**Energy Consumption**

Based on the provided utilization levels, you can categorize the virtual machines’ (VMs) CPU utilization into three different scenarios: high, medium, and low utilization. Here is a breakdown of these scenarios:

**High Utilization:**

**CPU Utilization Levels: 80%, 90%, and 100%**

This scenario represents the VMs operating under heavy load conditions, where they are using a substantial portion of their available CPU resources.

**Medium Utilization:**

**CPU Utilization Levels: 60%, 70%, and 80%**

In this scenario, the VMs are moderately loaded, utilizing a fair amount of their CPU resources but not to their maximum capacity.

**Low Utilization:**

**CPU Utilization Levels: 30%, 40%, and 50%**

This scenario depicts the VMs under light load conditions, using only a small fraction of their available CPU resources.

Utilization Category CPU Utilization Levels (%)

High Utilization 80, 90, 100

Medium Utilization 60, 70, 80

Low Utilization 30, 40, 50

These utilization levels can be used in the simulation to assess how different workloads impact the performance of the data center. By running simulations under these three scenarios, you can observe how the system behaves under various load conditions, which can help in understanding the performance, resource allocation, and potential bottlenecks within the data center.

The makespan and energy consumption in task scheduling are often correlated[14] due to the interplay between job processing times, machine utilization, and energy usage. Here’s how the makespan and energy consumption are correlated:

**Machine Utilization**:

* High machine utilization, where machines are continuously busy processing jobs, can lead to a shorter makespan but higher energy consumption. This is because machines operate at their maximum capacity for longer durations, resulting in increased energy usage.

**Job Processing Times**:

* Longer job processing times typically result in a longer makespan as more time is required to complete all jobs. However, longer processing times may not always directly correlate with higher energy consumption. It depends on factors such as machine speed and efficiency.

**Idle Time**:

* Idle time, where machines are not actively processing jobs, contributes to higher energy consumption without reducing the makespan. Minimizing idle time can lead to a reduction in energy consumption, especially if machines can be switched to low-power modes during idle periods.

**Energy-Efficient Scheduling**:

* Optimizing task scheduling to minimize energy consumption while maintaining a reasonable makespan involves finding a balance between job sequencing, machine allocation, and energy-aware scheduling policies. Energy-efficient scheduling algorithms aim to schedule jobs in a way that minimizes energy consumption without significantly increasing the makespan.

**Trade-off**:

* There is often a trade-off between minimizing the makespan and minimizing energy consumption [13]. Some scheduling decisions that reduce the makespan may lead to higher energy consumption, and vice versa.
* Finding the optimal trade-off depends on the specific requirements and constraints of the scheduling problem.

Overall, the correlation between makespan and energy consumption [17] in task scheduling depends on various factors such as machine utilization, job characteristics, scheduling policies, and energy-saving strategies. Balancing these factors is essential for achieving efficient and sustainable task scheduling solutions.

Minimize both the makespan (Cmax) and the total energy consumption (TEC), computed as follows:

* Total Energy Consumption (TEC) = PEC + IEC

Where:

* Processing Energy Consumption (PEC) = Σ(Pj \* Ti) / 1000
* Idle Energy Consumption (IEC) = 0% utilization consumption as per server

The power consumption of a data center varies depending on its utilization level[16]. Here’s a general overview of power consumption estimates for data centers at different utilization levels:

**High Utilization:**

At high utilization levels, when the data center’s servers and infrastructure[15] are running close to their maximum capacity, power consumption is typically at its peak.

The power consumption in a data center at high utilization is primarily driven by the energy consumed by servers, cooling systems, networking equipment, and other supporting infrastructure.

Cooling systems, in particular, may require more energy to maintain optimal operating temperatures when servers are running at full capacity.

Power Usage Effectiveness (PUE), which measures the ratio of total power consumed by the data center to the power consumed by IT equipment, tends to be lower at high utilization levels due to more efficient use of resources.

**Low Utilization:**

At low utilization levels, when the data center is operating well below its maximum capacity, power consumption is relatively lower compared to high utilization scenarios.

However, even at low utilization, data centers typically consume a significant amount of power due to the overhead associated with maintaining infrastructure readiness and availability.

Cooling systems may still require substantial energy to maintain optimal environmental conditions within the data center facility, even when server loads are minimal.

PUE may be higher at low utilization levels due to the relatively higher proportion of energy consumed by supporting infrastructure compared to IT equipment.

**Medium Utilization:**

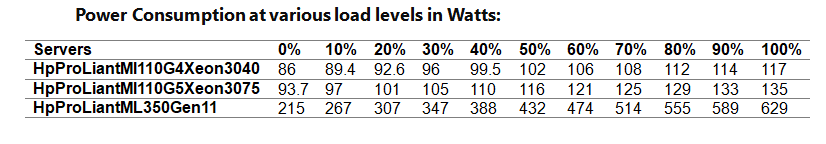
At medium utilization levels, power consumption falls between the extremes of high and low utilization.

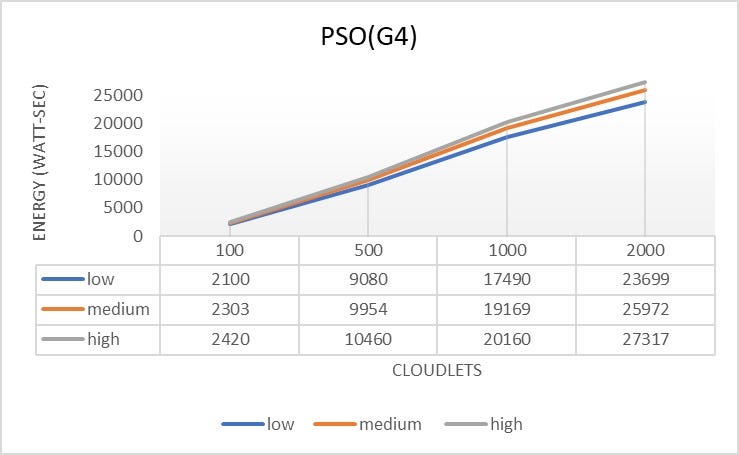
Power consumption in a data center at medium utilization is influenced by a combination of factors, including the number of active servers, workload distribution, and efficiency of cooling and power distribution systems.

