Title of the project: Object Storage for CMS in the HL-LHC era

Name: Nick Smith

Note: items 1-3 of this report are taken from the following proceedings that provide a good summary of the outcome of this project: https://doi.org/10.1051/epjconf/202429501003

1. Starting with a brief overview, explain the goal of the project, the problem you aimed to address in your R&D project, and the method/process. What was the expected impact on the HL-LHC CMS S&C operations?

> The HL-LHC poses a significant challenge for the CMS experiment's data management model. A combination of higher pile-up and finer detector granularity will result in each collision event of data or simulation requiring considerably larger byte-storage. The overall event rate to storage, currently at approximately 1 kHz, is also anticipated to increase to at least 7.5 kHz. In the early years of Run 4, several new subdetector components will come online, and their low-level data will need to be validated and synthesized into high-level reduced data suitable for analysis. This process will be challenging if the data products pertaining to those detector elements are not easily accessible. Future innovative analysis may necessitate different data reduction strategies. Given these considerations, the amount of data that is regularly accessed by end-user analysts is expected to grow significantly in the coming years.
>
> An inherent limitation in the current CMS data management model is the organization of data around files. File-based organization fundamentally forces certain data products, or collections of related fields (data columns) pertaining to an event to be grouped together with the same data access quality of service (QoS). In the current workflow model, large-scale tasks transform datasets from one set of columns to another more compact set, to afford keeping the latter available with high QoS (i.e., on magnetic disk at several sites.) It is not currently possible to have different subsets of data columns kept at high QoS for different datasets at granularities beyond the data tier, a statically defined set of columns. A more flexible data model that keeps only the needed data columns available at high QoS may significantly reduce storage volume risk.

2. What techniques did you use to address the problem? If more than one avenue were taken, what were the other techniques? And why were the chosen techniques expected to perform better than the alternative or baseline methods?

> Object stores, which operate on a key-value principle, allow for highly-granular access to information via efficient metadata lookup. Object stores are widespread in industry and are near-universal in cloud storage with the Amazon S3 API emerging as a de facto access protocol across them. A central object store can also provide the backbone of a content delivery network (CDN) for end-users and allow for analysis without need for an analysis NTuple format. To concretely explore the possibilities of object storage for

event data in CMS, we developed an object data format, a test cluster using Ceph object store technology, and a S3 input and output module in a prototype event processing framework similar to CMSSW. Using these components, we evaluated the storage and read/write performance against a baseline of current MiniAOD event size and processing rates. Scaling behavior is probed both in single- and multi-process settings, to understand threading efficiency and client load limits of the cluster, respectively.

3. Explain the outcome of the project, including alternative paths and their estimated impact as well. Was the outcome as expected as described in 3.? If not, why not?

> We found that the data volume is in line with expectations (a modest increase in kB/event due to limitations in ROOT serialization to byte blobs compared to TTree) and service scaling is promising, with one Ceph RadosGW node serving 350-400 parallel clients before performance saturation.

4. Was there a particularly difficult obstacle you encountered during this project? How did you navigate through it, and were there any pivotal breakthrough moments? What are some gained knowledge that others who may tackle similar problems in the future would want to know about beforehand?

> A very difficult obstacle was to properly design an asynchronous http I/O request interface within the Intel TBB library, so that it would not block any thread from performing CPU-bound work. Close work with Chris Jones allowed us to come to a satisfactory solution that may be of use in other contexts, such as SONIC.

5. What do you recommend for the future if the problem were to be tackled again, or to make improvements to your work?

> Closer integration with the ROOT team and leveraging RNTuple would help significantly. It was not possible to do so for this project due to mismatched time schedules.

6. What kinds of skills did you learn through the project? Do you think this experience has helped you understand how to develop and lead a research project?

> I learned quite a bit about kubernetes, Ceph, S3 protocol, Intel TBB and lock-free programming in C++, and ROOT low-level I/O through this project, all of which I expect to be useful in the future.

7. How was your work presented to the wider CMS community? Could you provide a list of meetings, or conferences where the work was presented?

> A list of presentations is available at
> https://uscms-software-and-computing.github.io/postdocs/nsmith-.html