Research and Reports Week 5

Name: Bingying Liang

ID: 48999397

CS7346 Cloud Computing

July 4 2023

Can optimal strategies for the five classes of policies, admission control, capacity allocation, load balancing, energy optimization, and QoS guarantees be actually implemented in a cloud? The term “optimal” is used in the sense of control theory. Support your answer with solid arguments. Optimal strategies for one may be in conflict with optimal strategies for one or more of the other classes. Identify and analyze such cases.

Cloud computing systems are highly complex and involve managing a broad range of resources and user demands, often involving trade-offs between different objectives. The five classes of policies mentioned – admission control, capacity allocation, load balancing, energy optimization, and Quality of Service (QoS) guarantees – represent some of the core challenges that these systems must address[3].

1. Admission Control:

   This policy determines which jobs to admit into the system. The optimal policy could be to accept all jobs, but that may lead to system overload and degradation in QoS[2].

2. Capacity Allocation:

   This policy involves allocating resources to different services, tasks, or users. The optimal policy from a capacity standpoint may be to allocate more resources to the tasks that require them, but this could cause certain tasks to monopolize resources and lead to imbalance.

3. Load Balancing:

   This policy distributes work across multiple computing resources to maximize throughput, minimize response time, avoid overload, etc. The optimal load-balancing strategy may involve distributing workload evenly, but this could conflict with energy optimization if some resources are more energy-efficient than others.

4. Energy Optimization:

   This policy aims to reduce energy usage. The optimal strategy may involve prioritizing energy-efficient resources, but this may conflict with QoS guarantees if these resources are not as performant or reliable.

5. QoS Guarantees:

   This policy aims to ensure a certain level of service quality. The optimal strategy may involve prioritizing high-priority tasks, but this could conflict with load balancing or capacity allocation if these tasks consume a disproportionate amount of resources[3].

However, there are inherent conflicts in optimally implementing these policies due to the trade-offs involved. To further analyze these trade-offs:

1. Admission control vs QoS:

   If user aims to maximize admission (i.e., accept all jobs), user might overload resources, leading to degradation in the quality of service provided to each task. So, the two objectives are in conflict.

2. Capacity allocation vs Load balancing:

   Over-allocating resources to high-demand tasks may lead to optimal usage of capacity, but it can also result in an imbalanced system where some resources are overutilized while others are underutilized.

3. Load balancing vs Energy optimization:

   Distributing workload evenly across all resources may lead to optimal load balancing but may also result in higher energy usage if some resources are less energy-efficient than others.

4. Energy optimization vs QoS:

   Reducing energy usage by utilizing more energy-efficient resources may conflict with QoS if those resources are slower or less reliable, leading to slower response times or more frequent service disruptions[4].

5. Admission Control vs Capacity Allocation:

   Optimal admission control requires accepting as many requests as possible, but this might conflict with optimal capacity allocation. Over-acceptance of requests might result in suboptimal allocation, leading to overutilization of resources and possible degradation of system performance[2].

While it's challenging to optimize across all of these areas simultaneously, techniques such as machine learning and control theory can be used to develop policies that strike a balance between these competing objectives. This involves constructing a mathematical model of the system's behavior, defining an objective function that reflects the system's desired goals, and using optimization algorithms to find the policies that maximize (or minimize) this function.

As continue to improve understanding and modeling of these systems, and as optimization algorithms become more sophisticated, it's possible that we'll be able to find strategies that perform well across all of these dimensions[1]. However, it's also likely that there will always be some trade-offs to consider, and the optimal strategy in a given situation will depend on the specific priorities and constraints of the system in question.

To address these conflicts, solutions often involve compromise, complex decision-making algorithms, machine learning and predictive analytics to dynamically adjust policies based on current workload and system state. These approaches aim to provide near-optimal performance across all dimensions, rather than optimizing for a single dimension at the expense of others. However, note that achieving truly optimal strategies in all these areas at the same time, especially in a dynamic and highly variable environment such as the cloud, is an extremely challenging task.

References

[1] Mehdi Hosseinzadeh Aghdam and Abbas Ali Sharifi. "A novel ant colony optimization algorithm for PAPR reduction of OFDM signals". In: *International Journal of Communication Systems* 34.1 (2021), e4648.

[2] M. Malekimajd et al. "An Optimization Framework for the Capacity Allocation and Admission Control of MapReduce Jobs in Cloud Systems". In: *J. Supercomput.* 74.10 (Oct. 2018), pp. 5314–5348. ISSN: 0920-8542.

[3] Dan C Marinescu. *Cloud Computing: Theory and Practice. Third edition*. Cambridge, MA: Morgan Kaufmann is an imprint of Elsevier, 2023.

[4] Yu Zhang and Sergei Gorlatch. "Optimizing Energy Efficiency of QoS-Based Routing in Software-Defined Networks". In: *Proceedings of the 17th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. Q2SWinet '21. Alicante, Spain: Association for Computing Machinery, 2021, pp. 87–94. ISBN: 9781450390804. DOI: `10.1145/3479242.3487325`. URL: `https://doi.org/10.1145/3479242.3487325`.