Energy-Efficient Task Scheduling in Cloud Computing Environments

AUTHORS

1. Thiruvikram A
Pg & Research Department of Computer Science,
Sri Ramakrishna College Of Arts & Science,
thiruvikram2k@gmail.com

2. Dr. M. Hemalatha
Associate Professor,
Pg & Research Department of Computer Science,
Sri Ramakrishna College Of Arts & Science,
mhemalatha@srcas.ac.in

Abstract

Cloud computing isn't slowing down anytime soon, and that means we need bigger, more powerful data centers to keep up with all the processing, storage, and networking we rely on every day. The catch? These centers burn through a massive amount of electricity. That jacks up bills and makes people worry about the environmental impact. So, making sure we use less energy when scheduling tasks has turned into a real priority. The challenge is to squeeze as much as we can out of these resources without letting energy use spiral out of control—or letting service quality slip.That's where our study comes in. We built an energy-aware task scheduling framework designed just for cloud setups. The system looks at workloads, chooses hosts in a smarter way, consolidates virtual machines on the fly, and actually models power use to bring it down. We split the whole thing into three parts: monitoring, scheduling, and resource management. This keeps the framework nimble and scalable, even if demand keeps growing.When we put everything through its paces, the results spoke for themselves. Our model dropped energy consumption, improved resource use, and kept the whole system reliable—even as the workload shifted around. Bottom line: if we want cloud computing that's greener and more sustainable, energy-aware scheduling isn't just nice to have; it's essential.

Keywords

Cloud Computing, Energy Efficiency, Task Scheduling, Virtual Machine Consolidation, Green Computing, SLA Management.

Introduction

Cloud computing has turned the old way of handling computers on its head. These days, companies lean on the cloud for just about everything—running apps, storing mountains of data, crunching massive numbers. But all that magic comes from huge data centers humming along 24/7. They suck up crazy amounts of electricity, no matter what. Even when servers are barely doing anything, they're still gobbling up almost as much power as when they're slammed. It's honestly a huge waste.Most scheduling algorithms don't really care about energy. They're built to boost speed, cut down wait times, or spread out the workload, and that's about it. So you end up with servers running when nobody needs them, wasting power and money. And now, with everyone worried about sustainability, it's obvious we need smarter ways to handle all this. Scheduling shouldn't just be about speed—it should help save energy too. That's where energy-efficient scheduling comes in. It lines up tasks with resources in a way that keeps power bills down, but still delivers the performance and reliability everyone expects.

Objective of study

What's this study aiming for? In short: a smarter way to schedule tasks in the cloud so we use less energy. The plan is to lower how much power physical servers chew through—by spreading out the work more intelligently and handling virtual machines more efficiently. But it's not just about saving energy. The system also has to stick to SLA standards, which means keeping delays low and performance steady. On top of that, the idea is to squeeze more out of the hardware we already have—move workloads around when needed and power down servers that aren't pulling their weight. The researchers put their new energy-aware approach up against old-school scheduling to see how it really stacks up in real-world scenarios.

Methodology

We kicked off the energy-efficient task scheduling system with a pretty straightforward plan: figure out what the thing needs to do, sketch out the structure, dive into the algorithms, actually build it, and then see if it does what we hoped. Right from the start, we really got into the weeds with how cloud infrastructure works—how workloads shift around, how servers get used, and how much juice everything pulls. That deep dive made it clear what our system had to handle: it needed to keep a close eye on what was happening in real time, predict what might happen next, and make smart calls about where each task should go. We weren't just thinking about the basics, either—we wanted something that could grow, wouldn't break under pressure, and always kept energy savings front and center.We split the system's architecture into layers: monitoring, scheduling, and resource management. The monitoring part checks things like CPU load, memory, and power for every machine. The scheduler grabs all that info, sizes up the situation, and figures out the best place to send each task—always with energy savings in mind. Then there's the virtual machine manager, which shuffles workloads off half-asleep servers so those can shut down instead of just burning power for nothing.To help the system make better choices, we built an energy consumption model that tracks how hard each server's working and estimates its power use. Before the scheduler sends a new task anywhere, it calculates how much extra energy that move will cost, then picks the spot with the lowest impact. We also brought in workload prediction, so the system can see resource spikes or dips coming and stay ready. The end result? Tasks keep flowing smoothly, and we cut down on wasted energy.

System testing

We really put the new scheduling framework through its paces. We started by picking apart every piece—the monitoring module, the scheduling engine, the virtual machine manager—just to make sure each one did its job. Once we felt good about that, we plugged everything together and watched how the whole system behaved. Cloud setups always have a few tricks up their sleeves, so we kept our eyes open.When it came to performance, we hit the system with all sorts of workloads. We wanted to see how fast it reacted and if it actually followed the service level agreements, not just on paper but in the real world. Then we cranked things up with some heavy load tests—think a flood of tasks all at once—to see if the system would keep its cool. We didn't stop there. We checked energy use too, stacking up our results against old-school algorithms like First-Come-First-Serve and Round Robin, just to see if we'd made a real difference. And, you know, security and reliability are a big deal. So we made sure the system could run safely and stay solid, even when it was spread out across different locations.

Results

The new energy-efficient scheduling model really delivered. It saved a lot of energy by powering down extra physical hosts when there wasn't much work to do. Plus, it made better use of resources by packing workloads together and spreading out tasks more intelligently. There were fewer SLA violations than before, so improving energy efficiency didn't mean sacrificing performance. In short, bringing power-aware metrics into the scheduling process made everything run smoother and took cloud computing a step closer to being truly sustainable.

Conclusion

Building a smarter, energy-saving task scheduler for cloud computing isn't just about keeping an eye on things. It's about predicting what's coming, picking the right hosts, and grouping virtual machines to cut down on power use—without messing up performance. This approach proves you really can balance top-notch service with being environmentally responsible. Add in

AI, better analytics, and hybrid optimization, and cloud systems get even more powerful and efficient. That's how we move closer to truly green data centers.

Bibilography

1. Beloglazov, A., and Buyya, R., "Energy-Efficient Resource Management in Virtualized Cloud Data Centers," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

2. Buyya, R., Broberg, J., and Goscinski, A., *Cloud Computing: Principles and Paradigms*, Hoboken, NJ, USA: Wiley, 2011.

3. Barroso, L. A., and Hölzle, U., "The Case for Energy-Proportional Computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.

4. Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M., *Feedback Control of Computing Systems*, Hoboken, NJ, USA: Wiley-IEEE Press, 2004.

5. Sommerville, I., *Software Engineering*, 10th ed., Boston, MA, USA: Pearson, 2015.