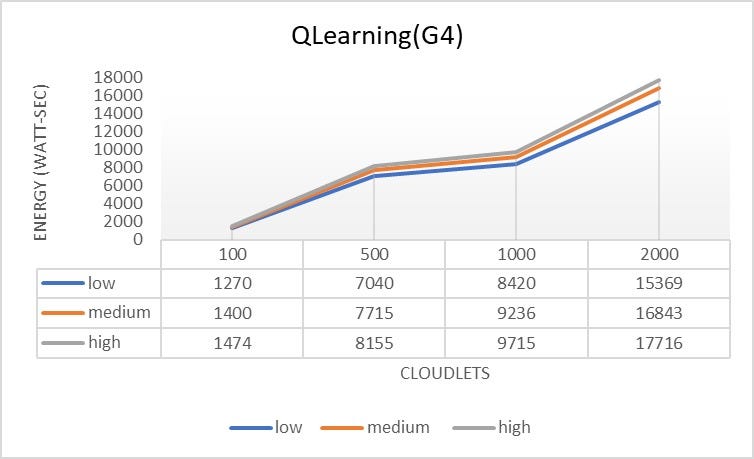
The efficiency of the data center’s infrastructure and operational practices can have a significant impact on power consumption at medium utilization levels.

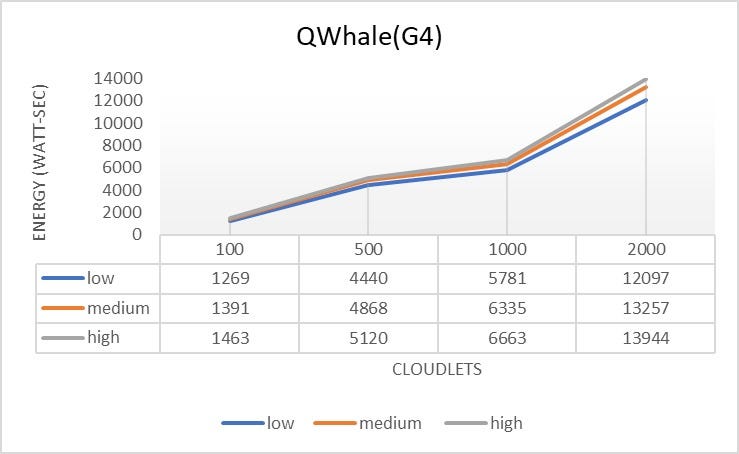
PUE values at medium utilization may vary depending on the effectiveness of energy management practices and resource allocation strategies.

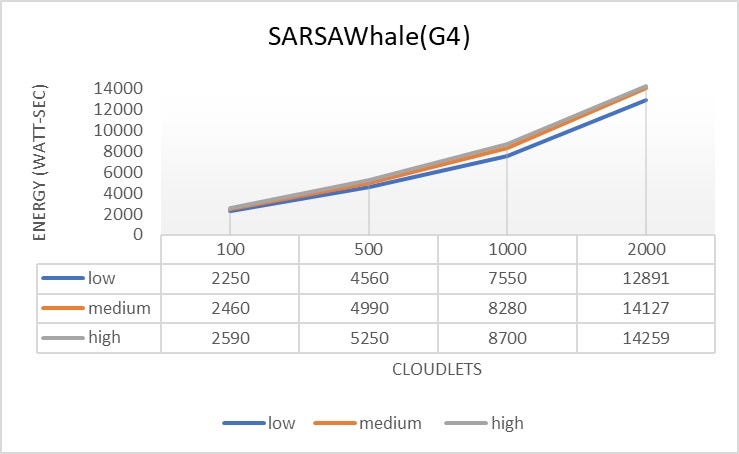
To estimate the power consumption of used servers like the HP ProLiant G4, G5, and ML350 Gen11, we can provide some general guidelines based on their specifications. However, it’s important to note that actual power consumption can vary based on factors such as server configuration, workload, and environmental conditions. Here’s a rough estimation of power consumption for each server model:

