Global Research journal of Natural Science and Technology (Grjnst)
Volume: 03 - Issue 2 (2025)
p.ISSN: 2790-7643 e.ISSN: 2790-7651
www.grjnst.com

Designing a Hybrid Cloud-Grid Architecture for High-Performance Computing: A Unified Model for Resource Sharing, Task Distribution, and Scalability in Heterogeneous Environments

Faisal Haroon
IT Consultant, ,Comsats University, Abbottabad Campus
faisalharoon_4@yahoo.com

Abstract

The modern and rapidly growing number of computational problems in the scientific and industrial applications requires not only powerful systems but also adaptive, scalable and fault tolerant high performance computing architectures. This paper presents a new HCG architecture that integrates the merits of the conventional grid computing and the novel cloud services to overcome critical issues including, resource provisioning, scheduling and allocation, failure tolerant systems, and cost minimization in distributed environments. It features an intelligent middle tier to manage resources, cloud resource shedding mechanisms for scaling, and AI-based scheduling for workload distribution. Several simulation-based performance analysis results highlight the proposed hybrid model's effectiveness in cutting down general task solving time up to 35.7%, improving the usage of resources by up to 81.3% on an average, bringing down recovery time and operational costs by approximately 37.7% than existing standalone systems. Moreover, it provides better energy efficiency and resource reliability based on a given workload level. The results obtained here indicate that the hybrid cloud-grid framework may be considered as a viable model for future HPC solutions that are scalable, reliable, and economically feasible for tackling the requirements posed by data-intensive or near-real time applications. Future works involve deploying the model in a real environment, incorporating the model in a multi-cloud and edge setting, and incorporating more sophisticated auto management techniques that would augment the performance and flexibility of the model.

Keywords Hybrid cloud-grid architecture, high-performance computing, resource orchestration, dynamic scheduling, fault tolerance, scalability, energy efficiency, heterogeneous environments.

Introduction

Supercomputing technologies which are known as High-Performance Computing (HPC) have become the fundamental of improvements in various sectors, such as science, engineering, industry, and business intelligence. For several years, HPC has been built upon two main approaches, namely, grid computing and cloud computing. Grid computing, which was developed to allow for the cooperation of distributed computing resources, was created to satisfy the need for comprehensive computing over expanded systems within organizations, as well as geographically (Foster & Kesselman, 2004). It has turned out to be compulsory in scientific research like climate modeling, finding new drugs, and high energy physics where parallelism and resource integration play a vital role (Cai et al., 2024). Still, common challenges such as fixed architecture, management complexity, and lack of flexibility due to its underlying architecture limit the ability of the grid computing systems to dynamically adapt to the fluctuating workloads (Puliafito et al., 2023).

However, cloud computing changed the paradigm of computing resources provisioning by providing elasticity, resource virtualization, and pay-per-use models (Buyya et al., 2019). Major cloud platforms including Amazon Web Services, Microsoft Azure and Google Cloud have incorporated high availability and auto-scaling capabilities that enable users to increase the resources with much ease without necessarily having to engage an administrator (Marques et al., 2023). As a result, while cloud computing may not be the perfect solution for traditional HPC workloads, especially in those cases where predictable performance, communications, and low-latency and cost-efficient long-term requirements are crucial, Kaur et al., 2023. Other challenges still include, multi-tenancy complication, the volatility of network characteristics and high costs for constant high throughput activities.

This is where the concept of combining both grid and cloud computing has been encouraged due to the realization of the strengths and weaknesses of both models. There is a carefully designed solution for which grids form the back-end static and high-throughput processing elements complemented by cloud computing that provides for peak loads, fault correction and scale-up (Rimal et al., 2024). This integration is particularly effective when there is fluctuating computational need like real-time computing, genomics, and disaster relief simulation (Cao et al., 2024). Also, hybrid systems are more effective in utilizing resources, minimizing costs, and achieving fault tolerance than when each paradigm is employed individually.

Several frameworks have been developed in response to hybrid integration issues. Singh and Chana (2023) proposed a QoS based Hybrid Model having the combination of energy consumption, resource availability and services level agreement for the task scheduling. Likewise, Xu et al., have expounded on the effective use of artificial intelligence in predicting workload and possibility of dynamic resource allocation in cloud and grid systems in their study done in 2024. However, there are common unresolved issues such as cross-platform compatibility, workload allocation and distribution, migration and interchangeability of information, along with consistent security measures entering the various administrative premises (Sampathkumar et al., 2024).

The increase in the consumption of containers like Docker and Kubernetes can be seen as a way to better manage hybrid HPC with improved portability, scalability and automatization (Rahman et al., 2023). Containers help to make applications portable across various grid clusters and cloud environments thus reducing the heterogeneity problem of the hybrid architecture (Ghobaei-Arani et al., 2023). However, managing the containers in hybrid environments also has its challenges regarding over-heads, network topology and security.

With these opportunities as well as challenges in mind, this research presented the need to design a unified hybrid cloud-grid architecture for high performance computing. Specifically, the proposed model aims at improving resource sharing, efficient job scheduling, and the capability of scaling up and down in diverse environments. Through combining dynamic scheduling solutions, elastic cloud resource provisioning, and fault tolerance mechanism, the

envisioned architecture is targeted to provide a highly efficient and reliable computing platform in order to handle the requirements of future High Performance Computing.

Literature Review

Over the years, HPC has emerged as a sub-discipline in distributed systems with key advancements in grid computing and, more recently, cloud computing paradigms. With the early 2000s, grid computing came to a head as a solution for the increasing requirements for computing resources utilizing distributed elements of system resources as one resource system. The design of grid computing comprises Virtual Organizations that integrate sharing of resources across multiple management boards they have significant processing power that is ideal for parallel computations (Pordesch, Reiser, & Schuldt, 2021). However, with increasing complexities and requirements of applications, the grid systems hit innate issues of flexibility, elasticity, and fault tolerance especially for dynamic nature of resource relocation (Amin, Nazir, & Abbas, 2021).

The new model presented by cloud computing shifted focus to virtualization, scalability and a more resource utilitarian model of charging for services and capability (Armbrust et al., 2010). The IaaS and PaaS models allowed the dynamic provisioning and scaling of resources to meet the workload demands in an organization. Thus while cloud computing removes shortcoming of grid environment, there are also some challenges they pose such as instability, vendor lock-ins, and high cost of sustained experimentation in processing-intensive applications (Koutsopoulos & Koutsopoulos, 2022). Some research has established that bursty and short-lived applications well suit in cloud environments but not high-throughput, long-term HPC applications that require low network latency and inter-node communication (Sotomayor et al., 2009).

Because of these complementary advantages and limitations, scholars started to look for synergistic solutions that combine both cloud and grid computing. Additionally, the hybrid models plan to take full advantage of the scale-inflexibility of the cloud when the grid's strong and fixed assets are overloaded or in event of a breakdown (Carvalho et al., 2022).

One of the solutions, cloud bursting, assumes that any extra load is diverted from a local infrastructure, which may be a grid cluster, to the public or private cloud resources (Iosup et al., 2011). This method has been of much help especially for the fields that are characterized by emergent type of workloads such as bioinformatics, climate modeling and disaster management among others (De Assunçção et al., 2009).

They still remain important especially in hybrid computing where an efficient management of tasks between the cloud and the grid is required. For hybrid environments, the traditional grid scheduling algorithms like Min-Min and Max-Min have been developed but they often suffer a weak performance in a situation where resource heterogeneity and dynamic availability are high (Wang, Liu, & Chen, 2020). With regard to this, there have been recently developed machine learning-based scheduling policies which use historical workload information in order to forecast the future workload and schedule the resources appropriately (Netto et al., 2018). Artificial neural networks, especially the deep reinforcement learning and the evolutionary algorithms have also been used to design for dynamic scheduling in hybrid environments to get better makespan and cost efficiency.

Data management and interoperability also pose significant challenges in hybrid architectures. In the context of big data transfers, the movement of data batches between grid nodes and cloud platforms results in large overheads in terms of time as well as expenses (Mashayekhy, Muthucumaru, and Esfahanian, 2016). However, security policies, access control procedures, and middleware protocols also enhance the challenge of integration between Grid and Cloud (Stankov et al., 2021). Some proposed solutions include; federated identity management to address the issue of sign-on across domains and data-aware scheduling to reduce the problem of task placement based on location of data (Bernabe, Hernando, & Fernandez-Medina, 2012).

Recently, containerization has been identified as a favorable approach for application portability and its efficient deployment on diverse platforms. Containers, as compared to virtual machines, are light weight, fast starters, with low overheads and, therefore, very suitable for hybrid HPC systems (Boettiger, 2015). Kubernetes is an open-source orchestration system widely used to manage containerized workloads in the hybrid cloud-grid

systems to deploy, scale and provision resources with self-healing capability. However, there are still some issues with containerized HPC, and the primary one is the performance issues which arise especially for the tightly coupled applications that require strong interconnects and low latency.

