Cluster Resource Management for Sustainable and Efficient Computing

1st Andrei Bersatti
School of ECE
Georgia Institute of Technology
Atlanta, United States

2nd Aaron Jezghani
PACE
Georgia Institute of Technology
Atlanta, United States

***Abstract*—Recent years have brought about a sense of urgency regarding global environmental health and this shines attention into the environmental impact of modern computing data centers. These are mostly powered by electricity originating from carbon sources. In addition to the general concern for the environment, modern data centers are only increasing in their thirst for power. The latest data center applications and the latest foundation machine leaning models are examples of this. In 2021, the Georgia Institute of Technology transitioned into a consumption-based model for its centralized computing resources and further adopted SLURM for cluster management and for job scheduling. This transition resulted in greater utilization, among other metrics, than with the prior system based on Torque/Moab.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Modern computing data centers are facing what sometimes can be dueling priorities. On the one hand, modern data centers must adopt new technologies and capacities as advancements in technologies and techniques enable resource-hungry applications that in the past were the subject of sci-fi films. On the other hand, data center administrators must carry out environmental stewardship in their operations by increasing the sustainability and the efficiency of their operations.

A. Need for Energy Efficient Computing

When OpenAI introduced ChatGPT [1] at the end of 2022, the public at large first became aware of and amazed at the power of large language models. For data center administrators, however, it was the continuation of the rise in the demand for data center computing resources brought about by the development of such innovations [1], [2]. This rise in demand for these is part of a larger worrisome trend. In 2022, estimated global data center electricity consumption was 240-340 TWh [3]. Data centers worldwide currently consume between 1% and 2% of overall power, and by 2030 data center power consumption worldwide is projected to grow into 3% to 4% of global power [4]. Climate researchers have found a correlation between coal-generated electricity and GHG emissions in all countries [5]. This electrical power is sourced primarily from carbon fossil fuels which contribute to a growing global carbon footprint and the resulting adverse environmental effects of a warming planet such as heatwaves, heavy precipitation, droughts, and tropical cyclones [6]. On the other hand, AI models such as LLMs also have a significant and worrisome water footprint [7]. Exacerbating the problem is the fact that the demand for resources and its environmental effects are increasing due to modern AI and data-heavy workloads [8]–[10]. The demand for resources has accelerated to the point that large tech corporations have made modifications to tamper their expectations and approaches to tackle their carbon footprint [11]. Such a rise in appetite for computing power is a result of resource-demanding cutting edge applications such as training of Llama 3.1 which took 39.3M GPUHours to train on 700W hardware, generating a carbon footprint of 11,390 tons CO2eq. [2], [12]. Future systems are projected to increase their hunger for power, such as the 1500W TDP Intel Falcon Shores [13] which is a hybrid processor which combines x86 CPUs and Xe GPU cores, or such as Nvidia's 120kW TDP DGX GB200 NVL72 rackscale system [14].

B. Current Challenges

The need for energy efficient computing is faced with a variety of challenges. Those challenges can be categorized into challenges to integrate efficiency and challenges specific to academia.

1) Challenges to Integrate Efficiency: The challenges to integrate efficiency can arise from generic hardware and software considerations and from transparency in reporting. Different generations of hardware and different usage scenarios will affect efficiency integration differently. For example, newer and more powerful equipment might have a higher TDP but also it might be able to achieve greater FLOPS. Also, different GPU architectures may support different frameworks with varying efficiencies (CUDA, OpenCL). Furthermore, even for the same hardware there may be differing software development workflow and software stack which can present efficiency challenges. Finally, besides the challenges presented by hardware considerations and by workflow and software development considerations, an additional challenge arises in the varying measures of transparency in reporting when referring to efficiency and particularly about sustainability.

2) Challenges Specific to Academia: There exist additional challenges specific to an academic environment besides the generic challenges to integrate efficiency. Due to the nature of cutting edge research in academia and the budget limitations of academia versus industry, the renovation cycle of systems and equipment may be slower in academia. This can lead to an issue of aging infrastructure that is more marked than in industry. Furthermore, the already mentioned budget limitations can force some limitations that may not occur in industry.