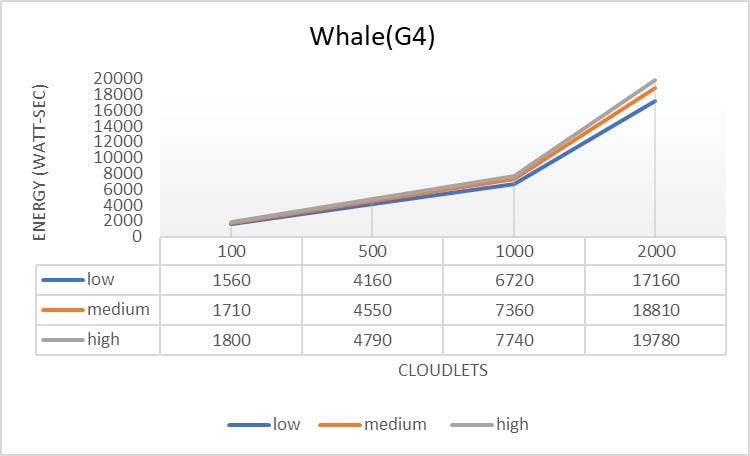




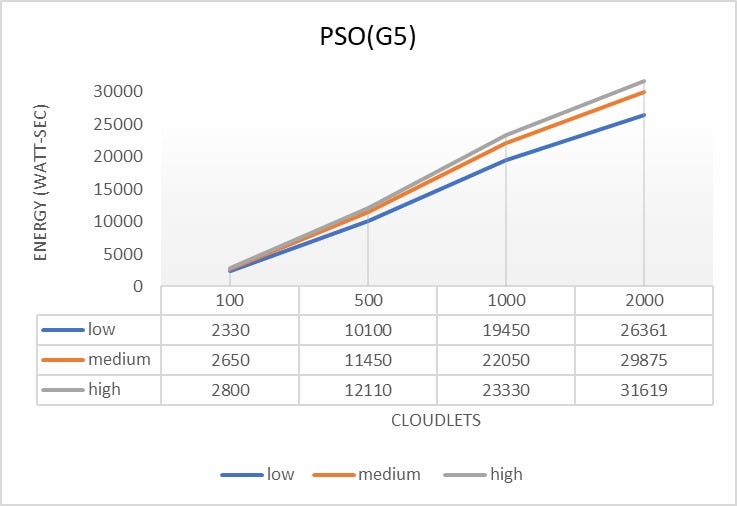


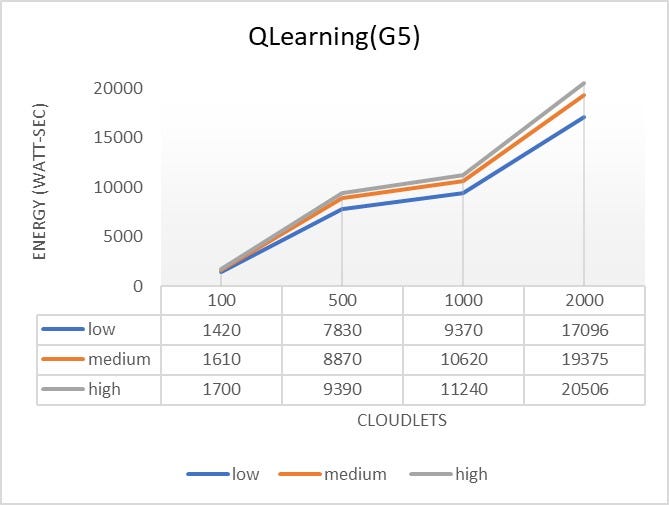


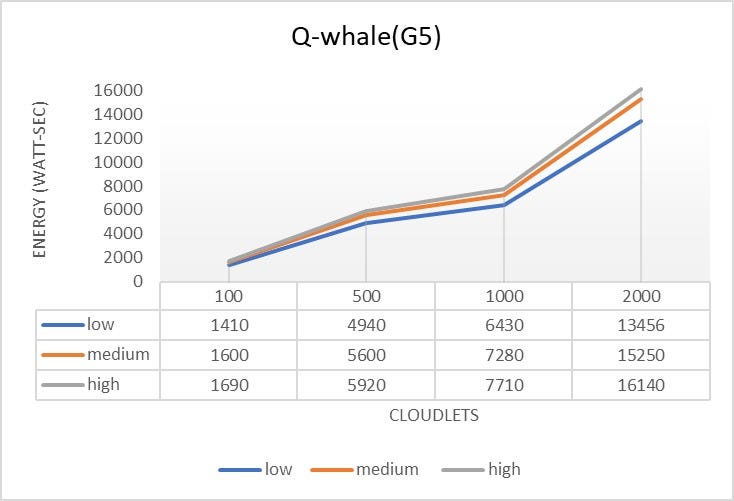


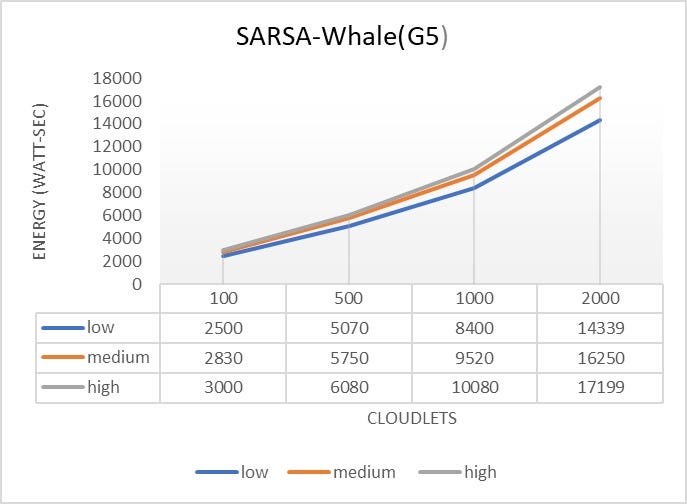


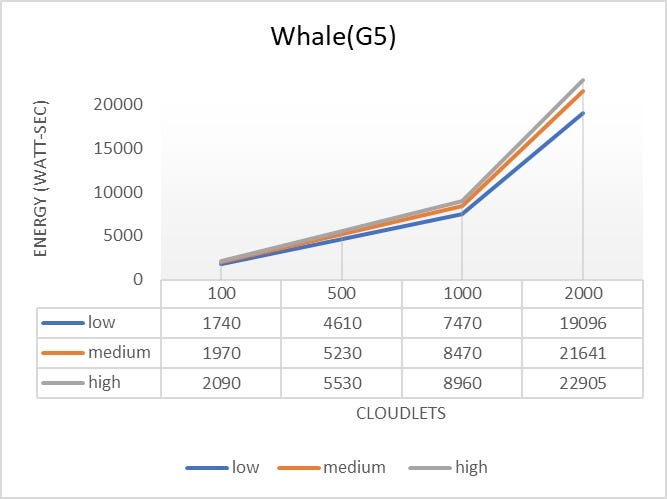
HP Proliant G4



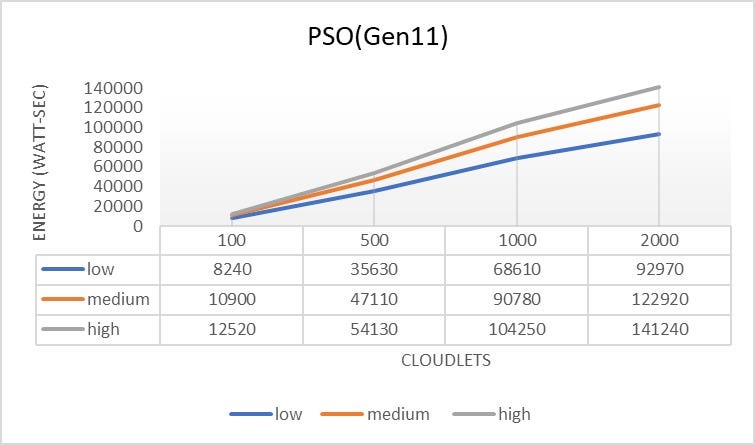


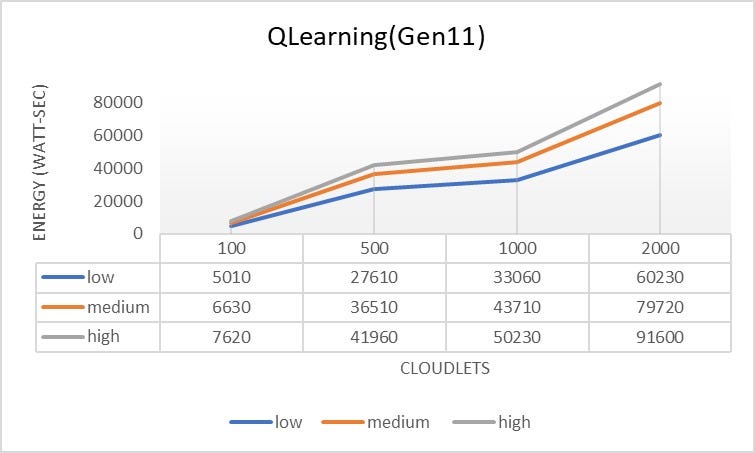


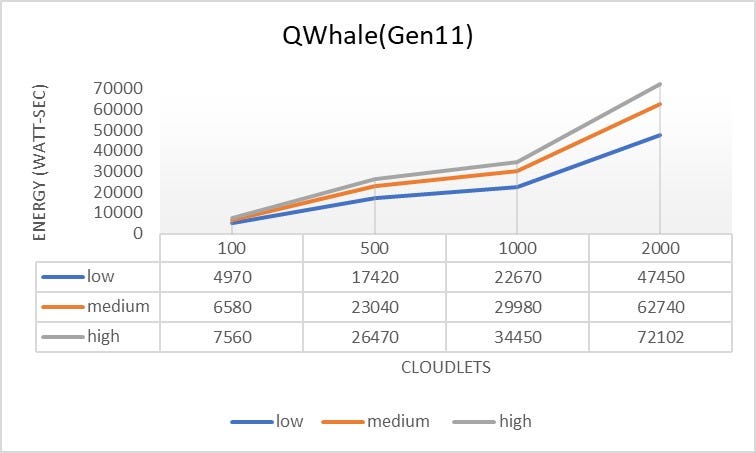


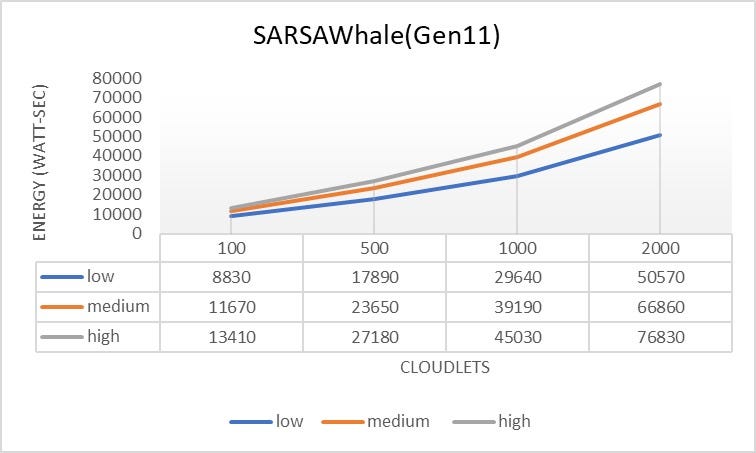


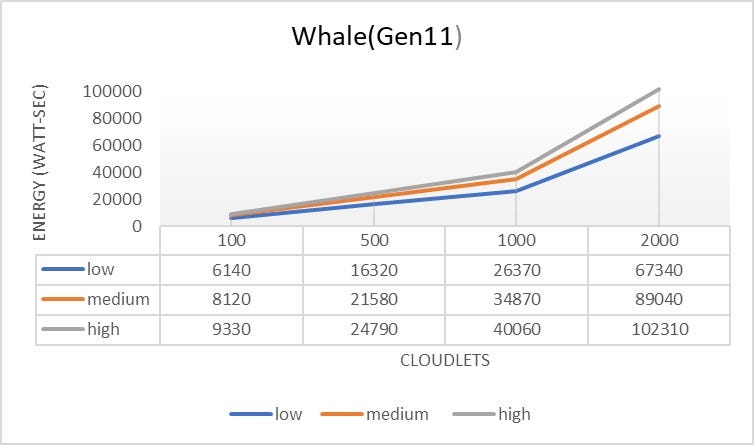
HP Proliant G5



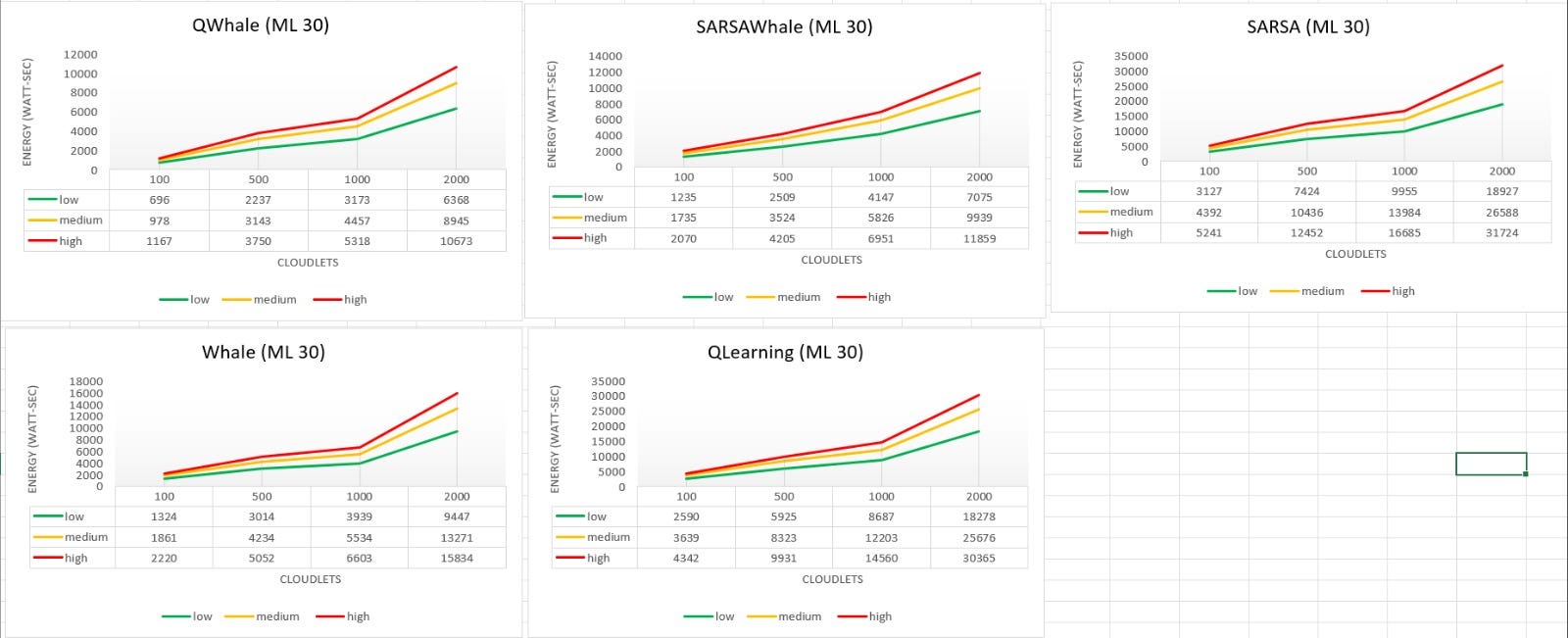


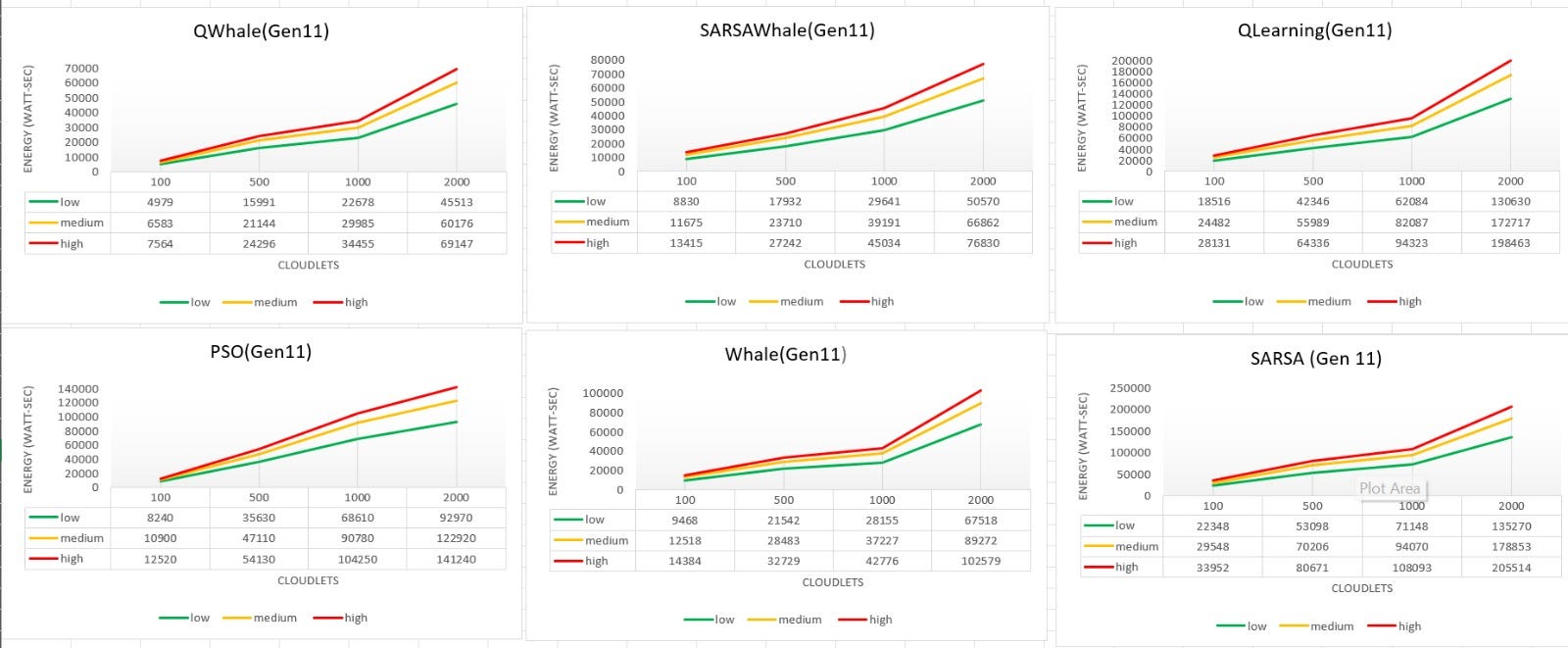






HP ProliantML350

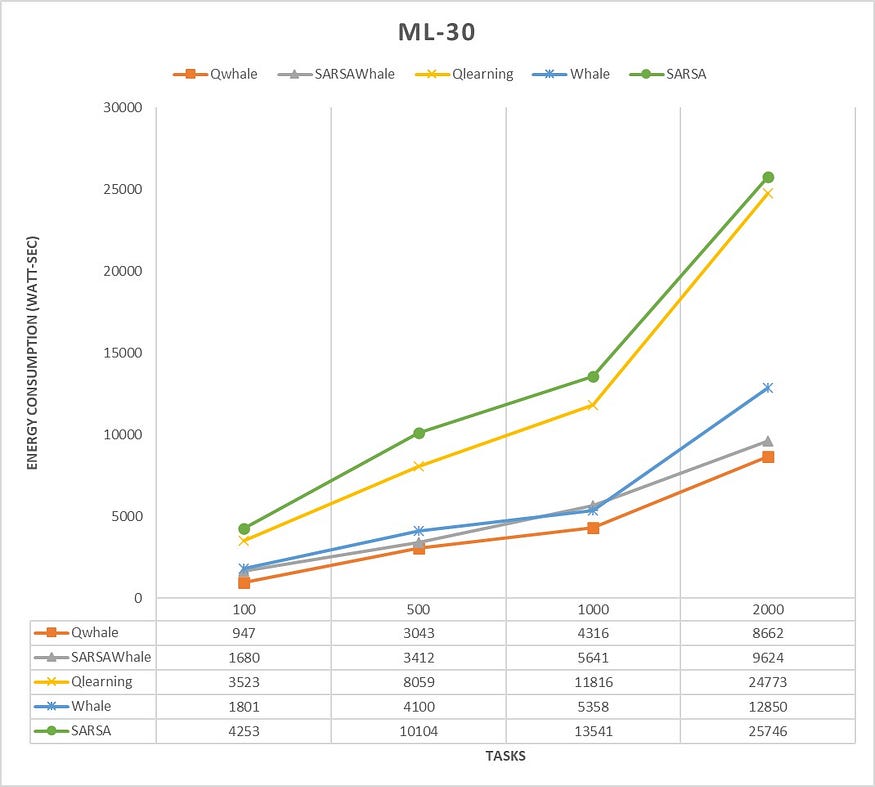


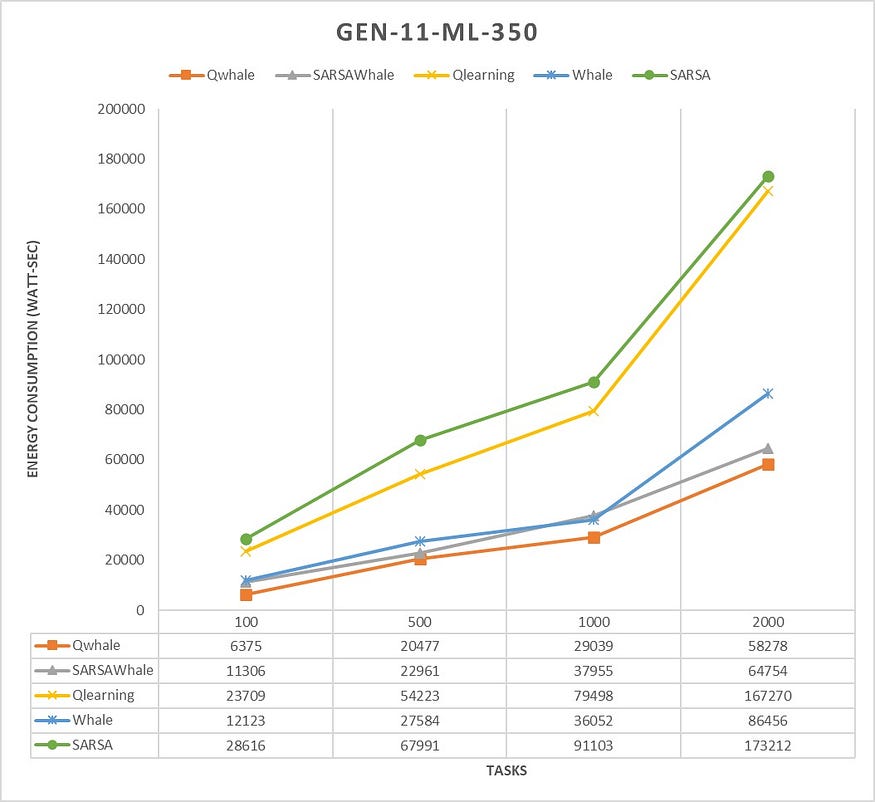


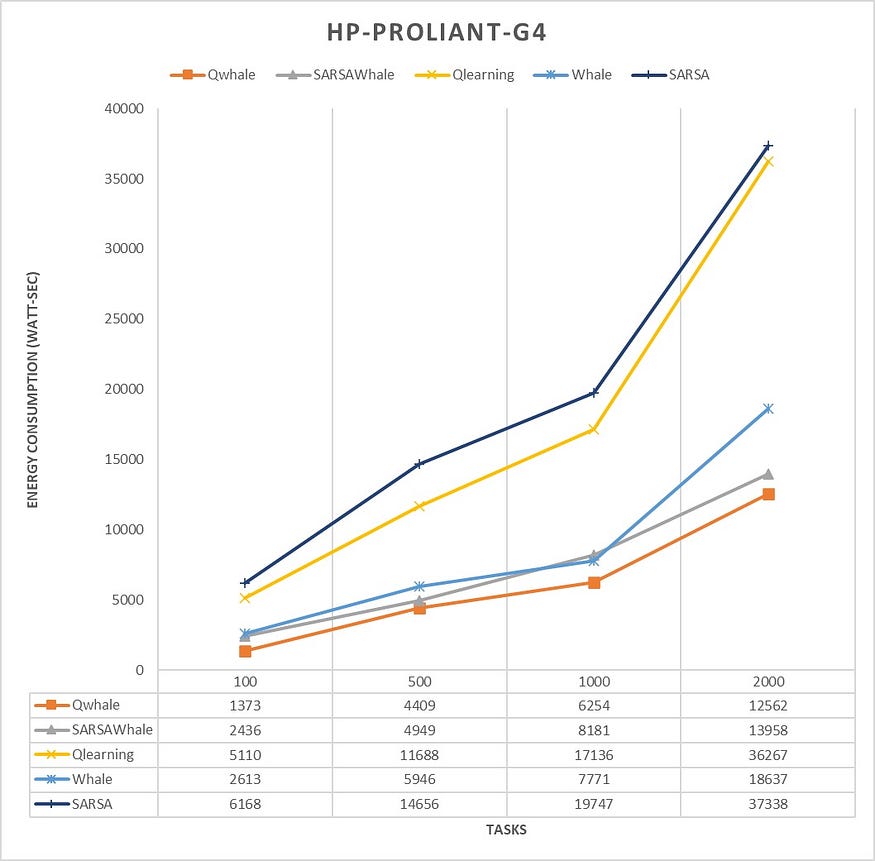