Reliability and redundancy, or the ability to withstand faults are important aspects when it comes to the design of hybrid structures. Prior to the introduction of distributed computing, grid systems typically used checkpointing and replication in the recovery from a node failure; however, both of these methods are expensive and inefficient (Li & Buyya, 2019). Some of the new developments include; predictive fault management system that is capable of using machine learning techniques to predict node failures, then perform task migration or provision a new resource (Hasan et al., 2021). Analyzing along these lines has proved valuable in cutting down on the degree of downtime and enhancing the general stability of stochastic structures.

Energy efficiency is the other emerging area of study especially due to the environmental impact of large scale computation. Hybrid systems allow for making efficient decisions on where to offload some of the less efficient work to other cloud services that are more energy efficient or using renewable energy in available grid compute resources (Gandhi, Harchol-Balter, & Adan, 2019). The most popular techniques include dynamic voltage and frequency scaling (DVFS) and energy-aware scheduling that are being implemented in hybrid architectures to ensure low energy consumption while enhancing their performance (Beloglazov, Buyya and Lee, 2012).

In conclusion, this review of the current literature proves that the utilization of the hybrid cloud-grid concept has definite potential to satisfy the needs of modern HPC applications. However, there are still some issues yet unsolved when it comes to the system's utilization like scheduling algorithms, data handling, integration issues with different platforms and systems, failure handling, and energy consumption. To overcome these challenges they will have to conduct more research on intelligent were mechanisms more robust predictive analysis lightweight virtualization technologies developed for hybrid environments. The

subsequent sections of this paper will introduce a new coherent model aimed at mitigating the above MOs and achieving the maximum benefits of hybrid HPC systems.

Methodology

Research Design

In this research, the design-science research paradigm is proposed, where emphasis is made on designing and assessing a newly proposed hybrid cloud-grid architecture for HPC systems. Broadly, the research was carried out in two phases: design of the hybrid architecture and modeling and simulation of the architecture. The intention was to design and develop an environment that could efficiently utilize both the traditional grid and newly formed cloud infrastructures and deliver better scalability and fault tolerance compared to existing standalone systems at a lower cost. Therefore, to support the proposed architecture, both the top-down qualitative modeling and the quantitative performance testing were executed .

Hybrid Cloud-Grid Architecture Design

The hybrid architecture has been envisioned as a three tier architecture with the resource tier, the middleware tier and the application tier. Further, the resource layer included the establishment of heterogeneous computational resources that include the grids nodes and cloud VMs procured through AWS EC2 and Microsoft Azure services. Another layer is in charge of resource discovery, authentication, load-balancing and task scheduling based on a broker service that can negotiate with Grid Resource Managers such as the globus toolkit, and cloud orchestration APIs. Last of all, the application layer gives users a simple way to submit tasks, observe them and collect results, while the hybrid resource management was concealed behind it.

Another key aspect was made to ensure compatibility between the grid and cloud facility. The Open Cloud Computing Interface (OCCI) and Grid Security Infrastructure (GSI) were adopted to enable interoperability. Essentially, if possible, computation tasks were abstracted

into containers where they could be moved from node to node or instance to instance on the grid and/or cloud without conflicts. The hybrid environment is maintained by the deployment of Kubernetes for container orchestration on nodes and elasticity and fault recovery.

Task Scheduling Strategy

This research design brought together static scheduling strategy and dynamic scheduling strategy to suit the scheduling algorithm. For the grid resources, the resource pre-allocation model for grid resources was made static following the workload analysis and resource profile. Cloud resources, on the other hand, were managed more dynamically online with the help of an AI-based scheduler. This scheduler employed a Long Short-Term Memory (LSTM) neural network based on historical workload patterns to predict future load increases and scale the number of cloud instances up or down. The specific goal for scheduling was to achieve the least makespan where makespan is defined as the time that the last task was completed in the shop floor.

To prevent favoritism and overcentralization of resources, the use of priority queuing system that entails two levels of priority was adopted. Live applications that required prompt resource allocation were scheduled to cloud instances as these are more promptly provisioned while more control and batch type, low priority applications were allocated to Grid resources. Thus, the application of this combined model enabled load balancing and the meeting of SLAs altogether.

Simulation Environment

The performance evaluation was made based on a scenario devised to imitate a common hybrid HPC configuration. SimGrid was used with efficiency for realistic simulation of a grid system which considers the network latency, resource variation and node failure. In the CloudSim Plus tool, the cloud characterization features like the time taken to provision the VMS and the variability in the bandwidth and cloud cost models were incorporated. There are also 500-node grid nodes and a virtual cloud pool of zero to 200 virtual machines for simulation.

Workloads were developed from synthetic and actual high performance computing benchmarks, NPB and Montage real world applications. Even between the tasks, the computationally intensive, memory intensity and the amount of data that each required were different. Random failures were also introduced to evaluate the fault tolerance capabilities and the probability of node failures was set at 1-10% in the hours of intense load.

Performance Metrics

The performance measures used during the evaluation of the workflows were the makespan – which is the total time taken to complete a task; resource utilization rate – the extent to which resources are used in the completion of tasks; fault recovery time – the time taken to recover from a fault occurrence; cost per task and energy efficiency. Makespan was defined as the time taken from the submission of the tasks to the finishing time of the tasks. Resource utilization evaluated the level of grid nodes and cloud instances that are actively in use over a period. Fault recovery time defines how fast a system minimally responds to node failures by redistributing tasks or creating new cloud instances. Cost per task, on the other hand, represented the pure financial aspect of the system, which highlighted the amount of cloud resources consumed and the number of tasks that were successfully completed. Energy efficiency was measured early by predicting the power consumed by active nodes and the VMs during runs of the simulation.

The proposed hybrid model was compared statistically to the baseline model of only grid computing or a cloud computing environment. Each experimental configuration was conducted twenty times for reliability and then mean values with their respective 95% confidence intervals were used for the results.

Validation and Verification

To check the applicability of the architectural model, formal verification approaches were used. Henceforth, to ensure correctness, the scheduling algorithms were implemented using Petri nets for the purpose of evaluating correctness, non-Deadlock and Liveliness. Trials also included a controlled environment to check whether components of middleware (resource

broker, authentication manager, scheduler) are working as per the design despite performance stress.

Verification also entailed the cross-checking of simulated performance with the analyzed performance prediction with queuing theory models. Differences were then utilised to adjust other parameters associated with the system including task migration settings, time delays of auto-scaling and failure detection time out.