Besides the operations differences specific to an academic setting, a very particular user base exists for an academic setting data center. This can present very particular challenges. For example, there may be faculty and user expectation for ”dedicated” resources. Another particularity of academic research is that, given that research tends to be carried out by students, there is a high researcher turnover (4-5 year tenure). This brings along its particular challenges such as a periodic loss of know-how in the user base.

C. High Level Overview of Solutions Implemented Today

Currently, computing clusters employ a variety of software, documentation, and training to achieve performance and efficiency goals. Various high-level software frameworks for portability and performance are used. Examples of these are OpenACC [15], Kokkos [16], and OpenMP Offload [17]. Beside the software, user education plays a role in managing resources in a sustainable and efficient manner. This takes the form of training via documentation and other means. Finally, transparency and completeness in reporting is paramount. In particular energy allocating and reporting must improve. For example, data centers might transition into using joule allocations rather than the current Service Units (SUs) allocation model.

II. Phoenix Cluster at GT

In April 2021, Georgia Institute of Technology transitioned centralized computing resources managed by the Partnership for an Advanced Computing Environment (PACE) from a condominium to a consumption-based model [18]. Apart from the opportunity for more sustainable operations through partial cost recovery, this change was driven by challenges in resource provisioning for faculty due to challenges such as procurement cycle timing and spend limits, supply-chain availability, and datacenter footprint optimization.

The primary research cluster ”Phoenix” is a heterogeneous resource, with approximately 1,400 servers, 35,000 CPU cores, 330 GPUs, and 7 PB of mixed storage. All resources are connected via 100 Gbps InfiniBand (IB) in a fat-tree topology, with all nodes in a rack connected to a switch, which then has 8x100 Gbps connected to one of 4 leaf modules in the central manager switch. The cluster was originally populated with servers using Intel Cascade Laek CPUs and DRAM totaling 192GB, 384GB, or 768GB; CPU nodes either provided moderate (1.5TB) NVMe or large (8.0TB) SAS local storage, while GPU nodes provided either 2x NVIDIA Tesla V100-PCIe (16GB and 32GB models) or 4x NVIDIA Quadro Pro RTX6000 accelerators. Further expansions through early 2024 introduced AMD Milan and Genoa CPUs, as well as NVIDIA Ampere A100 (40GB and 80GB PCIe, as well as SXM4) and Hopper H100 (SXM5) nodes.

Fig. 1. Phoenix is a heterogeneous compute cluster with approximately 1,400 servers in the Coda datacenter in Midtown Atlanta.

A. Resource Management

The initial resource allocation and accounting was managed using the Moab HPC Suite [19], with Moab for workload scheduling, Torque for resource management, and MAM for accounting. Each node architecture was designated as a quality of service (QOS) in Moab with a unique charge rate in MAM, while job priority and scavenged cycles were maintained through ”inferno” and ”embers” queues. Standing reservations were created based on task counts and node features, as the length of hostnames exceeded Moab string limits. To simplify user submissions and simultaneously provide enhanced scheduling policy support, a submit filter was implemented to map resource requests to the proper standing reservation, and thus QOS, as well as alert users if the job request violated policy.

In late 2021, the scheduling policy was updated to allow spillover of smaller memory jobs to idle, larger-memory architectures. This change was implemented to increase overall cluster utilization, as the majority of jobs required less than 8GB of memory per CPU, leaving larger-memory architectures idle. In Moab, this was achieved by updating the standing reservation configuration with additional QOS designations for the larger-memory architectures. Preference for the most appropriate QOS was implemented by indicating reduced priority for spillover in the standing reservation configuration.

