HDFS Improvements

James Thomas

DataNode block layout on disk

- Each block is a single file
- High memory cost to keep track of full path of each block
- 240 TB DataNodes in a few years -- huge number of blocks but can’t increase RAM much due to GC issues

DataNode block layout on disk

- Now we determine directory for block based on the block’s ID

$$\texttt{0x aa bb cc dd ee ff 00 }\underbrace{\texttt{11}}\;\underbrace{\texttt{22}}\texttt{ 33}$$

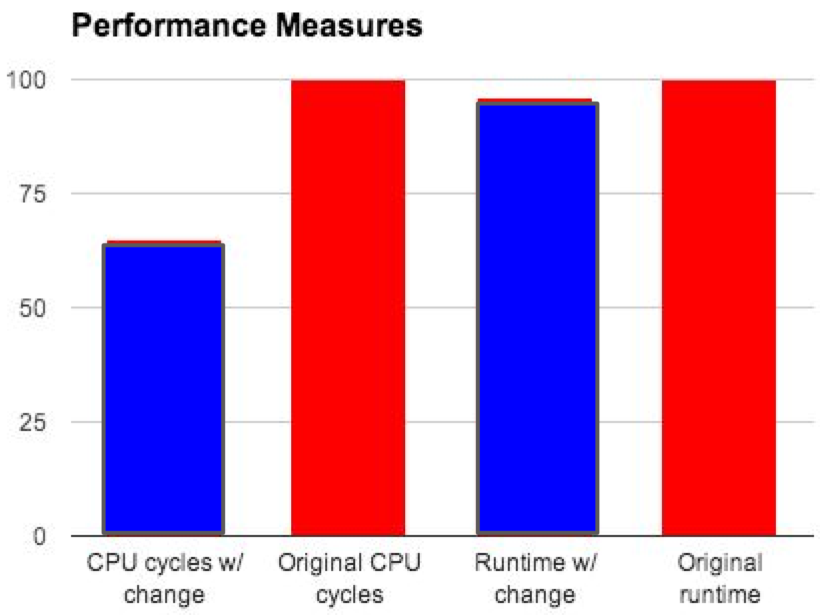
```
data / 11 / 22 / block.file
```

- Reduced DataNode **memory consumption by around 15%**

Native checksumming on write path

- Checksums computed on client and verified on datanodes
- Modify code paths to use C checksumming implementation that saturates processor pipeline

Native checksumming on write path



inotify in HDFS

- Linux inotify allows clients to watch directories and receive notifications of changes
- The same functionality is useful in HDFS -- search systems don’t have to scan the whole directory tree for changes, Impala can do automatic ETL

inotify in HDFS

- Events for file creation, append, close, deletion, rename, and metadata updates
- Immediately useful to Cloudera Navigator, which currently has to read the HDFS edit log directly using a private API

JIRAs

- DataNode layout -- HDFS-6482 (committed)
- Native checksumming -- HDFS-3528 (under review)
- inotify -- HDFS-6634 (in progress)