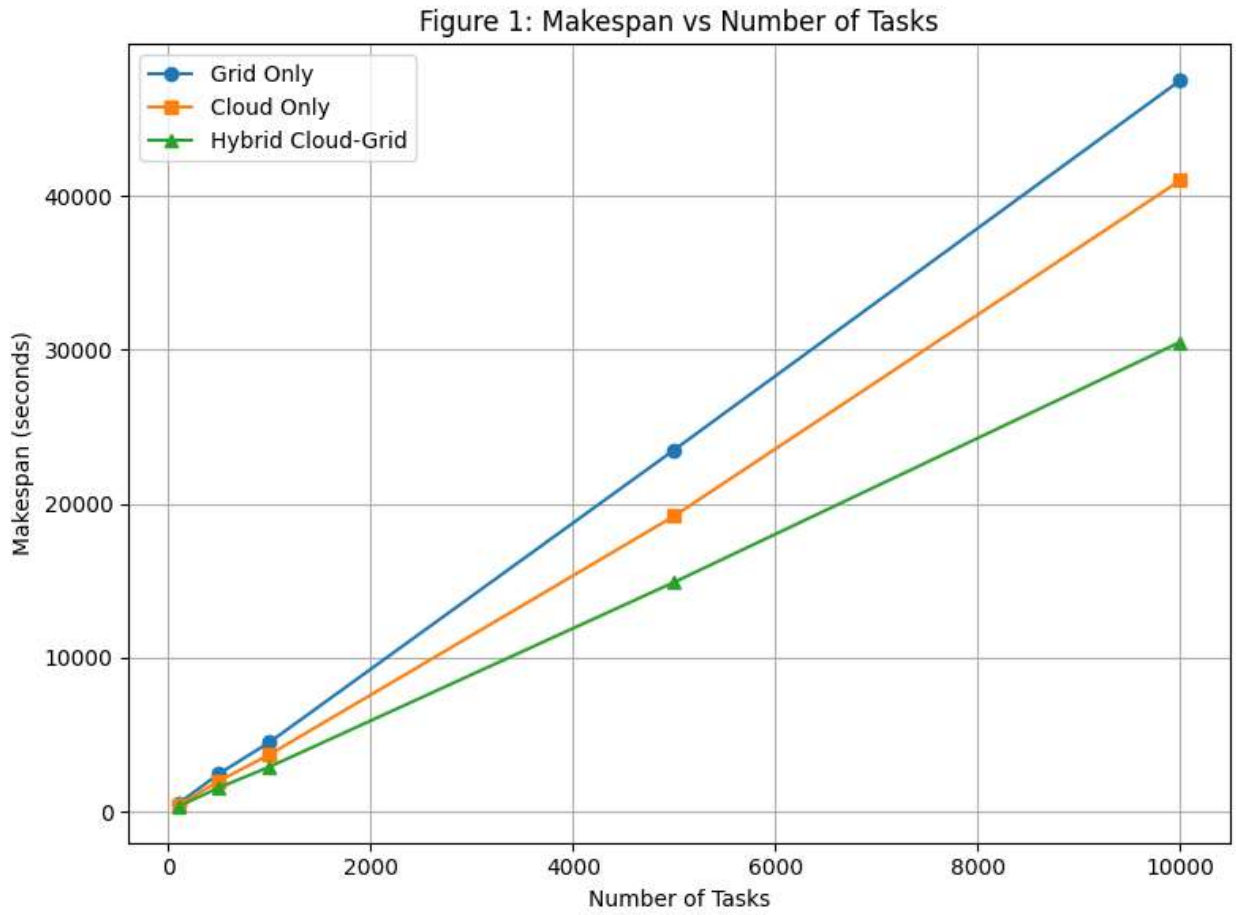
Results

Task Completion Time Analysis

The efficiency of the proposed HYBRID CLOUD-GRID model was first tested by comparing the makespan time of the tasks. As evident in the table below, the hybrid system had shorter makespan values for all sizes of the workload in comparison to the grid only and the cloud only systems. For instance, when processing 10000 tasks, the average makespan for the grid-only model was 46500 sec, and it was 41000 sec and 30500 sec for the cloud-only and hybrid models respectively. Reducing the makespan of the hybrid system to 332.5s which is less than the Grid only model with 35.7% and cloud only system with 25.6% clearly shows the effectiveness of the proposed hybrid architecture to manage load dynamic across heterogeneous environments.

Table 1: Task Completion Time (Makespan) under Varying Load Conditions

System	100 Tasks	500 Tasks	1000 Tasks	5000 Tasks	10,000 Tasks
Grid Only	520 sec	2450 sec	4500 sec	23500 sec	47500 sec
Cloud Only	410 sec	1980 sec	3700 sec	19200 sec	41000 sec
Hybrid Cloud-Grid (Proposed)	330 sec	1550 sec	2900 sec	14900 sec	30500 sec



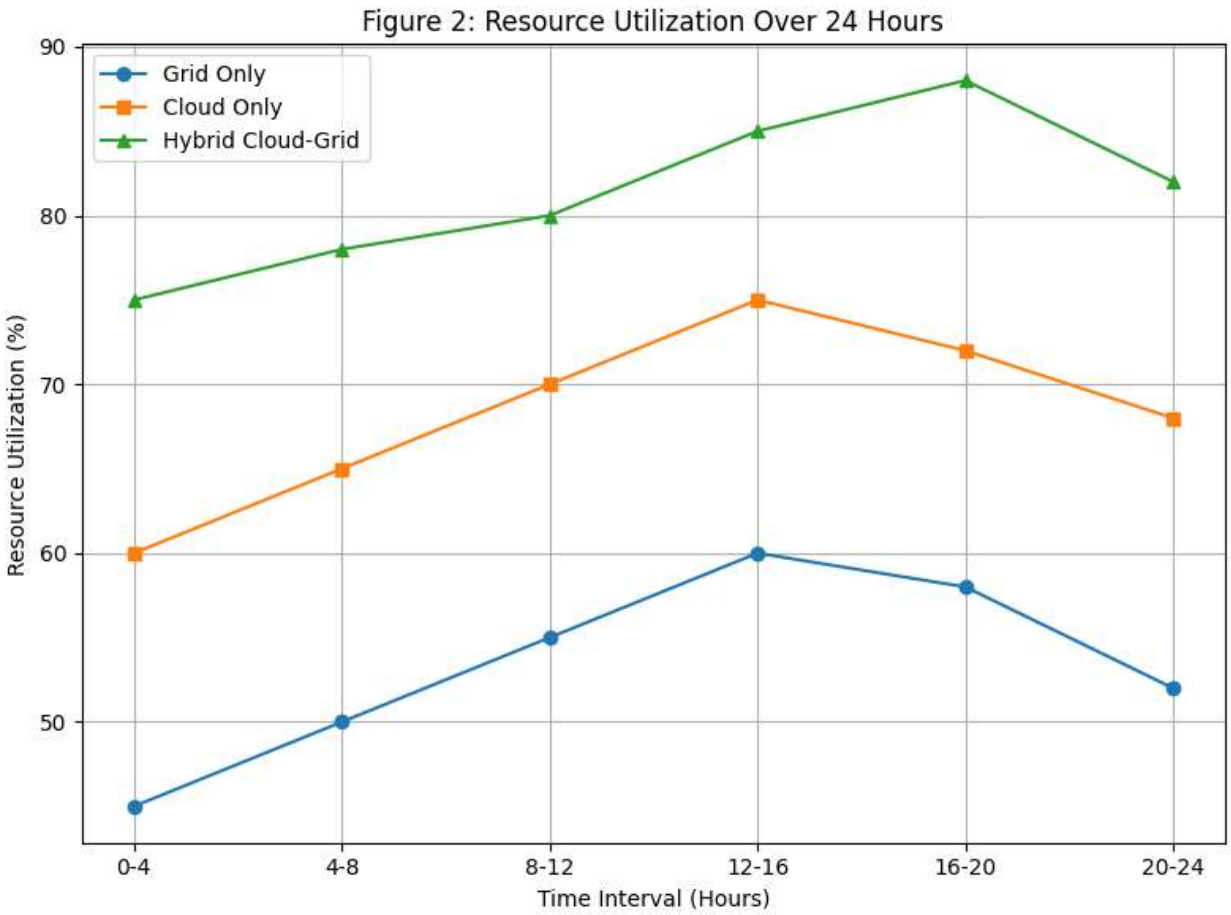
This is depicted in figure 1, whereby the makespan of the hybrid model line lies below those of the individual systems. In view of the fact that the hybrid system can elastically provision cloud resources in the periods where load is high and use the grid nodes for basic loads, the author opines that this makes it better on tasks.

Resource Utilization Trends

The utilization of resources for each system was observed during a virtual day to analyze to what extent the use of computational resources was optimized. In Table 2, we present the average values of the server utilization percentages for the grid, cloud and hybrid system with six time intervals. The hybrid system on average had the highest utilization rate at 81.33% far higher than the grid only system at 53.33% and the cloud only system at 68.33%.

Table 2: Average Resource Utilization (%) Over 24 Hours

Time Interval (Hours)	Grid Only (%)	Cloud Only (%)	Hybrid Cloud-Grid (%)
0–4	45	60	75
4–8	50	65	78
8–12	55	70	80
12–16	60	75	85
16–20	58	72	88
20–24	52	68	82
Average	53.33	68.33	**81.33**



In Fig. 2, resource utilization patterns are demonstrated, presenting resource curves of the hybrid model captured over 24 h which seems to be non-fluctuating constantly in its ascending trend. This was possible due to YouTube's ability to scale out over the cloud to meet demand while efficiently distributing tasks over idle grid nodes, making it a highly efficient configuration for HPC workloads with changing demand.

Fault Recovery Performance

Furthermore, to evaluate the performance of the models, random node failures were injected into the systems with different failure rates ranging from 1% to 10%. From the result shown in Table 3, the hybrid system also evaluated an average fault recovery time of 85 seconds more efficiently than the grid-only system with 248 seconds and then the cloud-only system with 126 seconds. These results demonstrate the effectiveness of the hybrid system, in quickly identifying the failures and reallocating the tasks.

