情報爆発時代のデータベースの新たな構成

A Proposal of Database Architecture Facing Info-Plosion

SSDを対象としたKey-Value Storeのデータ構造に関する研究

Data Structure for the Key-Value Store applied to SSD

山田淳二（東京大学情報理工学系研究科 大学院生）　浅野正一郎（NII）

どんな研究？

「Bigtable」はGoogle検索に使用されている技術で，KeyとValueのペアからなるシンプルなデータモデルによるデータストアで，「KVS(Key Value Store)」とも呼ばれています．

一方，NAND Flashメモリを利用したSSD(Solid State Drive)はHDDと比べ桁違いの高速アクセス性能を持つことから，KVSへの活用が検討されています．

しかしNAND Flashメモリは上書きすることができず，消去した後に書込みをしなければなりません．また消去単位は書込み単位より大きいため，Key-Valueの書込みの効率化と探索時間の高速化を同時に可能とする方式の考案が必要とされています．本研究は，このような技術課題を解決し，情報爆発時代の大規模データベースへSSDを利用するためのものです．

具体的には？

先行した研究ではKeyとValueの対を到着順にシーケンシャルにSSDに書き込み，Keyと記録位置の対応を主記憶上で管理する方式が提案されてきました．B. Debnath等により提案されたFlashStoreでは，主既往区の節約のためにKeyをハッシュ関数により2B（バイト）に縮約し，またSSD上の記録位置を4Bで記録する．これにより1記録要素の主記憶は6Bとなり，4GBの主記憶で1kBの記録要素の場合715GBのSSDのデータが管理できる．

本研究ではハッシュの代わりにBloom Filterを利用することで，主記憶のメモリ効率を高めることを提案しています．

ここでBloom Filterとは，「ある要素が集合に属しているかを判定する」が「実際に集合に属していない要素を“属している”と誤判定（擬陽性）することがあり得る」もので，集合への帰属を判定する機能を持つデータ構造に使用されるものです．

NAND Flash メモリによる Key Value Store を実用化することで大規模な情報検索の高速化が実現できます

ポイント：1

NAND Flash メモリは DRAM に比べて1ビット当たりの面積が小さく，小型でも大容量のメモリが造れます．これによるSSDはシーケンシャル・ライトが高速な特性を併せ持ちますので，SSD上にシーケンシャルにデータを書き込み，Keyと書き込み位置の対応を主記憶上のインデックスで保持するKVSが提案されています．しかしSSDへの書込みと消去を繰り返すと誤りが増加します．また，ページ単位に書込みが行われますが，追記はできません．消去単位はブロックと呼ばれ，ページより大きい単位となります．HDDのアクセスとは異なる動作となるために，一般には FTL(Flash Translation Layer)と呼ぶ階層を設けて，書込み回数の管理や誤り訂正を含むインタフェース変換を行います．販売されているSSDでは複数のFTLが実装されていると思われますが，この中でPage-level FTLと呼ばれる方式によるSSDはKVSの実現に適していると考えます．

ポイント：2

SSD上の記録位置を少ないビットで管理することが必要となります．

ハッシュは，ハッシュ値の衝突が充分抑制できるビット数を持たせると主記憶上のインデックスが巨大となります．一方，本件で対象とするBloom Filterは，要素について複数のハッシュ値（図1では4通りのハッシュ値を求めています）を求め，これをBloom Filter のビットに対応付けます．既に提案されている方式では，図2に示すようにKey-Value の対が記憶される領域をブロックに分割し，各ブロックに対応するBloom Filterに，そのブロック内に含まれるKeyを登録することでBloom Filterをインデックスとして用いるものです．Bloom Filter のビット数と擬陽性確率には関係があり，擬陽性時の探索は性能を低下させます．

提案するのは図3に示すBloom Filter Map方式です．ここでは各ブロックに対応するBloom Filterが配列となっており，与えられたKeyに対して得られたハッシュ値に対応する配列を主記憶から取得し，AND操作によりブロックを探索するものです．本提案は探索の高速化が期待でき，従来方式に比べて12.8倍の高速化が実現できることが評価されました．

ポイント：3

その他にも提案しています．

擬陽性の判定の高速化のために，ページ単位に，含まれるKeyの範囲を記憶することを提案しています．またハッシュでは要素数が多くなるとハッシュ値の衝突が無視できなくなりますが，要素数が少ない時点まではハッシュ値と記憶位置を単純に対応させ，要素数が増えるに従ってBloom Filter Mapを動作させる等の方式が有効であることも確認しています．

尚，一連の評価はPage-level FTLと呼ばれるFTLを採用しているSSDを用いて実データにより実施していますが，Block-level FTLを採用していると思われる製品もあり，この場合は効果が減じることになると思われます．

