Rising Star Award co-Winner

I am humbled and honored to be selected as a co-recipient of the IEEE TCDE Rising Star award this year, and I am grateful to the TCDE community for this recognition.

Designing a database system for warehouse-scale computers

My research is motivated by the increasing use of warehouse-scale computers to analyze massive datasets quickly. Current examples include a scientist who post-processes simulation results on a high-performance computer or an enterprise that periodically rents 100,000 CPU cores in the cloud for $500 per hour to sift through IoT data.

These scenarios pose two challenges for database systems. The first challenge is the lack of *interoperability* with other analytical tools. Massive datasets often consist of images (arrays) that are stored in file formats like JPEG, PNG, FITS and HDF5. To analyze these datasets, users write code that directly manipulates data in files and leverages domain-specific libraries or deep learning frameworks such as TensorFlow. The second challenge is *scalability*, as foundational data processing operations do not scale to the unprecedented concurrency of warehouse-scale computers at the compute, networking and storage levels. Examples include data shuffling that triggers an inherently unscalable all-to-all communication pattern; hash-based joins that partition the inputs and do not allow communication and computation to fully overlap; parallel aggregations that are oblivious to congested links in non-uniform network topologies.

To address the interoperability concern, we are developing ArrayBridge [code.osu.edu/arraybridge], an open-source I/O library that brings advanced data management capabilities under an array-centric file format interface. ArrayBridge currently allows scientists to use SciDB, TensorFlow and HDF5-based code in the same analysis pipeline without converting between file formats. Under the hood, ArrayBridge controls I/O concurrency, data materialization and data placement to fully utilize warehouse-scale parallel file systems. ArrayBridge does not modify the array file format API, so it remains backwards compatibility with legacy applications. In other words, with ArrayBridge, scientists can simultaneously benefit from the I/O optimizations of a database system and directly manipulate data through the existing API when they need to.

To address the scalability concern, we are developing algorithms that use high-performance networks more efficiently. One example is an RDMA-aware data shuffling algorithm that directly converses with the network adapter in InfiniBand verbs and shuffles data up to $4\times$ faster than the RDMA-aware MVAPICH implementation of MPI. This impressive performance gain is achieved by carefully designing its data structures for concurrent operations in a multi-threaded setting and by using a connectionless, datagram-based transport layer that scales better but requires flow control and error detection in software. Another example is a parallel aggregation algorithm that predicts bottlenecks caused by link congestion and then leverages partition similarity to transfer less data over the congested link. We have prototyped both algorithms in Pythia [code.osu.edu/pythia], an open-source distributed query execution engine.

Looking ahead, the modern datacenter is becoming more heterogeneous with the introduction of FPGA nodes, many-core nodes, large memory nodes, GPU nodes and nodes with local NVMe-based storage. How to harness heterogeneous hardware for data analysis is an open research problem.

Spyros Blanas
Ohio State
Columbus, OH