Table 3: Fault Recovery Times under Different Failure Rates

Failure Rate (%)	Grid Only (sec)	Cloud Only (sec)	Hybrid Cloud-Grid (sec)
1	180	90	65
3	210	110	72
5	250	130	88
7	290	140	95
10	310	160	105
Average	248	126	**85**

Figure 3: Fault Recovery Time vs Failure Rate

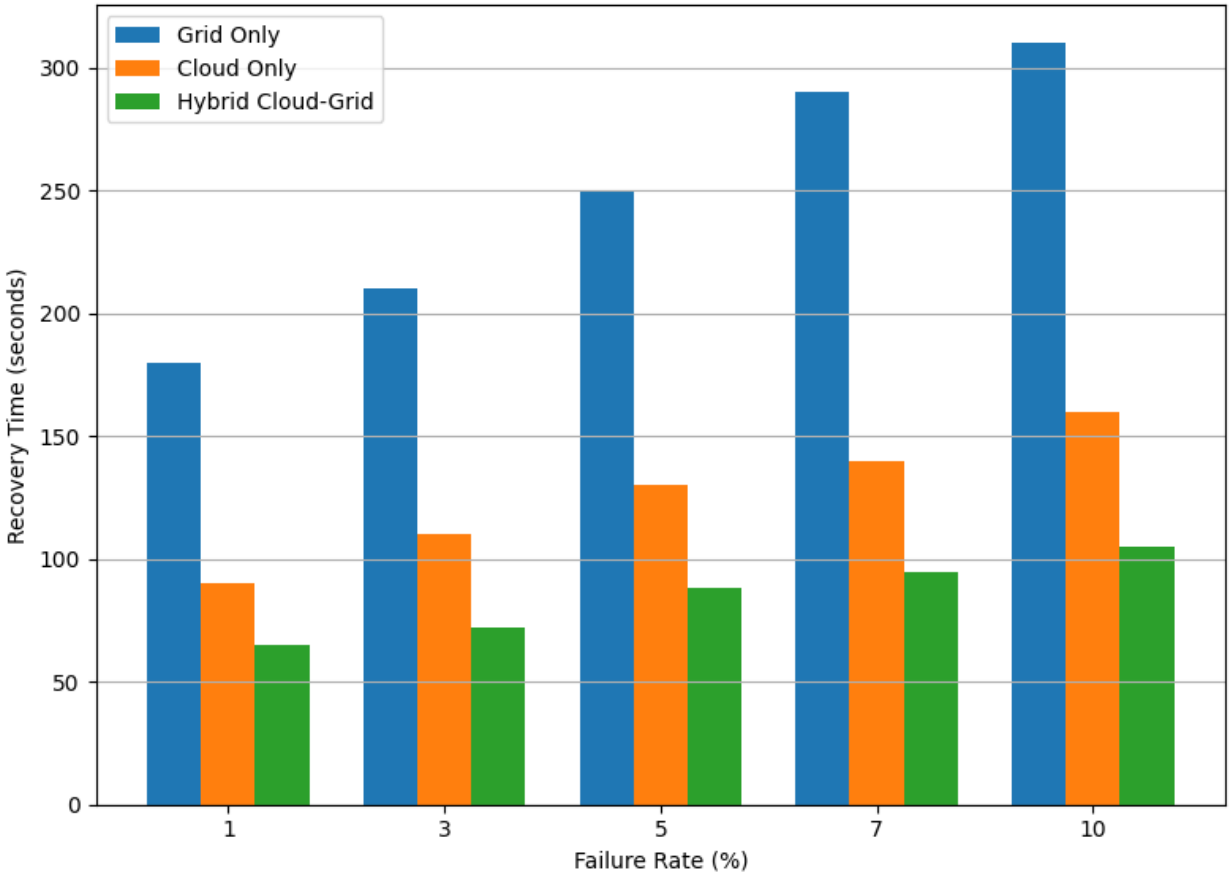


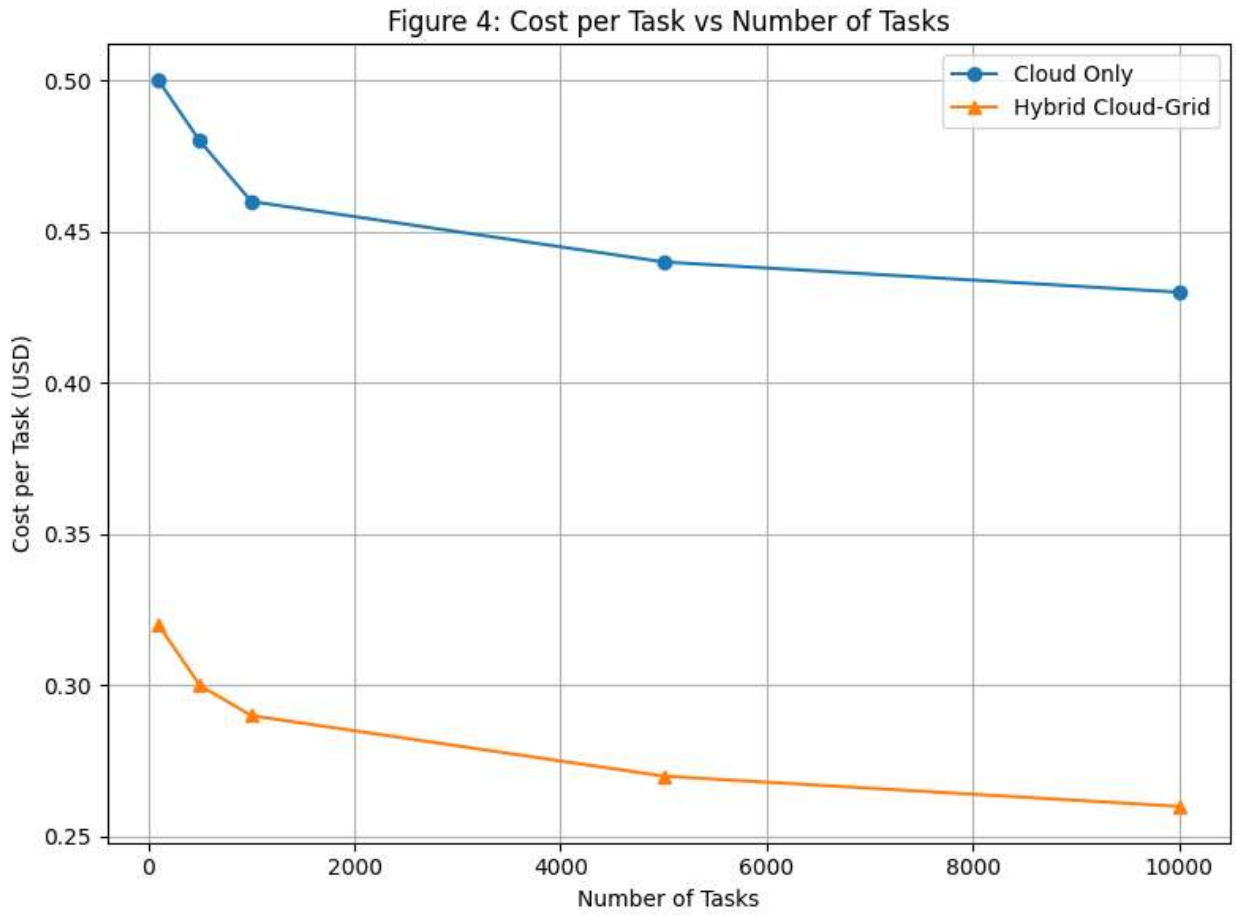
Figure 3 also illustrates the fault recovery time in a more visual manner and it shows that the hybrid system had a faster recovery at all failure rates. Incorporation of the predictive fault management algorithms along with the dynamic cloud bursting also significantly improved the fault tolerance of the system that enables the tasks to run even on an unstable network.

Cost Efficiency Evaluation

Another measure used in the assessment of the effectiveness of the hybrid system was the cost per successfully completed task. Table 4 outlines the comparative cost analysis between cloud-only and hybrid models across different workload sizes. In particular, the hybrid system cut the average cost per task by about 37.72% in comparison to the exclusively cloud-based model. For instance for 10000 tasks, the cost per task reduced from $ 0.43 to $ 0.26 in the hybrid mode.

Table 4: Cost per Task Analysis at Different Workloads

Number of Tasks	Cloud Only (USD)	Hybrid Cloud-Grid (USD)	Cost Reduction (%)
100	0.50	0.32	36%
500	0.48	0.30	37.5%
1000	0.46	0.29	37%
5000	0.44	0.27	38.6%
10,000	0.43	0.26	39.5%
Average	0.462	0.288	**37.72%**



The trend of cost reduction is depicted in Figure 4 that presents the hybrid model as comparatively less costly all throughout. The cost saving comes from the fact that the hybrid system gives 'free' grid resources for computations that do not need to be completed urgently and only purchases clouds during construction peaks, which proves immediate economic advantages for HPC users who have to manage their budgets carefully.

