Research Statement

Ashish Vulimiri

My research addresses performance bottlenecks in a range of large scale networked and distributed applications. In my work I identify performance issues in major applications, propose mitigating techniques, build systems incorporating my ideas, and evaluate them at large scales to demonstrate practical benefit. When possible I also leverage theoretical analysis to gain insight into the fundamental tradeoffs underlying the system design.

Research overview

My recent work pursues performance improvements in two key application scenarios: reducing the bandwidth cost of operating Big Data analytics pipelines in global scale organizations, and improving the latency experience in large scale end-user facing applications.

Geo-distributed Big Data analytics. Large organizations today have a planetary scale footprint, and collect massive amounts of data each day from users at several data centers around the world. Analyzing all this geographically dispersed data as a whole is necessary to expose business-critical insight. Current solutions operate by copying all the data to a single central location where the analytics is then processed, but this approach consumes a substantial amount of scarce, expensive cross-data center bandwidth, and is at odds with rising regulatory concerns over data sovereignty.

We instead built WANalytics [2], a system that orchestrates distributed query execution on data partitioned across multiple data centers. WANalytics targets the relational model and supports the full array of SQL operators, including joins and user-defined functions, over global data. We optimize query execution to minimize bandwidth cost while handling provided sovereignty and fault-tolerance constraints. Our techniques are inspired by revisiting traditional database problems from a networking perspective. For instance, our query plan optimizer fuses classical database query planning techniques with the integer-programming based task scheduling approach common in distributed execution engines; and we eliminate redundant data transfers across repeated subqueries via a caching mechanism inspired by filesystem deduplication techniques.

We built a Hadoop-stack prototype of WANalytics, a drop-in replacement for the Apache Hive single-data center SQL analytics framework that handles data partitioned across multiple data centers. In a large scale deployment, our prototype achieved a 257× reduction in bandwidth cost compared to the centralized approach on a Microsoft production workload, and up to a 360× cost reduction on a range of synthetic benchmarks, including TPC-CH, TPC-DS and YCSB.

More recently, we have been working on extending WANalytics to an even more general computational model: directed acyclic task graphs in which each node is an arbitrary computation. While this problem is significantly harder, our initial results have been encouraging [3]. In experiments with several non-relational benchmarks, we found that we could achieve a 330× reduction in bandwidth cost compared to the centralized approach.

Reducing end-user latency. Low latency is critical for ensuring user satisfaction in interactive networked applications. But while we know how to scale systems to increase capacity, reducing latency — especially the tail of the latency distribution — is much harder. We consider a simple, general purpose technique that trades off added capacity utilization to achieve a latency reduction: deliberately initiate redundant copies of latency-sensitive operations, for instance by contacting multiple servers in parallel for the same request, and take the first copy to complete.
Redundancy has been explored in some past systems, but these tend to be the exception, not the rule — the use of redundancy is generally avoided because of a fear of the overhead it adds. We take a two-pronged approach to show that this fear is ill-founded in many cases [4, 6]. First, we show empirically that redundancy achieves a significant performance benefit in several applications: DNS resolution, the TCP handshake, multipath overlay routing, key-value stores, and data center network transmission. Next, we analytically study the effect of redundancy in abstract queueing-theoretic and economic models, and show that it should yield a significant improvement in large classes of systems. Our results characterize when redundancy is and isn’t beneficial overall, and demonstrate a large class of systems in which it is a significant net positive, thus suggesting that the technique should be used much more widely than it currently is.

Our results with DNS redundancy are particularly intriguing: we find that replicating DNS queries to multiple publicly advertised DNS servers can improve raw DNS performance by 24 – 62%, and reduce browser page load times by 5 – 15% in both the mean and the tail [5]. We are currently discussing incorporating redundant DNS requests directly into the browser with browser developers.

Future work

In the future I intend to continue working on improving the performance and scalability of Big Data systems on two fronts: by developing more sophisticated solutions for Big Data analytics across data centers, as well as by improving system performance within data centers.

Big Data solutions across data centers. Rising data volumes and the scarcity of trans-oceanic bandwidth are rendering the centralized approach to global scale Big Data analytics, the norm today, increasingly unsustainable. My work on WANalytics provides a geo-distributed analytics solution that scales much better, but the optimizations we have implemented all target applications that are solely concerned with bandwidth cost. This is only one of many possible combinations of requirements in the geo-distributed analytics space. I am interested in addressing the several other concerns that can arise here, such as:

- Minimizing analytics latency, by extending our optimization model to take load levels and scheduling considerations at each data center into account.
- Supporting strong consistency requirements.
- Providing well-defined privacy guarantees, by incorporating differential privacy mechanisms or privacy preserving computation techniques.
- Incorporating awareness of computational costs. This is particularly important for multi-tenancy, allowing multiple global scale organizations to process analytics using the same physical infrastructure.

Improving performance within data centers. Inefficiencies remain to be addressed at various levels in the application stack in data centers. I am particularly interested in tackling two problems:

(a) Leveraging application-level knowledge. In order to achieve high performance, task schedulers in distributed execution frameworks must incorporate users’ knowledge of the resources that their applications need. Recent work has proposed partial solutions that, for instance, address per-task resource requirements without accounting for dataflow dependencies between tasks; or provide for detailed specifications of networking requirements without incorporating other resources into the model. The problem of designing a scheduling approach that supports and makes full use of a complete specification of users’ resource requirements remains unsolved. I believe my experience with characterizing resource requirements and tradeoffs in a variety of contexts [1, 2, 4] leaves me well-placed to address this important open challenge.

(b) Making full use of network capacity. Data center networks provision multiple paths between end-points, with time-varying capacity and congestion profiles. How do we make efficient use of all this available capacity to achieve low latency transmission? My work on redundancy provides a glimpse of what is possible here: by blindly replicating packets across multiple paths, we were able to reduce flow completion times in a
data center network by up to 40% in the mean and up to 90% in the tail. I am interesting in developing a systematic solution allowing individual applications to fully utilize all the resources available to them, incorporating online machine learning techniques to achieve adaptive path selection, and intelligent use of redundancy and erasure coding to mitigate performance uncertainty on individual paths.

References