Additional scheduling optimization was configured in the scheduler using node feature sets to align jobs based on InfiniBand fabric topology. Although Moab could direclty support topology-aware scheduling, this required that the workload manager be configured with all node identities, which was contrary to documented recommendations that all node management and configuration be left to the resource manager, Torque. To address this, each node was configured with features based on rack number, IB leaf side, and IB leaf, and then Moab was given a `NODESETLIST` defined with rack features, followed by IB leaf side features, and then IB leaf features, always proceeding from least-populated to most-populated groupigns, so that large jobs could theoretically be localized as much as possible.

Admittedly, this configuration presented challenges in transparency in reporting, as tools like XDMoD [20] lacked support for QOS tracking. Furthermore, writing an effective Torque submit filter proved challenging, for reasons such as:

- precedence rules for command line and batch directives had to be manually implemented to ensure resource requests were processed as the scheduler would receive them,
- regular expressions needed to be carefully constructed to correctly parse user scripts and accommodate the variety of Moab and Torque syntax, and
- the script needed to be written in such a way that use environment would not interfere with its functionality.

Nonetheless, the scheduler configuration was able to address the complex needs of scheduling while automatically routing user submissions to the correct bin based entirely on resources.

B. Impact on Resource Utilization

As reported in the PEARC22 paper, the transition to the consumption model brought about a number of favorable changes in the cluster's impact. Firstly, overall resource utilization nearly doubled after the first year, going from an average of 35% to 55%. Secondly, average job sizes increased by 40%. Lastly, average queue wait times decreased to well under an hour. All of these changes seem to be tied to the lack of fencing in the form of restrictive queues for active users, and the considerably larger collective pool of resources available for job submissions.

C. Challenges to Efficiency

Despite the successful initial deployment that led to the transition in operational model for research computing with PACE at Georgia Tech, there were several limitations with the Moab HPC Suite for cluster management. In particular, the incompatibility with operating systems newer than RHEL7 caused friction with adopting newer, more efficient architectures that required newer kernels for support. Additionally, after the first year, it was discovered that communication failures between the scheduler components required increased timeouts to ensure success in transmission of job and accounting data, which greatly reduced overall job throughput, as jobs sometimes took as long as 10 minutes before being enqueued by Moab. Coupled with needs for more robust scheduling capabilities, PACE proceeded to explore alternative options.

III. MIGRATION TO SLURM

In 2022, PACE began working on a transition to the Slurm cluster management and job scheduling platform [21]. Rather than adopting the Torque wrapper distributed with the scheduler [22], PACE opted for a staged migration, moving portions of each node class over a period of 4 months to allow users an opportunity to transition gradually, so that their research would be minimally impacted.

A. Translating Moab Configurations

The hardest part of the transition to Slurm was successfully preserving the job accounting functionality originally provided by MAM. Ultimately, the decision was made to leverage `TRESBillingWeights` on a per-partition basis to set the appropriate charge rate per node class; however, as usage is only tracked in integer minute quantities, the values were set large enough to provide precision sufficient to accurately match the 4 decimals of precision established by the rate study. Ultimately, this led to 10 partitions each for internal and external charge rates, as the relationship between the two is not linear in nature. An additional 7 partitions are provided for administrative or niche functionality.

As the accounting required node classes switch from QOS to partitions under Slurm, that meant the "inferno" and "embers" submissions moved from Torque queues to Slurm QOS designations. This offered considerable advantage, as the flexibility of QOSes in Slurm prevented accidental overlap in policy that had manifested under the combination of Torque and Moab.

As with Torque, a job submit filter was implemented to map resource requests to the appropriate partitions and reduce friction with job submissions. Fortunately, the job submit filter provided several benefits, including using a standardized API to properly capture resource requests, supporting Lua for ease of development, and residing on the controller node for simplified management and improved security. Rather than writing and routing custom error messages when users violate policy, the job submit plugin provides an extensive list of possible error codes and corresponding messages, although apart from a select subset, most have to be determined by reviewing the source code [23].