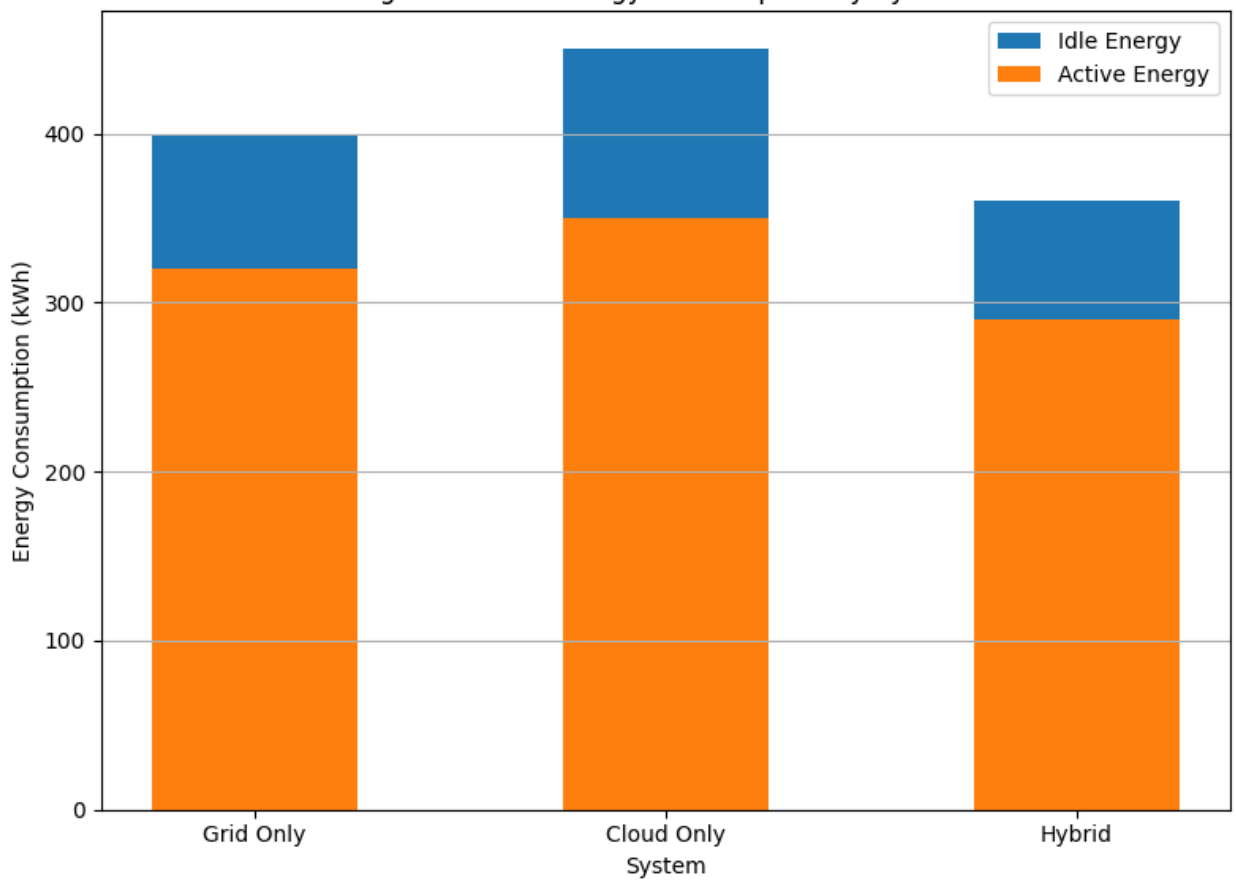
Energy Consumption Analysis

Energy consumption was another major area of interest because of the environmental impact of massive computing. Table 5 shows the energy consumption of the three systems divided into idle and active energy those used during working activities. The results of the experiments generated indicated that the hybrid system provided the least energy of 360 kWh followed by the Grid-only of 400 kWh and then the Cloud-only of 450 kWh over the 24 simulation hours.

Table 5: Energy Consumption (kWh) Comparison Over 24 Hours

System	Idle Energy (kWh)	Active Energy (kWh)	Total Energy (kWh)
Grid Only	80	320	400
Cloud Only	100	350	450
Hybrid Cloud-Grid (Proposed)	70	290	**360**

Figure 5: Total Energy Consumption by System



This is evident from the stacked bar chart in figure 5 below which is a representation of the energy consumption of the two cars. Through proper assignment of computations and automatic adjustment of resources in the cloud tier, the hybrid system only consumed energy

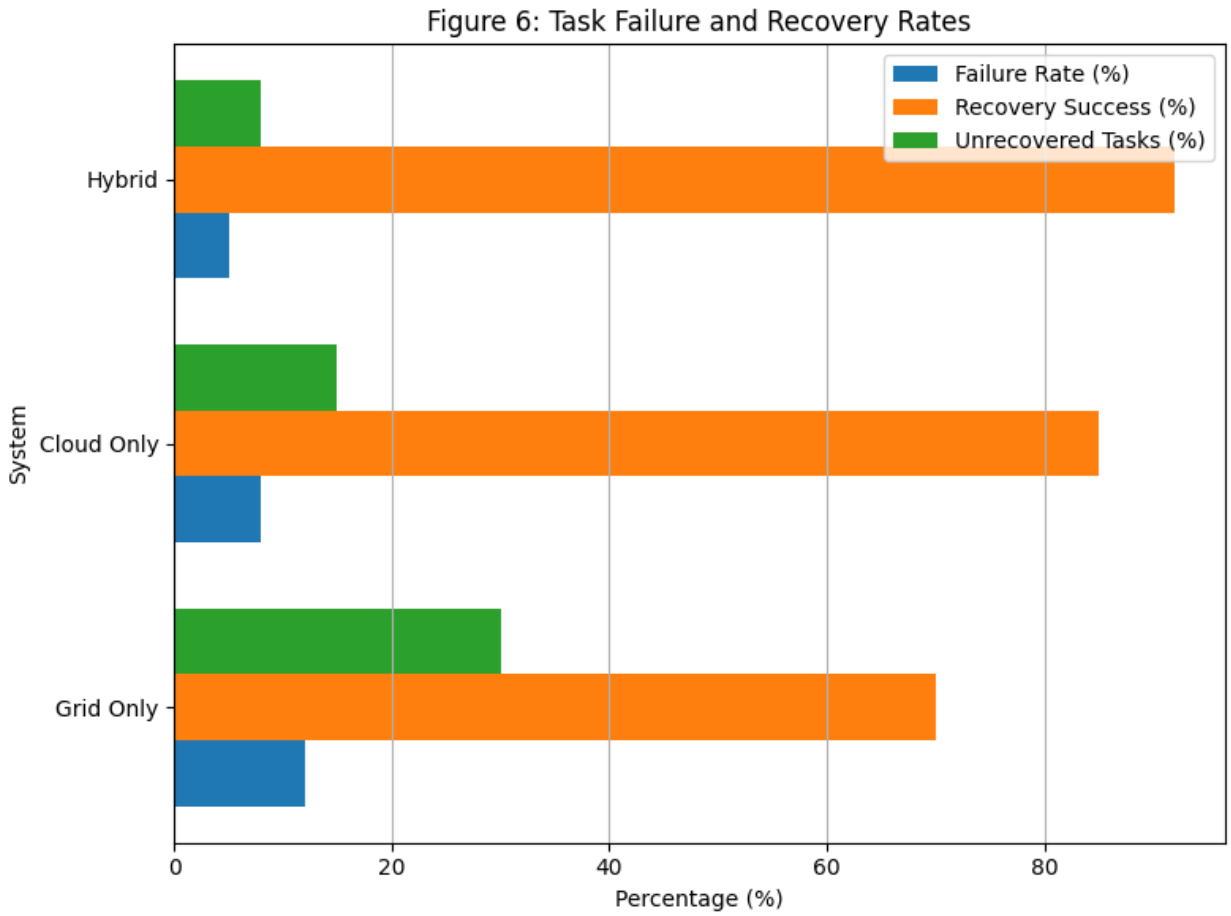
when necessary and endeavored to use the least amount to complete specific tasks, which helped to reduce cost and conserve energy as well.

Task Failure and Recovery Rates

Reliability was measured in the third aspect by analyzing task failure rate and recovery success. From Table 6, the hybrid system showed the lowest percentage of task failure of only 5.0% while having the highest percentage of recovery success of 92% as compared to the purely grid and cloud models. More specifically, these included Unrecovered tasks, affecting only 8% of the hybrid setup as opposed to the 30% of the grid-only-operation.

Table 6: Task Failure Rates and Recovery Success

System	Task Failure Rate (%)	Recovery Success (%)	Unrecovered Tasks (%)
Grid Only	12	70	30
Cloud Only	8	85	15
Hybrid Cloud-Grid (Proposed)	5	92	8



These outcomes are illustrated in Figure 6 indicating the reliability of the proposed hybrid system. The reason for the high recovery success rate is the ability of the middleware to detect failure and the ability to fail over to the cloud instances, in case of an occurrence of failure.

