Thrifty Workload Planning for Datacenter Sustainability and Efficiency

Nik Sultana
Dept of Computer Science
Illinois Institute of Technology
Chicago, USA

***Abstract*—Climate change and resource scarcity are forcing many changes to our day-to-day decision-making, but as users of datacenters and large-scale networks we are abstracted from choices that affect resource consumption. We can run tasks regardless of the cost of electricity or time of day, and we lack a user interface that allows us to weigh the costs of scarce or variable resources on our workloads—such as the consumption of non-renewable or renewable energy, or the effects of extreme weather. Reducing this mismatch through manual planning is error-prone and laborious; we need automated reasoning support.**

This paper describes ongoing research to make resource scarcity and variability into first-class considerations in the operation of large-scale networks. This forces a more resource-conscious approach to using these networks. The key idea in this research involves using logic-based automated planning to allocate tasks based on dynamic models of resource costs and environmental conditions. This research also leverages programmable network hardware to improve reaction time to different conditions. The research carefully combines these ideas to improve datacenter sustainability and efficiency, and starts by incubating these ideas into a university-scale prototype.

***Index Terms*—Sustainability, Efficiency, Planning, Programmable Networking**

I. Introduction

It is said that "if you fail to plan, then you plan to fail", but planning the behavior of large-scale networked systems is hard and ad hoc. A useful simplification when planning such systems involves *overprovisioning* the resources we rely on. Historically this has been viable since hardware tends to become cheaper over time, particularly as technology tends to become commoditized; so we can improve the reliability of systems through overprovisioned redundancy. Moreover, there has been a steady trend for hardware performance to improve steadily over time without requiring software changes—particularly during the Moore's Law era.

The way we use resources tends to abstract their operating cost, and this leads to waste. We see this with workstations and servers being left on unnecessarily, but cloud resources make this worse since they present a perverse incentive. That is, "waste" is profitable for cloud operators since the users get billed for their usage regardless of whether the usage was necessary or not.

These attitudes might seem viable during periods of abundance, but they fail when resources become scarce. The past few years have brought this into sharp focus: planning is becoming harder because of the simultaneous changes our systems are experiencing. We saw a change in usage patterns because of the pandemic, disruption of equipment supplies and increase in their costs, and the spikes in oil and gas prices are renewing concerns about energy security. Moreover, we are seeing more occurrences of extreme weather patterns arising from climate change, which in turn strains the assumptions we can make about the operating conditions around and inside datacenters, which are usually operated in stable energy and climatic environments.

This underscores the need for innovation in datacenter sustainability, not only for their operators, but also for how users plan their workloads. This innovation would enable fine-grained reasoning about resource prices, variability in cost and availability, and mitigation of risk. In turn this will enable users to be more frugal and thrifty in managing scarce resources, and in managing the cascades in costs or availability as a result of failures or price changes. Thus users' short- and long-term planning and decision-making can better react to change. For example, users might opt to run certain tasks during the day to opt for solar power, or avoid specific times when the datacenter is very busy—and thus more costly to cool.

This paper describes ongoing research into a reasoning framework where resource cost and variability are first-order considerations for users. This is being incubated in a university, which provides access to various workloads—including batch workloads from computational science, or interactive workloads from remote learning. This is inspired by the past success of using universities as incubation environments for new ideas, such as OpenFlow [6] and campus programmable networking [5]. In addition to the university incubation environment, Illinois Institute of Technology offers two more advantages: an independent microgrid [1] and the possibility of feedback loops to be built with grid management infrastructure; and the Ocient Computational Center [2], a state-of-the-art small datacenter with redundant power and links.

The key idea in this research is based on planning [7]–[9] and borrows from symbolic AI. It involves systematically exploring many alternatives based on models of behaviors and costs. The next section gives an example.

II. Example

Fig. 1 shows an example network that extends a model used in past work [7] to separate the processing in two clusters. It shows traffic flows between devices in these clusters, and the

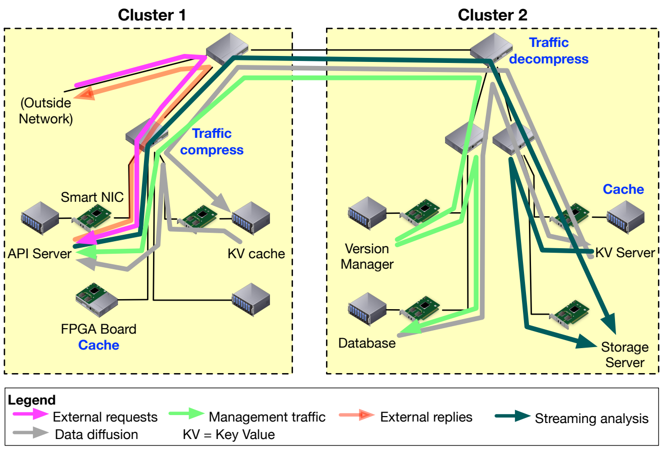

Fig. 1. Inter-dependent workloads and resources across clusters.