B. Enhanced Scheduling and Accounting Capabilities

With the adoption of Slurm, PACE transitioned to a local PMIx installation to provide a common library against which Slurm and the various MPI libraries installed in the PACE applications stack could be compiled. Additionally, Slurm's PMI2 library was built to support Intel MPI installations, and all modules were configured to set variables such as `SLURM_MPI_TYPE` appropriately. Largely, these decisions stemmed from the desire to simplify workflows under the umbrella of a single syntax so that users could more effectively run parallel jobs on cluster resources.

Additionally, Slurm was built with `hwloc` and NVML support where appropriate, so that intranode resource affinity could be more effectively implemented. Beyond the obvious performance gains due to smarter task placement, these additional libraries and their associated plugins allowed for resource utilization metrics to be passively captured and tracked in the accounting database, so that users can more readily assess their efficiency for jobs submitted to the cluster. This capability was further enhanced when GPU SM and memory occupancy was included with `TRESUsageIn` fields in the accounting database.

Lastly, PACE implemented the `pam_slurm_adopt` module to address challenges seen with interactive sessions causing resource contention on computational nodes. In particular, there had been several instances of VSCode [24] opening external sessions on allocated compute nodes, but running on hardware assigned to other jobs. This module places external login sessions under the most recent job step, mitigating the issue; however, one challenge PACE encountered was the apparent conflict with `pam_systemd`, which was necessary to gracefully manager administrative sessions for activities such as system reboots. A solution was achieved by using `pam_succeed_if` to enable `pam_systemd` rather than `pam_slurm_adopt` for administrative login sessions.

C. Addressing Node Class Affinity

Even under Moab, PACE had become aware that while the memory spillover capability increased overall cluster utilization by allowing jobs requiring less memory to run on idle servers, it would often prefer topology-aware scheduling over parameters designed to map jobs correctly to the appropriate resource. In particular, Slurm's topology plugin prevents node weights from being considered in scheduling algorithms, so its use precluded any means to indicate which nodes should be scheduled first within a partition.

Fortunately, Slurm release 22.05.4 [25] added the capability to include the `--prefer` flag in the LUA job submit plugin to re-prioritize node class features over topology. Although features requested using this method are optional, PACE discovered caveats to their use, primarily that (1) sufficient resources must exist with the designated feature, and (2) if satisfied, the prefer flag will be used in place of job constraints. Thus, the job submit plugin required updating to not only add the appropriate node class features to the

These lessons have recently proved useful, as the large number of partitions in the current configuration presented a challenge in the upgrade to RHEL9. Rather than moving whole partitions or creating new partitions as resources were converted, an OS-based feature was introduced on all nodes. By using a CLI filter to set default requests to the OS matching the submission host, and then gracefully handling the feature through submission via the job submit plugin, user jobs are guided by default to compute resources of the same OS version, thus reducing the likelihood of job failures due to mismatched software.

IV. Impact of Scheduler

As part of our study of the impact of the scheduler and the changes in scheduler we present the following results gathered utilizing [20]. In these results, "Phoenix" represents 'Phoenix utilizing Torque,' and "Phoenix Slurm" represents 'Phoenix utilizing Slurm.' Of note is that we believe it is possible that Torque utilization numbers are inflated due to issues such as same job counted multiple times. Figure 2 shows the XDMoD generated Cluster Utilization CPU Hours total. Figure 3 shows the Cluster Utilization CPU Hours per job. In order to portray a clearer idea of the utilization, we show a chart by allocated FLOPS as well as a normalized chart for it in Figures 4 and 5. Figures 6,7, 8, and 9 portray the data on job submissions, jobs running, jobs started, and jobs ended. Of note is that the improvement in FLOPS allocated is clearly inferred visually from the picture, particularly in Figure 4, as it shows a clear increase in FLOPS allocated towards January 2023, which Figures 6,7, 8, and 9 make clear that it corresponds to the period of transition from Torque to SLURM.

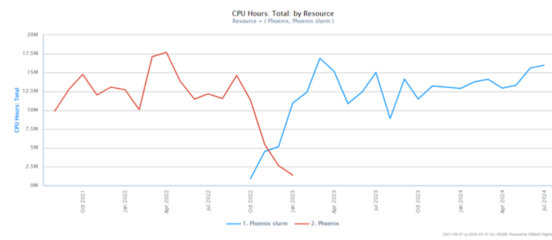

Fig. 2. Phoenix utilization by CPU Hours for Torque and for SLURM.