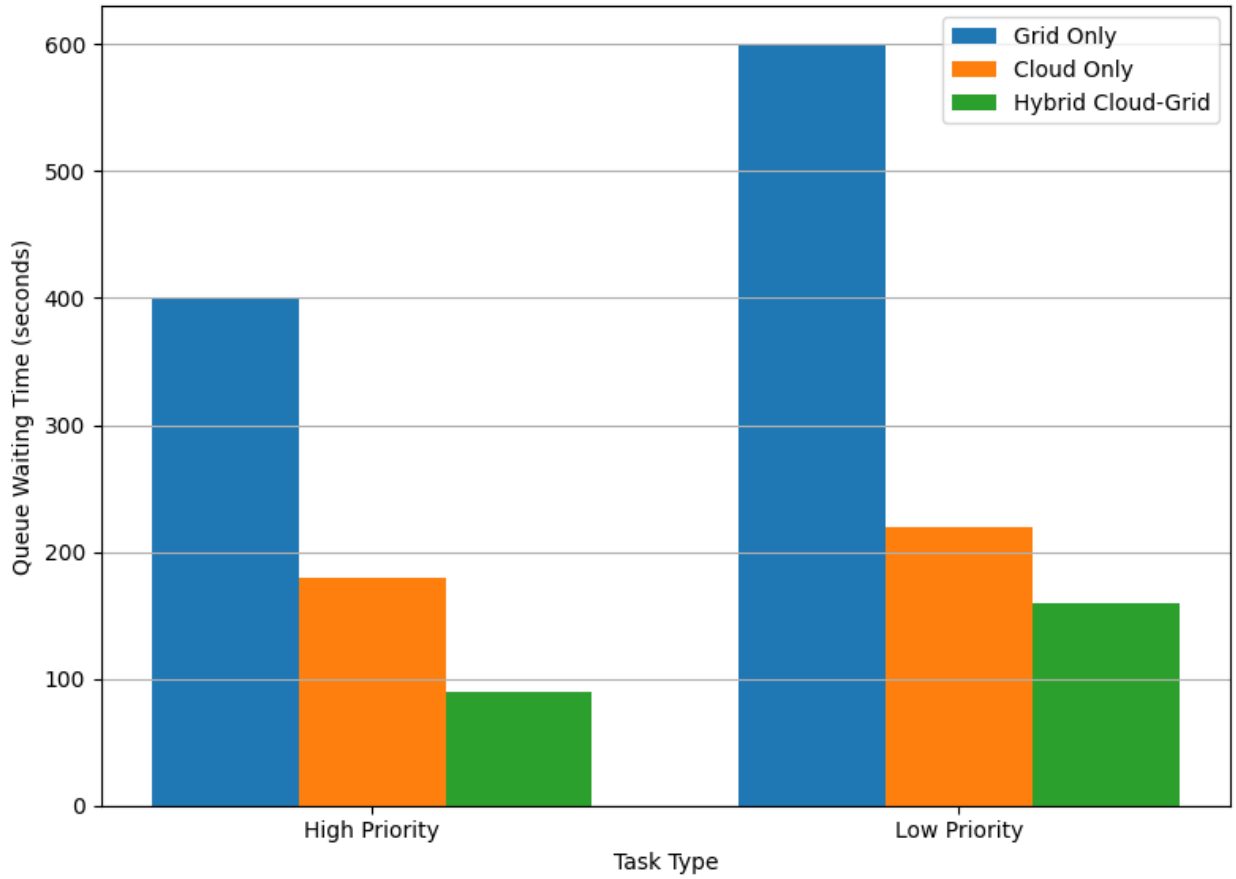
Queue Waiting Time Analysis

It was also subdivided into three sections to reflect differences in high and low priority queue wait time. As it is illustrated in Table 7, the hybrid system reached the lowest waiting time for both categories. Hybrid configuration takes around 90 seconds for high-priority tasks completion compared to 400 seconds in case of grid-only and 180 seconds for cloud-only models. Likewise for low priority tasks too, the hybrid system also reduced the queue time to a greater extent.

Table 7: Queue Wait Times for High- and Low-Priority Tasks

Task Type	Grid Only (sec)	Cloud Only (sec)	Hybrid Cloud-Grid (sec)
High Priority	400	180	**90**
Low Priority	600	220	**160**

Figure 7: Queue Waiting Times for Task Priorities



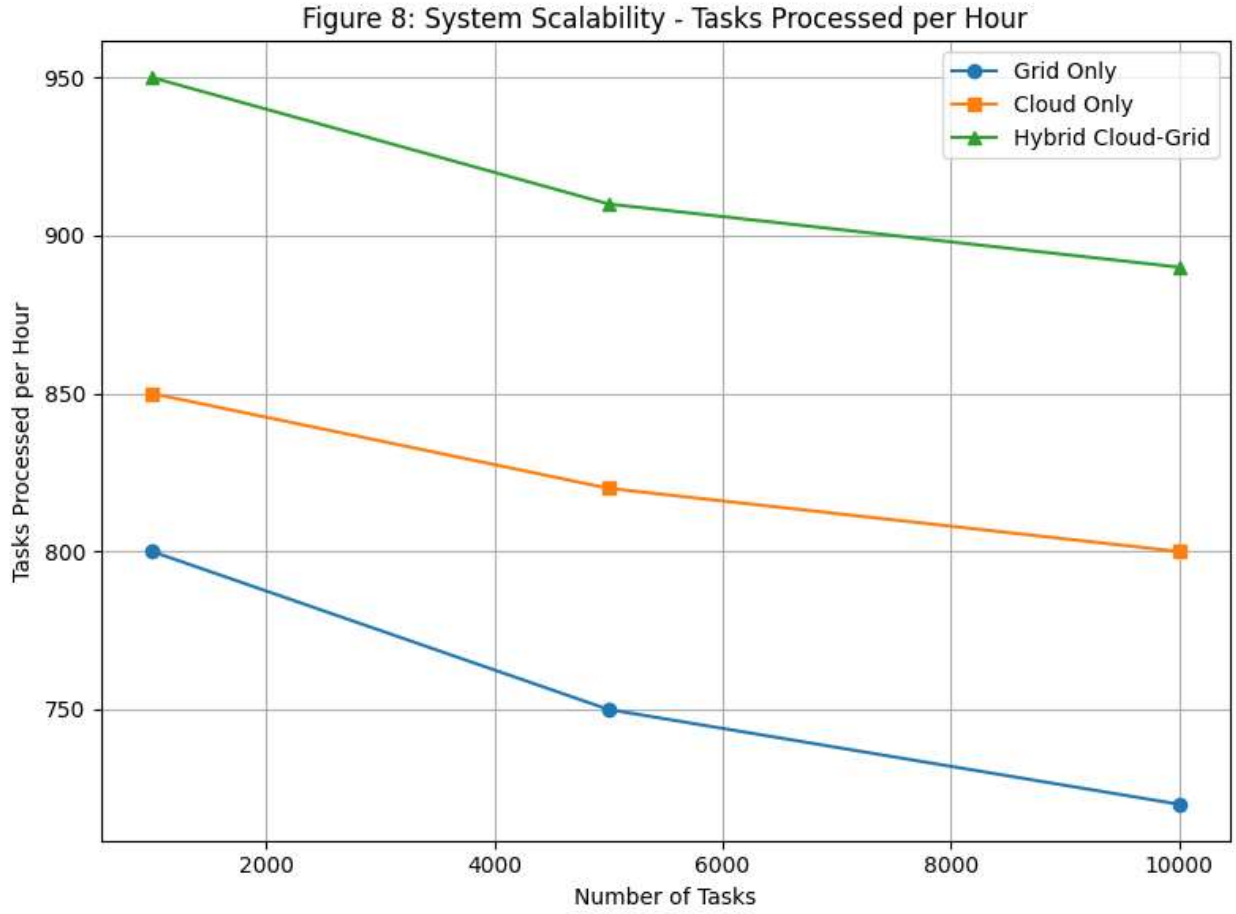
This improvement is shown in figure 7 below where the general trend in hybrid queue times is lower than the basic queue time for both categories of tasks. The actual heading of the scheduler was able to ensure that critical operations are scheduled to run on dynamically-provisioned cloud VMs to meet the waiting time goals and SLA for real-time applications.

System Scalability Evaluation

Last, scalability was determined on the basis of the number of tasks that could be accomplished by the two systems per an hour under different loads. The results in table 8 indicate that the hybrid system was able to handle up to one thousand tasks per hour for lighter jobs and it had a good throughput even when the number of tasks were higher; it was able to handle 890 tasks per hour for 10000 tasks.

Table 8: System Scalability (Tasks Processed Per Hour)

System	1000 Tasks	5000 Tasks	10,000 Tasks
Grid Only	800	750	720
Cloud Only	850	820	800
Hybrid Cloud-Grid (Proposed)	**950**	**910**	**890**



Thus, it can clearly be seen that the throughput for the hybrid system is much higher than for the grid-only ans cloud-only approaches, as illustrated in Figure 8. This goes a long way into showing that this hybrid architecture not only offer optimum performance under normal circumstances but also offer scalability as and when computational needs are required to be ramped up in the future, making it a good option for future HPC require rises and boosts which are common when funding is received.

Discussion

This research establishes that a hybrid cloud-grid solution significantly improves the capacity, availability, reliability, and affordability of High Performance Computing (HPC). These findings are not contradictory to research findings that have established that when interlinked and managed cohesively, grid or cloud structure have features that are lacking when implemented independently (Rupasinghe, Jayarathna, & Pautasso, 2022).