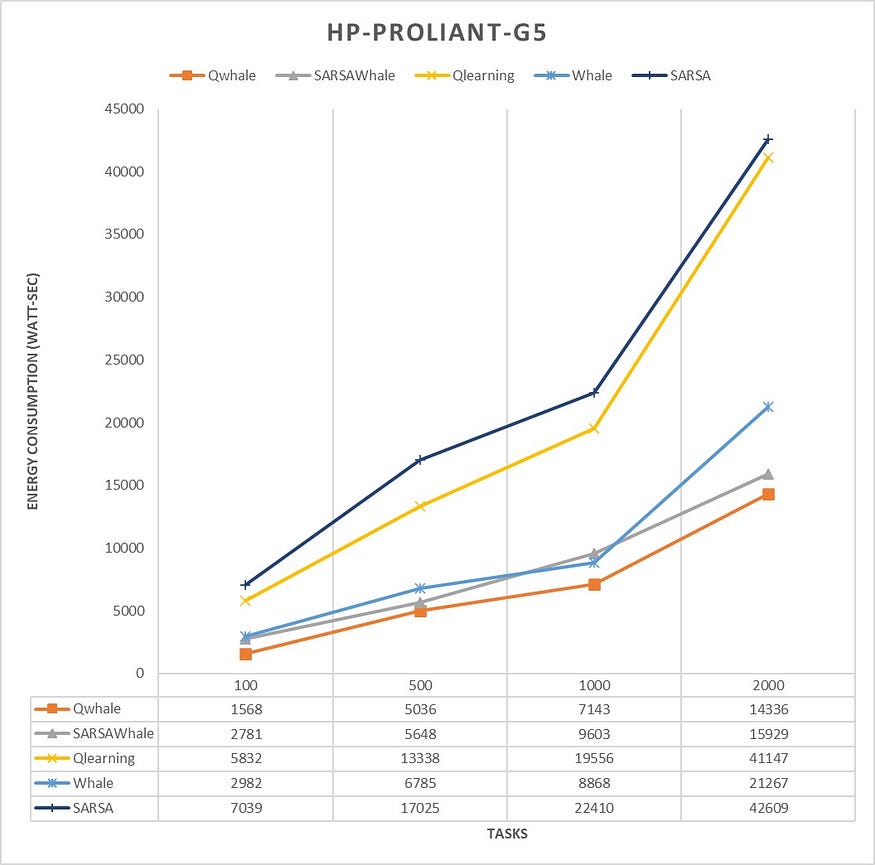
A screenshot of a graph

Description automatically generated







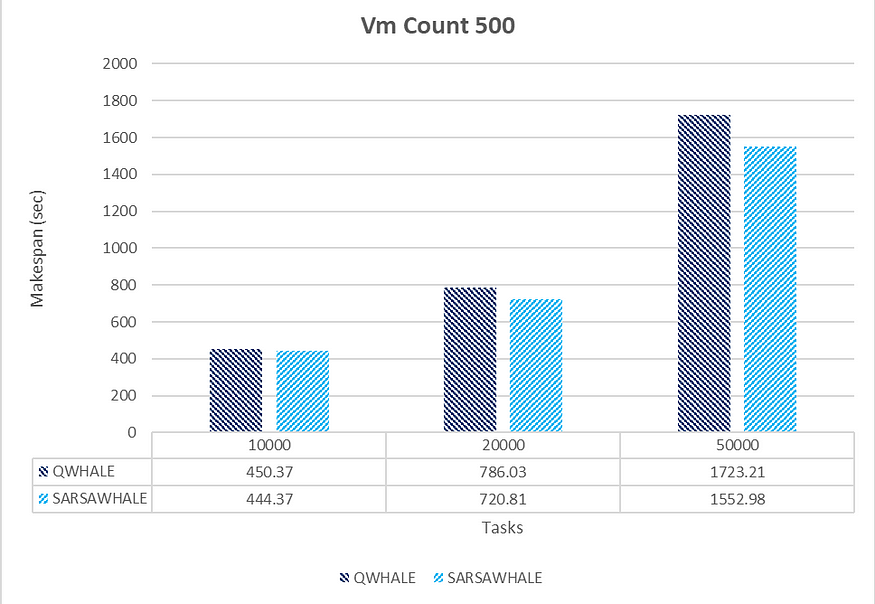


* The Qwhale algorithm consistently demonstrates the lowest energy consumption across all task counts.
* For 100 tasks, Qwhale consumes 1568 watt-seconds, which is significantly lower compared to the next best SARSawhale at 2781 watt-seconds and substantially lower than Q-learning and SARSA, which consume 5832 and 7039 watt-seconds, respectively.
* As the number of tasks increases to 500, 1000, and 2000, Qwhale maintains its efficiency with energy consumption values of 5036, 7143, and 14336 watt-seconds, respectively. This trend highlights the scalability and efficiency of Qwhale in handling larger workloads.