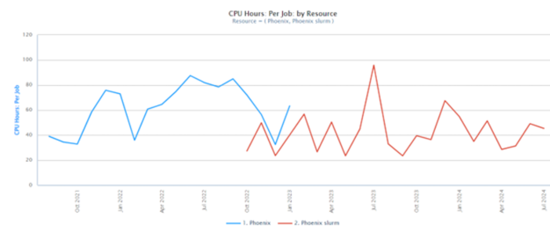

Fig. 3. Phoenix utilization by CPU Hours per job for Torque and for SLURM.

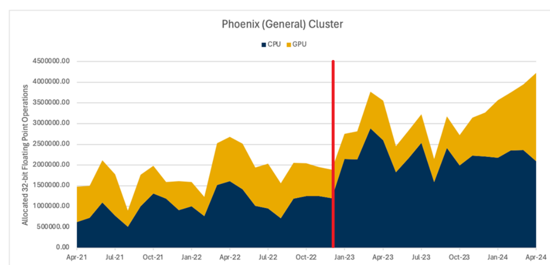

Fig. 4. Phoenix Allocation of FLOPs.

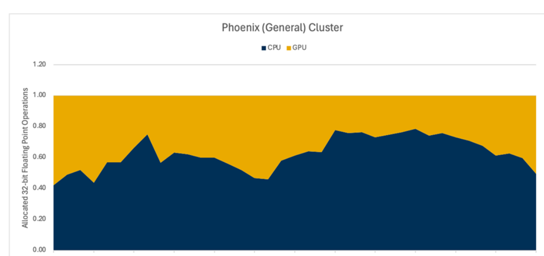

Fig. 5. Phoenix Allocation of FLOPs in normalized form.

Figure 10 shows the impact of scheduling policy on placement of jobs on cluster resources. For each day, the number of nodes matching the requested resource type is divided by the number of nodes allocated for jobs to determine the node affinity ratio. A value of 1 indicates 100% of jobs are assigned

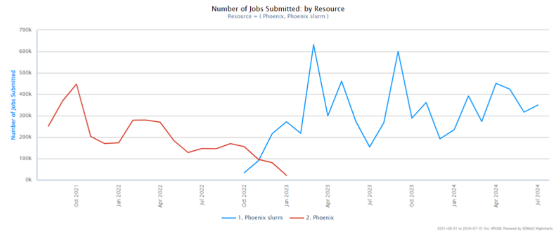

Fig. 6. Throughput in Jobs Submitted for Torque and SLURM.

Fig. 7. Throughput in Jobs Started for Torque and SLURM.

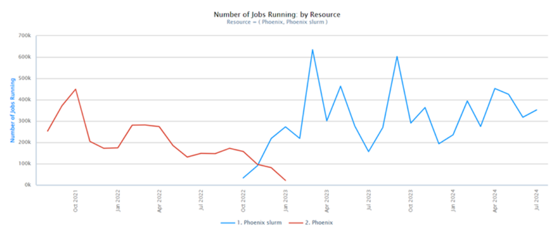

Fig. 8. Throughput in Jobs Running for Torque and SLURM.

Fig. 9. Throughput in Jobs Ended for Torque and SLURM.

to their designated node types, while 0 means no nodes match the appropriate type. Notably, the Torque data prior to 2022 is nearly 1 at all times, largely because memory spillover was not enabled, and thus jobs could only run on their assigned hardware. Once the new capability was enabled to improve cluster utilization and reduce wait times, the ratio plummeted, and continued through to the migration to Slurm. However, with the introduction of the `--prefer` capability in the job submit plugin, node affinity improved considerably, meaning that resources could be more effectively used.