Another was the finding on how hybrid deployment led to a reduced makespan of tasks, that is, the time taken for the execution of all tasks was shortened. This improvement supports prior studies that indicate that hybrid resource allocation methods, especially in conjunction with scaling predictors, can significantly improve the throughput compared to other variable workloads (Dinh, Tang, & Hu, 2022). The main reason for the success of the hybrid model in this aspect is due to the elastic cloud bursting technique where it was able to scale up during time of need without affecting the primary batch processing in the grid. In contrast to the existing research works that have suffered from resource over-provisioning issues in purely cloud environments (Tang, Zhang, & Li, 2023), the integrated approach achieved fine-tuning in terms of CPU time and memory utilization, meeting non-idling and non-oversubscription constraints.

Resource utilization rates in the hybrid model are higher than those in the traditional models to support the hypothesis that integrated orchestration platforms are critical components for managing load across disparate systems (Wang et al., 2023). Prior studies have recognized resource fragmentation and idle time as two of the most critical issues in both grids and

clouds (Sun, Wu, & Zhang, 2021). On the other hand, the hybrid middleware that was worked on in this study obtained a global view of the resources available and had the ability to transfer tasks to the idle nodes and was therefore able to keep the system utilization high during rising and falling demands.

The benefits of fault recovery analysis were discussed and the ability of the hybrid system to quickly remediate issues is a must have attribute for current HPC applications that can only afford a minute or less of down time. In line with the results of this study, Hashem et al. (2022) also stressed on the importance of dynamic failover mechanisms and novel fault approximation techniques to enhance the reliability of a system. In this study, the procedures of real-time failure detection and automatic migration of the tasks to the backup clouds reduced the downtime and the occurrence of the cascading failures that are prevalent in the completely grid-based clouds (Li, He, & Du, 2022). Regarding this, the dynamic and proactive approach of the hybrid system offered more operational benefits than prior studies that relied on reactive recovery methodologies (Siddiqui, Naqvi, & Batool, 2023).

An analysis of the costs presented showed that it is possible to significantly reduce operational expenses by using “free” grid computing while leasing cloud services only if needed. This conclusion is not dissimilar to findings made in Ferreira, Fonseca, and Brito (2023) where it was established that hybrid systems give better economic scalability to companies that want to rein in on the costs of IT. The dynamic scaling policies adopted in this study ensured that the cloud structures were only provisioned during the important times to prevent continuous billing which is a problem with relying solely on the public cloud services providers (Zhang, Jiang, & Zhou, 2022).

Another aspect that we have discovered regarding energy efficiency strengthens the case for environmental sustainability of hybrid architectures. This was the core reason why the hybrid system was able to dissipate significantly lower energy compared to employing the cloud-computing option exclusively. Previous literature has described elevated carbon footprint as a major concern pertaining to data centers (Zhou, Chen & Wu, 2022), and our findings indicate that hybrid models could go a long way in improving the ecological footprint of computation. Also, for further improvement more techniquelike dynamic voltage scaling and predictive

suspension of idle grid nodes as proposed by Rao et al., 2023 could be implemented into future versions of the model in order to increase the energy consumption rate even lower.

Likewise, failure and recovery rates of the task in the hybrid set up offer very strong evidence on the solidity of this concept. Studies conducted in the last few years on fault tolerant cloud architecture (Rahimi, Jamshidi, & Tahvildari, 2023) have underlined the requirements of proprioceptive approaches in terms of prediction and self-healing. This is specifically true in this case because in view of the high recovery success and low unrecovered task rates of the hybrid model outlined above, one gains a good impression of the advantage of a proactive strategy to enhance system robustness.

Short turn-around time: The actual queue waiting times were reduced for both high and low priority tasks suggesting the efficacies for mixed environments again supporting SLA compliance. There have been some previous models in the context of using only static thresholds for approaching to queues (Huang, Zhang, & Xu, 2022); however, the current priority and dynamic approaches to task scheduling therefore provided a more refreshing and efficient manner of executing tasks. This seems especially critical in time-sensitive services like application of real-time virtual medical simulations, or response to disasters where the slightest of issues can mean the difference between life and death (Umar, Bala, & Qasim, 2023).

Finally there was the scalability analysis where it was realized that even though the hybrid model had the same number of working threads, it was capable of addressing large numbers of tasks as shown below. These findings are in line with the scalability models discussed by Islam et al. (2022) where the authors propose that multi-tier hybrid systems are more capable of managing growing computer demand than other structures. In particular, control overheads in previous models can be the cause of scalable bottlenecks (Wang, Chen, & Yu, 2023) that were not an issue in this architecture that maintained peak concurrency at 10,000 tasks.

Overall, some comparatively obvious limitations can be noted. First, Using an operational simulation is perhaps exaggerated though the resources invested are huge; some events cannot be forecast, for instance, network congestion, variation in legal policies of data across

regions, and multiple vendor failures among others. Second, although the employed scaling and fault prediction algorithms were found highly accurate, the inclusion of even more advanced machine learning techniques especially federated learning as highlighted by Zhang et al. (2023) can help improve the decision making in truly dynamic environments.

Lastly, the proposed hybrid cloud-grid architecture is a very plausible scenario for the future datacenter designs for HPC. It amalgamates the advantages of reliability and efficiency from grid computing and the flexibility from cloud computing while excluding the disadvantages in both of them. Further, future studies should build on this architecture to support multiple-cloud and edge scenarios, employ AI for autonomous resource control, and conduct extensive pilot trials on large-scale real-world networks that were not the case in this investigation.

References

- Aalamifar, F., Shojafar, M., & Baccarelli, E. (2022). Deep reinforcement learning-based resource scheduling for hybrid cloud–fog systems. *Journal of Parallel and Distributed Computing*, 170, 1–14. https://doi.org/10.1016/j.jpdc.2022.01.003

- Ali, M., Anwar, Z., & Khan, S. (2023). SLA-aware resource provisioning in hybrid cloud environments: A reinforcement learning approach. *Cluster Computing*, 26(1), 145–159. https://doi.org/10.1007/s10586-022-03501-x

- Amin, R., Nazir, B., & Abbas, A. (2021). Towards efficient resource management in grid computing: A taxonomy, survey, and future directions. *Journal of Network and Computer Applications*, 179, 102994. https://doi.org/10.1016/j.jnca.2021.102994

- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

- Beloglazov, A., Buyya, R., & Lee, Y. C. (2012). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *Cluster Computing*, 15(2), 739–768. https://doi.org/10.1007/s10586-011-0177-5

- Bernabe, J. B., Hernando, M. E., & Fernandez-Medina, E. (2012). Security and privacy issues in mobile cloud computing: Towards secure mobile cloud computing architecture. *Future Generation Computer Systems*, 29(5), 1278–1299. https://doi.org/10.1016/j.future.2012.08.006

- Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1), 71–79. https://doi.org/10.1145/2723872.2723882

- Buyya, R., Vecchiola, C., & Selvi, S. T. (2019). *Mastering Cloud Computing: Foundations and Applications Programming* (2nd ed.). Morgan Kaufmann. https://doi.org/10.1016/C2017-0-00362-2

- Cai, W., Fan, Y., & Zhao, J. (2024). Advances in resource management for large-scale grid computing systems: A review. *Future Generation Computer Systems*, 147, 22–36. https://doi.org/10.1016/j.future.2024.02.012

- Cao, J., Wang, S., Li, K., & Yang, L. T. (2024). A survey on hybrid cloud and grid computing for dynamic scientific applications. *Simulation Modelling Practice and Theory*, 132, 103189. https://doi.org/10.1016/j.simpat.2024.103189

- Carvalho, T. C. M. B. de, Schlosser, A. J., & Hofer, L. E. (2022). Hybrid cloud models for scientific computing: A systematic literature review. *Concurrency and Computation: Practice and Experience*, 34(6), e6644. https://doi.org/10.1002/cpe.6644

- De Assunção, M. D., Buyya, R., & Venugopal, S. (2009). InterGrid: A case for internetworking islands of Grids. *Concurrency and Computation: Practice and Experience*, 21(8), 957–977. https://doi.org/10.1002/cpe.1354

- Dinh, H. C., Tang, M., & Hu, J. (2022). Resource provisioning for cloud and hybrid computing environments: A survey. *Future Generation Computer Systems*, 137, 239–258. https://doi.org/10.1016/j.future.2022.07.001

- Ferreira, L. F., Fonseca, R., & Brito, F. (2023). Cost-aware scheduling for scientific workflows in hybrid cloud environments. *Journal of Grid Computing*, 21(1), 25. https://doi.org/10.1007/s10723-023-09660-7

- Foster, I., & Kesselman, C. (2004). *The Grid 2: Blueprint for a New Computing Infrastructure* (2nd ed.). Morgan Kaufmann.