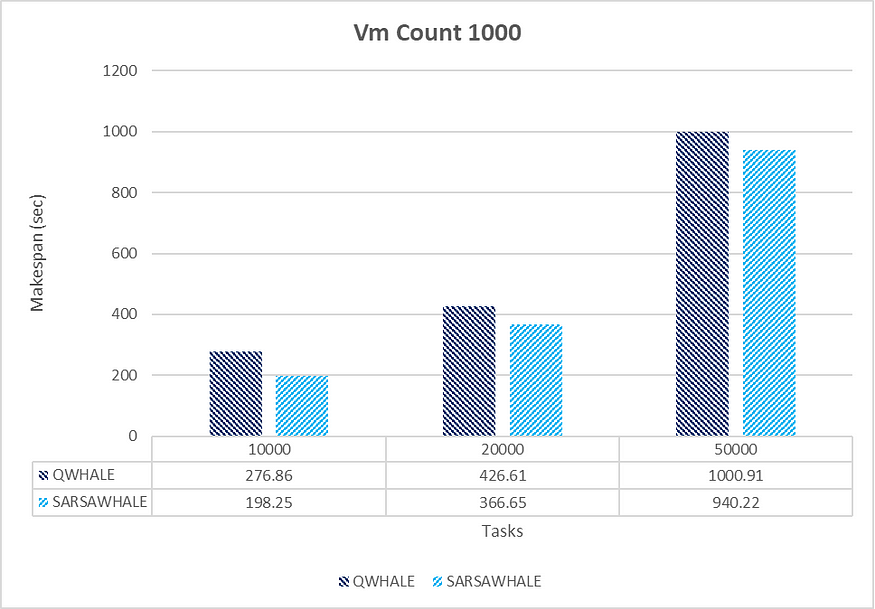
**Comparison with Other Algorithms**:

* SARSawhale shows the second-best performance but still consumes more energy than Qwhale, particularly noticeable as the task count increases (5648 watt-seconds for 500 tasks, 9603 for 1000 tasks, and 15929 for 2000 tasks).
* Traditional Q-learning and SARSA algorithms exhibit much higher energy consumption, with Q-learning peaking at 41147 watt-seconds and SARSA at 42609 watt-seconds for 2000 tasks. These results underscore the inefficiency of these methods in energy management compared to Qwhale.
* The Whale algorithm, while better than Q-learning and SARSA, still consumes considerably more energy than Qwhale, especially as task volume grows (21267 watt-seconds for 2000 tasks).