allocation of different tasks to different hardware. For example, “Traffic Compress” is assigned to a particular switch, and the Application Server is running on a particular hosts. The traffic flows are a consequence of these allocations.

Past work explored the allocation of tasks to optimize costs and performance, but it ignored external conditions that might effect costs and availability. For example, the models did not include information about periods of cheaper electricity, periods of cooler weather, or expected upgrades that will bring about performance changes.

If you knew that rain is being forecast, you will likely change your behavior: you might carry an umbrella or use a different means of transport, or opt to work from home if it is very stormy. But no such decision making is possible when it comes to datacenter-level tasks.

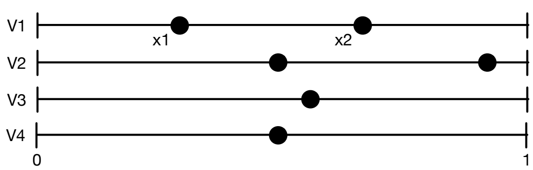

Fig. 2. Models are parametric to scaling conditions.

Fig. 2 shows how models from earlier work are being extended with richer information to enable decision making related to the scarcity and variability of resources. In this sketch, each line refers to a variable in the model—including performance and power use. For example, V_1 has two thresholds, x_1 and x_2, at which different actions can be taken. If V_1 is the price of electricity, then when the price surpasses x_1 we can decide to postpone lower-priority tasks, and at price x_2 we can suspend our system entirely.

III. Methodology and Results so far

The building blocks for the ideas outlined in this paper were first built in Flightplan [9] system for reasoning about programmable dataplane programs,[1] such as those written in P4 [3]. This was then extended for network-aware applications [7], to include application-level requirements in the planning. This was then generalized [8] by replacing the custom reasoning framework with Prolog [4] and formulating models as rules.

The methodology consists of writing Prolog specifications of the resource-needs of hardware and software. These specifications tend to be short—they abstract most features of the hardware and software. These specifications are written according to an API and executed by a planner program [8].

Current results show that this method works for cluster-level workloads even when heterogeneous hardware and various traffic flows are involved [7], and can capture different phases of operation—to distinguish between busy and idle periods.

[1]Flightplan is open-sourced at https://flightplan.cis.upenn.edu

IV. Next steps

Ongoing work consists of extending specifications to support external variables as described in §II. Additionally, the specification are being scaled up to describe processing across several clusters that follow different rules—for example, the two clusters in Fig. 1 might be placed in different datacenters that have different power and cooling costs.

Acknowledgment

This material is based upon work supported by a gift from Google and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-19-C-0106. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of either organization.

References

[1] Illinois Institute of Technology’s Galvin Center for Electricity Innovation. http://iitmicrogrid.net/.

[2] Illinois Institute of Technology’s Ocient Computational Center. https://www.iit.edu/computing/research/research-centers.

[3] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. P4: Programming Protocol-Independent Packet Processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, jul 2014.

[4] William F Clocksin and Christopher S Mellish. *Programming in Prolog*. Springer Science & Business Media, 2003.

[5] Hyojoon Kim, Xiaoqi Chen, Jack Brassil, and Jennifer Rexford. Experience-Driven Research on Programmable Networks. *SIGCOMM Comput. Commun. Rev.*, 51(1):10–17, March 2021.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[7] Nik Sultana. Leveraging In-Network Application Awareness. In *NAI’21: Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration, Virtual Event, USA, August 27, 2021*, pages 63–67. ACM, 2021.

[8] Nik Sultana. Experiment Planning for Heterogeneous Programmable Networks, June 2022. International Workshop on Test and Evaluation of Programmable Networks.

[9] Nik Sultana, John Sonchack, Hans Giesen, Isaac Pedisich, Zhaoyang Han, Nishanth Shyamkumar, Shivani Burad, André DeHon, and Boon Thau Loo. Flightplan: Dataplane Disaggregation and Placement for P4 Programs. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 571–592. USENIX Association, April 2021.