

日曜研究室

技術的な観点から日常を綴ります

[xv6 #55] Chapter 5 – File system – Buffer cache Layer

テキストの64ページ

本文

バッファキャッシュには2つの仕事がある。

- (1) あるブロックのメモリ上のコピーが一つしかない状態と、そのコピーを一度に一つのカーネルスレッドだけが利用出来るということを保証するために、ディスクブロックへのアクセスを同期する。
 - (2) 遅いディスクから再読込しないようにするために、よく使われるブロックをキャッシュする。
- この仕事を実装したコードはbio.cにある。

バッファキャッシュの主なインターフェイスは、`bread`と`bwrite`から成る。

前者は、ブロックのコピーを含むバッファ（メモリ上で読んだり変更したり出来る）を得、後者は、変更されたバッファをディスク上の適切なブロックに書き込む。

カーネルスレッドは、バッファを使い終わったら、`brelease`を呼んでバッファを解放しなければならない。

バッファキャッシュは、多くても一つのカーネルスレッドだけが、あるブロックのバッファへの参照を持てるようにすることで、各ブロックへのアクセスを同期する。

あるカーネルスレッドが、あるバッファへの参照を得、なおかつまだそれを解放してない場合、他のスレッドによる同じブロックに対する`bread`の呼び出しは待たされるだろう。

ファイルシステムのより高い層は、それらのインバリアントを維持しやすくするために、バッファキャッシュのブロックに対する同期化機構を頼りにしている。

バッファキャッシュの、ディスクのブロックを保持するためバッファの数は固定であり、それは、キャッシュに載ってないブロックに対するアクセスがあった場合、バッファキャッシュが現在保持している他のブロックをいくつかリサイクルしなければならない、ということを意味する。

バッファキャッシュは、使われてから一番時間が経っているバッファを新しいブロックのためにリサイクルする。これは、使われてから一番時間が経っているバッファが、すぐ再び使用される可能性が

一番低いという仮定に基づく。

感想

ファイルシステムが一番下の層であるバッファキャッシュについての概要です。

次の節がバッファキャッシュのコードの説明なので、ここではまだコードは載せません。

この層がxv6のファイルシステムの屋台骨ですね。

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/3/28 水曜日 [<http://peta.okechan.net/blog/archives/1587>] |
