonomic and self-healing systems can ensure continuous operation with minimal human intervention. QoS-driven architectures that prioritize latency, throughput, and reliability will cater to diverse application requirements, and experimentation with quantum computing may open new avenues for solving complex resource allocation and computation challenges. Together, these advancements aim to enhance scalability, efficiency, and user-centric experiences in cloud and big data environments

ACKNOWLEDGEMENTS

The research has been supported by the Academic Excellence Program Universitas Gadjah Mada 2024.

REFRENCES

[1] Teng, F., 2012, Management Des Donnees Et Ordinnnancement Des Taches Sur Architectures Distributes, Desertation, Ecole Cenrale Paris Et Manufactures, Centrale Paris.

[2] O'Brien, L., Merson, P., Bass, L., 2007, Quality Attributes for Service-Oriented Architectures, *International*

Workshop on System Development in SOA Environments, p.3

[3] OAIS, Open Archival Information Standard, access at http://public.ccsds.org/publications/archive/650x0m2.pdf

[4] Ambrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M., 2009, Above the Clouds: A Berkeley View of Cloud Computing, *Technical Report No. UCB/EECS-2009-28*, University of California, Berkeley.

[5] Kumar, N., Saxena, S., 2015, A Preference-based Resources Allocation In Cloud Computing Systems, *in 3rd International Conference on Recent Trends in Computing 2015*. Procedia Computer Science, vol 57, pp. 104-111.

[6] Liu, C., 2014, A Cloud-computing-based Resource Allocation Model for University Resource Optimization. *International Journal of Grid Distribution Computing*, vol.7, issue 3, pp.113-122.

[7] Singh, S., Sarkar, M., Roy, S., Mukherjee, N., 2013, *Genetic Algorithm Based Resource Broker for Computational Grid* of the International Conference on Computational Intelligence: Modeling, Techniques and Applications pp. 572-580.

[8] Nguyen, N., Khan, M.M.H., 2015, A Codes-loop Context Aware Data Acquisition and Resource Allocation Framework for Dynamic Data Driven Applications Systems (DDAS) on the Cloud. *Journal of Systems and Software*, vol. 109, pp. 88-105.

[9] Palmieri, F., Fiore, U., Ricciardi, S., Castiglione, A., 2016, GRASP-based Resources Re-Optimization for Effective Nig Data Access in Federated Clouds. *Future Generation Computer Systems*, vol.54, pp.168-179.

[10] Srinidhi, B., Yan, J., Tayi, G. K., 2015, Allocation of resources to cyber-security: The effect of misalignment of interest between managers and investors. *Decision Support Systems*, vol.75, pp.49-62.

[11] Kaushik, A., Vidyarthi, D. P., 2014, A Green Energy Model for Resource Allocation in Computational Grid. *The Computer Journal*, vol.58, issue 7, pp.1532-1547.

[12] Yemini, Y., Dailianas, A., Florissi, C., Huberman, G., 2000, MarketNet: Protecting Access to Information Systems Through Financial Market Controls. *Decision Support Systems*, vol.28, Issue 1-2, pp.205-2016.

[13] Abramson, D., Buyya, R., Giddy, J., 2002, A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. *Future Generation Computer Systems*, vol.18, pp.1061-1074.

[14] Bawa, M., Cooper, B. F., Crespo, A., Daswani, N., Ganesan, P., Gracia-Molina, H., Kamvar, S., Marti, S., Schlosser, M. Sun, Q., Vinograd, P., Yang, B., 2003, Peer-to-Peer Research at Standford. *ACM SIGMOD Record*, vol. 32, Issue 3, pp.23-28.

[15] Venugopal, S., Broberg, J., Buyya, R., 2009, Openpex: An Open Provisioning and Execution System for Virtual Machines, *Technical Report CLOUD-TR-2009-8*, CLOUDS Laboratory, The University of Melbourne, Australia, 2009.

[16] Rahul, L., Sharavana, K., Deployment of Virtual HPC Clusters on Demand from Volunteer Computing

[17] Resources. *International Journal of Advances Research in Computer and Communication Engineering*, vol.3, Issue 4, pp.6266-6268, 2014.

[18] Mishra, S., Kushmaha, D. S., Misra, A. K., 2011, Hybrid Reliable Load Balancing with MOSIX as Middleware and Its Formal Verification Using Process Algebra. *Future Generation Computer Systems*, vol. 27, pp.506-526.

[19] Buyya, R., Murshed, M., 2002, Gridsim: A toolkit for The Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, *Concurrency and Computation: Practice and Experience*, 14(13): 1175-1220.

[20] Rani, M., Nayak, R., Vyas, O.P., 2015, An Ontology-based Adaptive Personalized e-learning System, Assisted by Software Agents on Cloud Storage. *Knowledge-Based Systems*, vol.90, pp.33-48.

[21] Yu, J., 2007, QoS-based Scheduling of Workflows on Global Grids, *Dissertation*, Department of Computer Science and Software Engineering, The University of Melbourne, Australia.

[22] Garg S.K., 2010, Meta Scheduling for Market-Oriented Grid and Utility Computing, *Dissertation*, Department of Computer Science and Software Engineering, The University of Melbourne, Australia

[23] Pandey, S., 2010, Scheduling and Management of Data Intensive Application Workflows in Grid and Cloud Computing Environments, *Dissertation*, Department of Computer Science and Software Engineering, The University of Melbourne, Australia.

[24] Sedaghat, M., Rodriguez, F. H., Elmroth, E., 2016, Decentralized Cloud Data Center Reconsolidation

Through Emergent and Topology Aware Behavior. *Future Generation Computer Systems,* vol. 56, pp.51-63.

[25] Faheem, U., Shagun, D., Xiaoyu, X., Ali, B.M., 2024, Evaluation of distributed data processing frameworks in hybrid clouds, Journal of Network and Computer Applications, Volume 224,2024, 103837, ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2024.103837.

[26] Srđan, D.S., Nikola, T., Darko E., 2023, Big data BPMN workflow resource optimization in the cloud, Parallel Computing, Volume 117, 103025, ISSN 0167-8191, https://doi.org/10.1016/j.parco.2023.103025.

[27] Oumaima, G., Said, E.K., Mohamed, H., 2024, Adaptive workload management in cloud computing for service level agreements compliance and resource optimization, Computers and Electrical Engineering, Volume 120, Part A, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2024.109712.

[28] Li, L., Chen, J., Cui, H., Huang Tao, Liu, Y., 2015, Resource scheduling virtualization in service-oriented future Internet architecture,

[29] The Journal of China Universities of Posts and Telecommunications, Volume 22, Issue 4, Pages 92-100, ISSN 1005-8885, https://doi.org/10.1016/S1005-8885(15)60672-0.

[30] Reen, P, Supriya, M., 2022, Dynamic resource provisioning for service-based cloud applications: A Bayesian learning approach, Journal of Parallel and Distributed Computing, Volume 168, Pages 90-107, ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2022.06.001.

[31] Jian-De, L., Sying-Jyan, W., Katherine, S.L, Tsung-Yi H., 2023, Design-for-reliability and on-the-fly fault tolerance procedure for paper-based digital microfluidic biochips with multiple faults, Integration, Volume 89, Pages 185-196, ISSN 0167-9260, https://doi.org/10.1016/j.vlsi.2022.11.013.