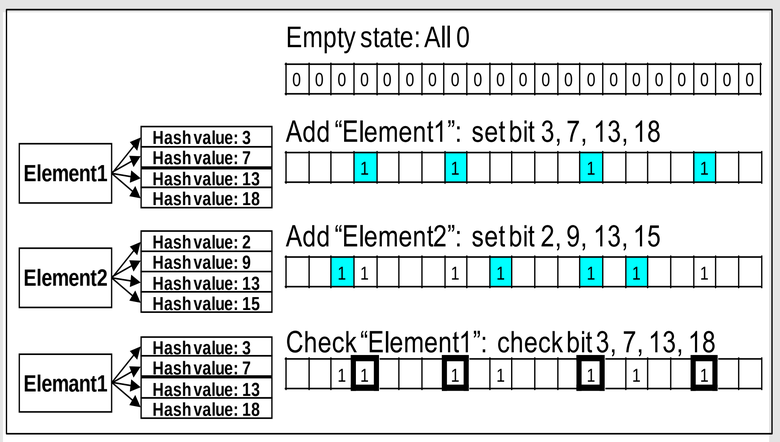
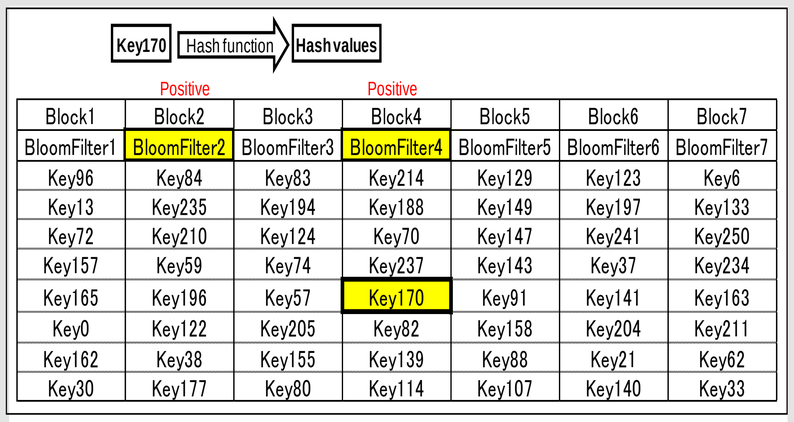


図1　Bloom Filterの動作



Block1	Block2	Block3	Block4	Block5	Block6	Block7
BloomFilter1	BloomFilter2	BloomFilter3	BloomFilter4	BloomFilter5	BloomFilter6	BloomFilter7
Key96	Key84	Key83	Key214	Key129	Key123	Key6
Key13	Key235	Key194	Key188	Key149	Key197	Key133
Key72	Key210	Key124	Key70	Key147	Key241	Key250
Key157	Key59	Key74	Key237	Key143	Key37	Key234
Key165	Key196	Key57	Key170	Key91	Key141	Key163
Key0	Key122	Key205	Key82	Key158	Key204	Key211
Key162	Key38	Key155	Key139	Key88	Key21	Key62
Key30	Key177	Key80	Key114	Key107	Key140	Key33

図2　Bloom Filterによる主記憶上のインデックス

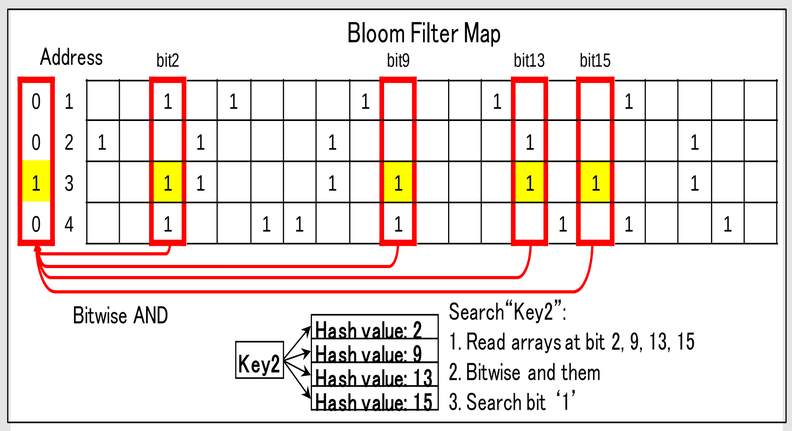


図3　提案するBloom Filter Map方式

NII

連絡先：　浅野正一郎（Shoichiro ASANO）／ 国立情報学研究所 アーキテクチャ科学研究系　教授

TEL：03-4212-2505　　Email：asano@nii.ac.jp