Project Caerus Manifesto

Motivation

Big data and AI applications are proliferating rapidly, thanks to advances in new technologies such as the Internet of Things, 5G, and machine learning. These applications are compute-, IO-, and memory-intensive, imposing significant performance and scalability challenges on the underlying compute and storage systems. Typically, distributed data-parallel compute engines (e.g., Spark and TensorFlow) and distributed storage systems (e.g., S3 and Ceph) are developed and managed independently. The de-coupling of compute and storage has made it difficult to perform end-to-end optimization or to fully exploit hardware resource capacity available in the overall system. Furthermore, compute engines interact with the storage substrate only through standardized APIs that constrain the information exchanged between compute and storage and render their coordination inefficient and expensive, relying on application developers to work out their own optimization strategy.

Objective

Caerus is an initiative focused on bridging the gap between distributed compute and distributed storage platforms commonly used for big data and AI applications. Caerus aims to create a new open ecosystem that allows compute and storage platforms from different sources to operate in a concerted fashion to substantially improve application performance, resource utilization, and application developer productivity.

Key Principles

Caerus develops technologies that optimize end-to-end performance through well balanced and cost-effective tradeoffs between compute and storage. These could include, for example, exploiting local storage in the data processing system to reduce the amount of I/O's incurred on the network and external storage server, exploiting the compute capacity within the storage system to reduce demand on the data processing engine, and substituting less costly operations for more expensive ones.

In order to facilitate more meaningful and more fruitful cooperation between compute and storage systems, new APIs should be developed that extend conventional I/O requests and bridge the semantic gap between different systems. Information such as data processing primitives, workload patterns, and data profile could all be exchanged through the new APIs. The new APIs should be part of a standardized framework for compute-storage coordination that accommodates a wide variety of technologies and providers.

It is important to abstract away from development complexity related to data management, execution planning, and resource allocation. Collaboration between compute and storage involves a lot of low-level decisions and fine-grained tuning in order to yield best performance results. Automating decision-making and tuning on behalf of application developers and operators is crucial for the adoption and success of the Caerus vision.

Technical Directions

Here are a few technical directions Project Caerus will initially focus on.

Near Data Processing offloads part of data processing from a compute engine to a storage system. Project Caerus plans to develop a general NDP framework that allows a wide variety of query operations to be pushed down and that can be integrated with a broad range of data processing engines and storage systems. Pushed-down query operations, when properly planned, translate into reduced network I/O and storage I/O and result in major savings of CPU and memory resources on the compute side.

Semantic Caching intelligently caches data and metadata on the compute side, by leveraging semantic information on the executed workload and stored content. While cached data and intermediate results can be used in future queries directly, cached metadata can be leveraged to optimize data organization in storage and query execution plans. Semantic caching promises to lessen the burden on storage and substantially improve the performance of the compute engine.

Smart Shuffle optimizes a critical performance bottleneck in MapReduce workloads. Traditionally, data shuffling causes a lot of I/O burden for the storage side and renders the compute side ineffective, because allocated compute resources are underutilized waiting for I/O requests to complete. Smart Shuffle aims to produce new shuffle protocols that reduce I/O overhead and excessive CPU cycles.

Semantic Cache, Smart Shuffle, and NDP Processing work in sync with the Caerus end-to-end approach, make use of the extended storage API, and emphasize on automated optimization decisions via comprehensive analytical modeling of performance, overhead, fairness, priority, and economics.

Conclusion

Project Caerus is an open collaboration project with open design, open APIs, open communication, and open governance. It welcomes contributions from the broad community and in all forms, such as research, design, implementation, integration, testing, operation, documentation, education, financing, training, facilitating, and evangelizing. It looks forward to close partnership with both academia and industry to impact both the state of the art and the state of the practice for big data and AI infrastructure, through a strong and open ecosystem.