Bringing the Power of Object Stores to SAP IQ

Güneş Aluç

SAP Labs, Canada
gunes.aluc@sap.com

1 Introduction

In this presentation, we describe our journey of transforming SAP IQ into a relational database management system (RDBMS) that utilizes cheap, elastically scalable object stores in the cloud [3, 4]. SAP IQ is a three-decade old, disk-based, columnar RDBMS that is optimized for complex online analytical processing (OLAP) workloads. Traditionally, SAP IQ has been designed to operate on shared storage devices with *strong consistency guarantees* (e.g., high-caliber storage area network devices). Therefore, deploying SAP IQ on the cloud, as is, would have meant utilizing storage solutions that provide a POSIX compliant file interface and strong consistency guarantees, but at a much higher monetary cost. These costs can accumulate easily to diminish the economies of scale that one would expect on the cloud, which can be undesirable. Instead, we have enhanced the design of SAP IQ to operate on cloud object stores such as AWS S3 [1] and Azure Blob Storage [2]. Object stores rely on a weaker consistency model, and potentially have higher latency; however, because of these design trade-offs, they are able to offer (i) better pricing, (ii) enhanced durability, (iii) improved elasticity, and (iv) higher throughput. By enhancing SAP IQ to operate under these design trade-offs, we have unlocked many of the opportunities offered by object stores. Experiments using the TPC-H benchmark demonstrate that we can gain an order of magnitude reduction in data-at-rest storage costs while improving query and load performance.

2 Contributions

When developing the cloud-native version of SAP IQ, we have decided to exploit the strengths of the product as much as possible and avoid reinventing the wheel. In particular, SAP IQ benefits from three decades of research and development when it comes to techniques such as data compression, partitioning, indexing, prefetching and loading that we wanted to exploit. Consequently, in this presentation, we discuss the following aspects of our design:

1. SAP IQ makes a clear distinction between the logical and the physical representation of pages in the system; therefore, we directly map logical pages to objects in object stores;
2. We enforce a "never write an object twice" policy in the transaction and buffer managers to handle the weaker consistency model used in object stores;
3. We implement techniques for efficiently allocating object keys in a multi-node setting and discuss the lessons we have learnt in working with object stores; and lastly
4. We discuss the design considerations of the Extended Cache Manager (ECM), that acts as a read/write cache between the existing buffer manager and the object store [4].

References

[1] Amazon Simple Storage Service (S3). `http://aws.amazon.com/s3/`.

[2] Azure Blob Storage. `https://azure.microsoft.com/en-us/services/storage/blobs/`.

[3] M. Abouzour, G. Aluç, I. T. Bowman, X. Deng, N. Marathe, S. Ranadive, M. Sharique, and J. C. Smirnios. Bringing cloud-native storage to SAP IQ. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, pages 2410–2422. ACM, 2021.

[4] S. Shedge, N. Sharma, A. Agarwal, M. Abouzour, and G. Aluç. An extended SSD-based cache for efficient object store access in SAP IQ. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pages 1861–1873. IEEE, 2022.