Similarly, Figure 11 looks at the efficacy of the scheduler algorithms to localize jobs according to the IB topology. Here, each level of the tree is assigned a numerical value, where single nodes are assigned a value of 0, single rack job are assigned a value of 1, single sides of an IB leaf are designated

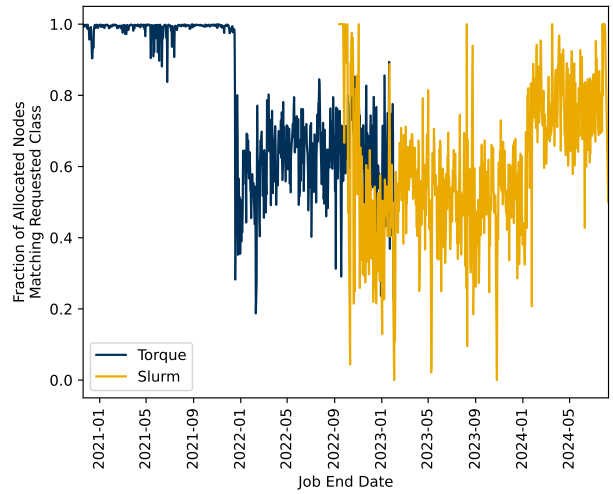

Fig. 10. Node class affinity as impacted by scheduling method. At the end of 2021, memory spillover was enabled on Torque, which allowed for better overall utilization of resources, but reduced node class affinity considerably. With the transition to Slurm, the same trend was observed until the '–prefer' flag was introduced in early 2024.

2, single IB leafs are valued at 3, and all other jobs are assigned 4. The timeseries data reflects the average value for all jobs with daily granularity. Immediately clear is that Moab using the `NODESETLIST` approach appears to be more effective at localizing jobs, although comparatively, more and larger jobs were run under Slurm, which in theory should make it harder to achieve locality. Additionally, there are additional parameters in the Slurm configuration that merit investigation to see if the topology-aware scheduling can be more effective.

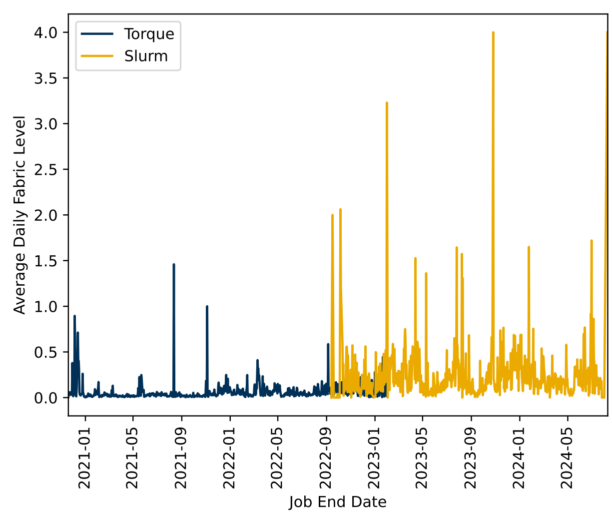

Fig. 11. Results from topology aware scheduling implemented using node feature sets in Moab and the fat-tree topology plugin for Slurm, averaged across all jobs per day. Greater values indicate more hops in the interconnect fabric, which results in reduced overall efficiency from higher latency in communication. On average, the approach in Moab was more effective at localizing jobs.

V. Future Work

Future efforts are focused on enhancing existing capabilities and adding new features to provide more transparency into cluster efficiency and user control. Recently, Stanford's SPANK Lua plugin was deployed with the intent to provide capabilities for users interested in exploring GPU and CPU power management for improved efficiency in workflows. Similarly, efforts are ongoing to effectively implement power reporting using the IPMI plugin, although challenges with inconsistent support across server manufacturers are presenting a challenge. Similarly, efforts are currently underway to explore opportunities in dynamic node power management through suspend/resume workflows, should there be an ability to reduce capacity at times.