The Qwhale algorithm offers significant improvements in energy consumption for task scheduling in cloud computing environments. Its hybrid approach effectively combines Q-learning with the Whale Optimization Algorithm, leading to enhanced efficiency and lower energy use. This makes Qwhale particularly suitable for large-scale data centers where energy consumption is a critical factor. The consistent performance of Qwhale across various task loads demonstrates its scalability and robustness, marking it as a superior choice for modern cloud resource management.



SARSAWhale Performing better in High Load conditions [500 VMs]



Tasks-10,000; 20,000; 50,000 respectively

SARSAWhale shows better efficiency and scalability compared to QWhale as the number of tasks increases. The performance gap between the two algorithms narrows as the task count increases, but SARSAWhale still maintains a consistent edge.

The superior performance of SARSAWhale under high load conditions can be attributed to its reinforcement learning-based approach, specifically the SARSA (State-Action-Reward-State-Action) algorithm. This method enables SARSAWhale to dynamically adapt to changing conditions and optimize decision-making processes based on current states and anticipated rewards. Unlike static algorithms, SARSAWhale continuously evaluates actions not only based on the present state but also with consideration for future states, leading to more informed and proactive resource allocation. The algorithm’s capability to fine-tune its performance through ongoing feedback from the environment allows for efficient optimization of resource utilization. This adaptability and continuous learning enhance SARSAWhale’s scalability, making it particularly effective in managing increased workloads. In high load scenarios, where task arrival rates and resource demands are highly variable, SARSAWhale’s robust approach ensures efficient handling and better overall system throughput. Thus, SARSAWhale’s advanced adaptive mechanisms and resource management strategies result in significant performance improvements over traditional algorithms, especially as the number of tasks increases.