

Table of Contents

CMSSW I/O with the 2-cache schema.....	1
--	---

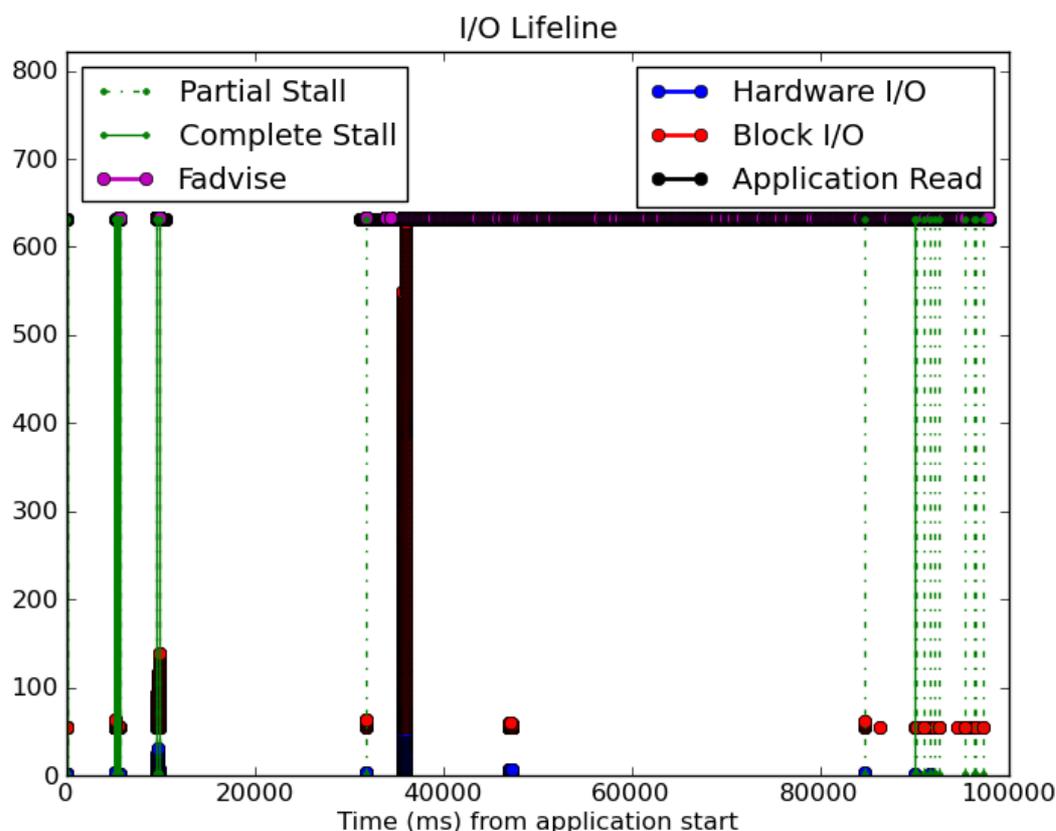
CMSSW I/O with the 2-cache schema

This page discusses CMSSW I/O under the following conditions:

- Running PAT tuple creation.
- With the CMS I/O patches found here. This enables the 2-cache scheme
- cacheSize set to 20MB.
- readHint mode set to "storage-only".
- Reading out 300 events.

Below is a few graphs showing the I/O activity using the BlockTrace utility:

- I/O activity on analyzed file over the whole run of the application:



There are four main bands of activity - first, when ROOT opens the file. Second, when ROOT starts opening and processing the TTree itself (figuring out what it needs to deserialize, event metadata). Third is the initial fill of the "raw tree", which caches all branches. After this, there's a long pause in I/O when the conditions data is loaded. Finally, there is the run through the file itself. The first 20 events are read using the raw tree. Then, the TTreeCache buffer is then filled for the specialized tree. Afterward, you see (almost) no more stalls for the rest of the file. The partial stalls later on in the job are due to a branch being used which is not cached.

- Notice that the disk-level parallelism is high. However, the job does stall in the posix_fadvise call as in the other asynchronous I/O like the other storage cache jobs. We will (in the long run) be working to avoid the disk-level blocking.

This topic: Main > CmsIoCmssw2Cache

Topic revision: r2 - 2010-05-13 - BrianBockelman



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)