Looking further into the future, we aim to obtain better insight on the carbon footprint for jobs. Furthermore, we aim to propose and measure techniques that will help scheduling to minimize carbon footprint particularly in academic settings. For this, we aim to leverage previous lessons on temporal and spatial migration of workloads [26]–[32], with the differentiating characteristic that data centers in academic environments are not as distributed as the hyper-scale data centers belonging to the large tech corporations and which have been studied more. As a result, we can consider temporal shifting for flexible local workloads and spatial shifting through cloud migration for appropriate workloads [33], taking advantage of the fact that cloud services tend to have optimized PUEs and may offer the tradeoff of relaxing deadlines versus processing in low carbon footprint datacenters.

VI. Conclusion

The transitions into a consumption-based model and the transition to utilizing SLURM have resulted in greater utilization, greater control, and greater visibility into the job efficiency. In particular, the transition into SLURM has enabled new opportunities for reporting and for greater control into the scheduling. These opportunities will result in enhancing SPANK plugins such as GPU power management and CPU governor/turbo settings as well as IPMI plugin implementation.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[3] I. E. A. (IEA), "Data centres & networks - IEA — iea.org," https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks, [Accessed 07-08-2024].

[4] G. S. Research, "Ai is poised to drive 160% increase in data center power demand," 2024. [Online]. Available: https://www.goldmansachs.com/insights/articles/AI-poised-to-drive-160-increase-in-power-demand

[5] E. Kocak, E. E. Ulug, and B. Oralhan, "The impact of electricity from renewable and non-renewable sources on energy poverty and greenhouse gas emissions (ghgs): Empirical evidence and policy implications," *Energy*, vol. 272, p. 127125, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544223005194

[6] J. Borland, P. Dakins, D. Johnson, and R. Williams, "Ipcc, 2023: Summary for policymakers. in: Climate change 2023: Synthesis report. a report of the intergovernmental panel on climate change." Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, Tech. Rep., 2023.

[7] P. Li, J. Yang, M. A. Islam, and S. Ren, "Making ai less "thirsty": Uncovering and addressing the secret water footprint of ai models," 2023. [Online]. Available: https://arxiv.org/abs/2304.03271

[8] S. Narasimhan, "Beyond 'Data-Driven': How Energy-Efficient Computing for AI Is Propelling Innovation and Savings Across Industries — blogs.nvidia.com," https://blogs.nvidia.com/blog/energy-efficient-ai-industries/, [Accessed 07-08-2024].

[9] S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing: Watts driving the cost of ai deployment?" in *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2024, pp. 85–99.

[10] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," 2021. [Online]. Available: https://arxiv.org/abs/2104.10350

[11] A. Rathi, "Google is no longer claiming to be carbon neutral," *Bloomberg.com*, Jul. 2024. [Online]. Available: https://www.bloomberg.com/news/articles/2024-07-08/google-is-no-longer-claiming-to-be-carbon-neutral

[12] HuggingFace, "meta-llama/meta-llama-3.1-70b," Aug. 2024. [Online]. Available: https://huggingface.co/meta-llama/Meta-Llama-3.1-70B

[13] A. Shilov, "Intel's 1500w tdp for falcon shores ai processor confirmed — next-gen ai chip consumes more power than nvidia's b200," May 2024. [Online]. Available: https://www.tomshardware.com/pc-components/gpus/intels-1500w-tdp-for-falcon-shores-ai-processor-confirmed-consumes-more-power-than-nvidias-b200

[14] T. Mann, "A closer look at nvidia's 120kw dgx gb200 nvl72 rack system." [Online]. Available: https://www.theregister.com/2024/03/21/nvidia_dgx_gb200_nvk72/

[15] [Online]. Available: https://developer.nvidia.com/openacc/

[16] C. R. Trott, D. Lebrun-Grandié, D. Arndt, J. Ciesko, V. Dang, N. Ellingwood, R. Gayatri, E. Harvey, D. S. Hollman, D. Ibanez, N. Liber, J. Madsen, J. Miles, D. Poliakoff, A. Powell, S. Rajamanickam, M. Simberg, D. Sunderland, B. Turcksin, and J. Wilke, "Kokkos 3: Programming model extensions for the exascale era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 805–817, 2022.