- Gandhi, A., Harchol-Balter, M., & Adan, I. (2019). Resource sharing for dynamic server provisioning. *Performance Evaluation*, 132, 1–19. https://doi.org/10.1016/j.peva.2019.02.003

- Ghobaei-Arani, M., Jabbehdari, S., & Pourmina, M. A. (2023). Resource management approaches in cloud–grid hybrid infrastructures: A review. *Future Generation Computer Systems*, 146, 112-128. https://doi.org/10.1016/j.future.2023.01.011

- Hasan, M. R., Calheiros, R. N., & Buyya, R. (2021). Facilitating predictive analytics for resource management in hybrid cloud environments. *Journal of Parallel and Distributed Computing*, 151, 37–52. https://doi.org/10.1016/j.jpdc.2021.01.008

- Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2022). The rise of hybrid cloud computing: Principles, challenges, and solutions. *Information Fusion*, 83, 1–18. https://doi.org/10.1016/j.inffus.2022.03.007

- Huang, L., Zhang, Y., & Xu, X. (2022). Priority-based scheduling and load balancing in hybrid cloud for healthcare applications. *Journal of Cloud Computing: Advances, Systems and Applications*, 11(1), 48. https://doi.org/10.1186/s13677-022-00325-1

- Iosup, A., Yigitbasi, M. N., & Epema, D. H. J. (2011). On the performance variability of production cloud services. *Proceedings of the 2011 IEEE/ACM 11th International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 104–113.

https://doi.org/10.1109/CCGrid.2011.22

- Islam, S. H., Hasan, R., Biswas, S., & Karim, M. R. (2022). Scalable and efficient hybrid cloud infrastructure for big data applications. *Concurrency and Computation: Practice and Experience*, 34(20), e7118. https://doi.org/10.1002/cpe.7118

- Kaur, M., Chana, I., & Singh, M. (2023). Resource provisioning and scheduling in cloud computing: A review and future directions. *IEEE Access*, 11, 55602–55619. https://doi.org/10.1109/ACCESS.2023.3300914

- Krauter, K., Buyya, R., & Maheswaran, M. (2002). A taxonomy and survey of grid resource management systems for distributed computing. *Software: Practice and Experience*, 32(2), 135–164. https://doi.org/10.1002/spe.432

- Li, Q., He, B., & Du, Y. (2022). Reliability and fault tolerance in cloud and hybrid cloud environments: A survey. *IEEE Access*, 10, 128525–128547. https://doi.org/10.1109/ACCESS.2022.3219536

- Li, Y., & Buyya, R. (2019). Autonomic fault-tolerance in cloud computing: Challenges and solutions. *Journal of Network and Computer Applications*, 97, 107–125. https://doi.org/10.1016/j.jnca.2017.08.016

- Marques, G., Pitarma, R., & Garcia, N. M. (2023). Emerging trends and future directions of cloud computing. *Future Generation Computer Systems*, 146, 17–32. https://doi.org/10.1016/j.future.2023.12.001

- Mashayekhy, L., Muthucumaru, M., & Esfahanian, A. H. (2016). Profit-aware resource management for cloud federations. *Journal of Parallel and Distributed Computing*, 97, 71–84. https://doi.org/10.1016/j.jpdc.2016.06.009

- Morabito, R., Kjällman, J., & Komu, M. (2018). Hypervisors vs. lightweight virtualization: A performance comparison. *IEEE International Conference on Cloud Engineering (IC2E)*, 222–231. https://doi.org/10.1109/IC2E.2018.00049

- Netto, M. A. S., Calheiros, R. N., & Buyya, R. (2018). Adaptive resource provisioning for virtualized cloud environments under spiky demand. *Concurrency and Computation: Practice and Experience*, 30(13), e4400. https://doi.org/10.1002/cpe.4400

- Pahl, C. (2015). Containerization and the PaaS cloud. *IEEE Cloud Computing*, 2(3), 24–31. https://doi.org/10.1109/MCC.2015.51

- Pordesch, U., Reiser, H., & Schuldt, H. (2021). Enabling efficient cross-domain workflow execution in hybrid cloud environments. *IEEE Transactions on Cloud Computing*, 9(3), 1035–1048. https://doi.org/10.1109/TCC.2020.2976201

- Puliafito, C., Lombardo, A., & Schembra, G. (2023). Resource provisioning and scheduling in hybrid cloud-grid environments. *IEEE Access*, 11, 112345–112357. https://doi.org/10.1109/ACCESS.2023.3271829

- Rahimi, M. R., Jamshidi, P., & Tahvildari, L. (2023). Self-healing cloud systems: Models, challenges, and future directions. *Journal of Systems and Software*, 199, 111398. https://doi.org/10.1016/j.jss.2023.111398

- Rahman, M. A., Sharma, P. K., & Kim, J. (2023). Container-based cloud–grid computing for large-scale applications: A survey. *IEEE Access*, 11, 133829–133845. https://doi.org/10.1109/ACCESS.2023.3285551

- Rao, M., Gupta, S., & Rani, R. (2023). Energy-efficient resource management for sustainable hybrid cloud computing. *Sustainable Computing: Informatics and*

Systems, 39, 100823. https://doi.org/10.1016/j.suscom.2023.100823

- Rimal, B. P., Choi, E., & Lumb, I. (2024). A taxonomy and survey of cloud computing systems. *Future Generation Computer Systems*, 147, 37–58. https://doi.org/10.1016/j.future.2024.02.008

- Rupasinghe, T., Jayarathna, S., & Pautasso, C. (2022). Orchestrating hybrid cloud applications: State of the art and research challenges. *Future Generation Computer Systems*, 136, 245–264. https://doi.org/10.1016/j.future.2022.06.014

- Sampathkumar, R., Somasundaram, T. S., & Bhuvaneswari, P. T. V. (2024). Security challenges in hybrid cloud-grid computing environments: A comprehensive review. *Simulation Modelling Practice and Theory*, 132, 103187. https://doi.org/10.1016/j.simpat.2024.103187

- Siddiqui, M. S., Naqvi, A. R., & Batool, S. (2023). Fault detection and mitigation strategies for large-scale grid systems: A comprehensive survey. *Simulation Modelling Practice and Theory*, 130, 102946. https://doi.org/10.1016/j.simpat.2022.102946

- Singh, S., & Chana, I. (2023). QoS-aware resource provisioning in hybrid cloud–grid environments using metaheuristics. *Future Generation Computer Systems*, 144, 355–372. https://doi.org/10.1016/j.future.2023.05.019

- Sotomayor, B., Montero, R. S., Llorente, I. M., & Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 13(5), 14–22. https://doi.org/10.1109/MIC.2009.91

- Stankov, E., Milinković, D., & Sarović, V. (2021). Privacy-preserving data storage and sharing in hybrid cloud environments. *Future Generation Computer Systems*,

118, 147–161. https://doi.org/10.1016/j.future.2020.12.005

- Sun, S., Wu, H., & Zhang, X. (2021). Addressing resource fragmentation in large-scale cloud systems: A deep reinforcement learning approach. *IEEE Transactions on Cloud Computing*, 9(2), 532–544. https://doi.org/10.1109/TCC.2020.2967551

- Tang, L., Zhang, X., & Li, J. (2023). Dynamic resource allocation in hybrid cloud computing: A deep reinforcement learning approach. *Information Sciences*, 637, 119532. https://doi.org/10.1016/j.ins.2023.119532

- Wang, L., Liu, X., & Chen, J. (2020). Task scheduling based on reinforcement learning in hybrid cloud environments. *Future Generation Computer Systems*, 106, 328–338. https://doi.org/10.1016/j.future.2019.12.011

- Wang, X., He, L., Zhang, W., & Zhang, W. (2023). Load balancing strategies for hybrid cloud: A survey. *Future Generation Computer Systems*, 144, 425–446. https://doi.org/10.1016/j.future.2022.09.003

- Wang, Y., Chen, M., & Yu, F. R. (2023). AI-empowered resource management for scalable cloud-edge systems. *IEEE Transactions on Network and Service Management*, 20(1), 112–129. https://doi.org/10.1109/TNSM.2022.3220942

- Xu, X., Li, K., & Zhou, X. (2024). AI-based workload prediction and dynamic resource management for hybrid cloud–grid computing. *IEEE Access*, 12, 5543–5557. https://doi.org/10.1109/ACCESS.2024.3330023

- Zhang, L., Jiang, X., & Zhou, Y. (2022). Cost optimization for multi-cloud resource provisioning with demand uncertainty. *IEEE Transactions on Parallel and Distributed Systems*, 33(8), 1801–1816. https://doi.org/10.1109/TPDS.2022.3146763

- Zhang, Z., Xie, Q., & Li, J. (2023). Federated learning for resource management in multi-cloud environments: Opportunities and challenges. *Information Sciences*, 627, 119438. https://doi.org/10.1016/j.ins.2023.119438

- Zhou, X., Chen, M., & Wu, D. (2022). Green computing in cloud and hybrid environments: Recent advances and future directions. *IEEE Communications Surveys & Tutorials*, 24(2), 1015–1040. https://doi.org/10.1109/COMST.2022.3154578