Table of Contents

Instrumenting CMSSW I/O

This page discusses how to instrument CMSSW I/O at the Linux kernel in order to better understand how the application interacts with the block layer. The goals are:

1. Provide insights into how CMSSW (and hence ROOT) I/O's usage of various system calls results in disk activity.
2. Identify potential performance bottlenecks.
3. Experiment with potential solutions.

Pages in this guide

BlockTrace, a set of scripts to record and visualize Linux kernel I/O.

1. Using BlockTrace and SystemTap on your system. How to run BlockTrace on your systems in order to repeat the measurements I have made yourself.
2. BlockTrace graphs. How to understand and read the BlockTrace graphs I will be showing.
3. User-level SystemTap tracing. Correlating between user-level events and kernel-level events.

Findings for CMSSW

1. CMSSW with no cache. This is the out-of-the box behavior of CMSSW.
2. CMSSW with readHint=application-only. This turns on TTreeCache, but does I/O in synchronous mode.
3. CMSSW with readHint=storage-only. This turns on TTreeCache with I/O in asynchronous mode.
4. CMSSW with the 2-cache scheme. This turns on the 2-cache scheme, as described by CmsIOWork2.

This topic: Main > CmsIOInstrumenting
Topic revision: r5 - 2010-05-13 - BrianBockelman