[17] R. K. M. Nitya Hariharan, "Use openmp* accelerator offload for heterogeneous architectures," *Intel*. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/openmp-accelerator-offload-heterogeneous-arch.html

[18] A. Jezghani, S. Sarajlic, M. Brandon, N. Bright, M. Belgin, G. Beyer, C. Blanton, P. Buffington, J. E. Coulter, R. Lara, L. Lefton, D. Leonard, F. C. Liu, K. Manalo, P. Manno, C. Moseley, T. Nightingale, N. B. Bonner, R. Rahaman, C. Stone, K. J. Suda, P. Wan, M. D. Weiner, D. Womack, N. Zhang, and D. Zhou, "Phoenix: The revival of research computing and the launch of the new cost model at georgia tech," in *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You*, ser. PEARC '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3491418.3530767

[19] A. Computing, "Moab-torque/pbs integration guide." [Online]. Available: https://docs.adaptivecomputing.com/mwm/7-0/Content/pbsintegration.html#overview

[20] J. T. Palmer, S. M. Gallo, T. R. Furlani, M. D. Jones, R. L. DeLeon, J. P. White, N. Simakov, A. K. Patra, J. Sperhac, T. Yearke, R. Rathsam, M. Innus, C. D. Cornelius, J. C. Browne, W. L. Barth, and R. T. Evans, "Open xdmod: A tool for the comprehensive management of high-performance computing resources," *Computing in Science & Engineering*, vol. 17, no. 4, pp. 52–62, 2015.

[21] M. A. Jette and T. Wickberg, "Architecture of the slurm workload manager," in *Job Scheduling Strategies for Parallel Processing: 26th Workshop, JSSPP 2023, St. Petersburg, FL, USA, May 19, 2023, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2023, p. 3–23. [Online]. Available: https://doi.org/10.1007/978-3-031-43943-8_1

[22] SchedMD, "How easy is it to switch from pbs or torque to slurm?" 2024. [Online]. Available: https://slurm.schedmd.com/faq.html#torque

[23] ——, "slurm_errno.h," 2024. [Online]. Available: https://github.com/SchedMD/slurm/blob/master/slurm/slurm_errno.h

[24] Microsoft, "Visual studio code," 2024. [Online]. Available: https://code.visualstudio.com/

[25] SchedMD, "Changes in slurm 22.05.4," 2022. [Online]. Available: https://github.com/SchedMD/slurm/blob/slurm-22-05-4-1/NEWS

[26] A. Radovanović, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, S. Talukdar, E. Mullen, K. Smith, M. Cottman, and W. Cirne, "Carbon-aware computing for datacenters," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1270–1280, 2023.

[27] J. Zheng, A. A. Chien, and S. Suh, "Mitigating curtailment and carbon emissions through load migration between data centers," *Joule*, vol. 4, no. 10, pp. 2208–2222, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542435120303470

[28] A. Bersatti, E. Kim, and H. Kim, "Quantifying co2 emission reduction through spatial partitioning in deep learning recommendation system workloads," *IEEE Micro*, pp. 1–8, 2024.

[29] J. Lindberg, B. C. Lesieutre, and L. A. Roald, "Using geographic load shifting to reduce carbon emissions," 2022. [Online]. Available: https://arxiv.org/abs/2203.00826

[30] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud," in *Proceedings of the 22nd International Middleware Conference*, ser. Middleware '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 260–272. [Online]. Available: https://doi.org/10.1145/3464298.3493399

[31] I. n. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: managing datacenters powered by renewable energy," in *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 51–64. [Online]. Available: https://doi.org/10.1145/2451116.2451123

[32] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and challenges for data center demand response," in *International Green Computing Conference*, 2014, pp. 1–10.

[33] D. Zhao and J. Zhou, "An energy and carbon-aware algorithm for renewable energy usage maximization in distributed cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 156–166, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074373152200079X