

SSD Problem

Furukawa, Takashi
Graduate School of Mathematics, Kyushu University

<https://hdl.handle.net/2324/1434357>

出版情報 : C0E Lecture Note. 48, pp.101-113, 2013-03-28. 九州大学マス・フォア・インダストリ研究所
バージョン :
権利関係 :



SSD Problem

九州大学大学院数理学府
古川 貴司

- 1) モデルの説明
- 2) アルゴリズム
- 3) 数値実験
- 4) 結果の考察

1)モデルの説明

2)アルゴリズム

3)数値実験

4)結果の考察

Flash SSD (Flash Solid State Drive) はコンピュータに用いられる補助記憶装置の一種である。

HDD と比べるとデータ処理がスムーズに行えたり、衝撃に強いなどの利点があるが、メモリの書き換え可能回数が少ないという欠点もある。

モデルの説明

V	X	V	V	...	V	UW
V	V	V	V	...	X	UW
⋮	⋮	⋮	⋮	⋮	⋮	⋮
V	V	V	V	...	X	UW

ブロック ページ

ページには 1) *valid(V)*: 情報として意味のあるページ,
 2) *invalid(X)*: 情報として意味のないページ,
 3) *unwritten(UW)*: データを保存できるページ
 の状態がある。

ブロックのうち1つはすべてUWの*free block*が存在する。

1)モデルの説明

2)アルゴリズム

3)数値実験

4)結果の考察

アルゴリズムの詳細

*Page Update*を*free block*以外に
*unwritten page*が存在しなくなるまで行う



交互に行う

*Garbage collection*を行い新しく
*unwritten page*を作る

アルゴリズムの詳細

• *Page Update*

*valid page*をランダムに選択し*unwritten page*に書き込む。
(元の*valid page*は*invalid page*に変わる。)

例

V	X	X	V	V	V	UW
V	V	V	X	X	X	UW
X	V	V	V	UW	X	UW
X	X	V	V	UW	X	UW

アルゴリズムの詳細

• Page Update

*valid page*をランダムに選択し*unwritten page*に書き込む。
(元の*valid page*は*invalid page*に変わる。)

例

V	X	X	V	V	V	UW
V	V	V	X	X	X	UW
X	X	V	V	V	X	UW
X	X	V	V	UW	X	UW

アルゴリズムの詳細

• Garbage collection

free block 以外で*valid page*が最も少ないブロック(複数ある場合はその中からランダムに選択する)をさがし、そのブロックの*valid page*を*free block*に移す。選ばれたブロック全体を*unwritten*の状態にすることで新たな*free block*とする。

例

V	X	X	V	V	V	UW
V	V	V	X	X	X	UW
X	X	V	V	V	X	UW
X	X	V	X	V	X	UW
2	1	3	2	3	1	free

*valid page*の数

アルゴリズムの詳細

・Garbage collection

free block 以外で *valid page* が最も少ないブロック (複数ある場合はその中からランダムに選択する) をさがし、そのブロックの *valid page* を *free block* に移す。選ばれたブロック全体を *unwritten* の状態にすることで新たな *free block* とする。

例

V	UW	X	V	V	V	X
V	UW	V	X	X	X	UW
X	UW	V	V	V	X	UW
X	UW	V	X	V	X	UW
2	free	3	2	3	1	1

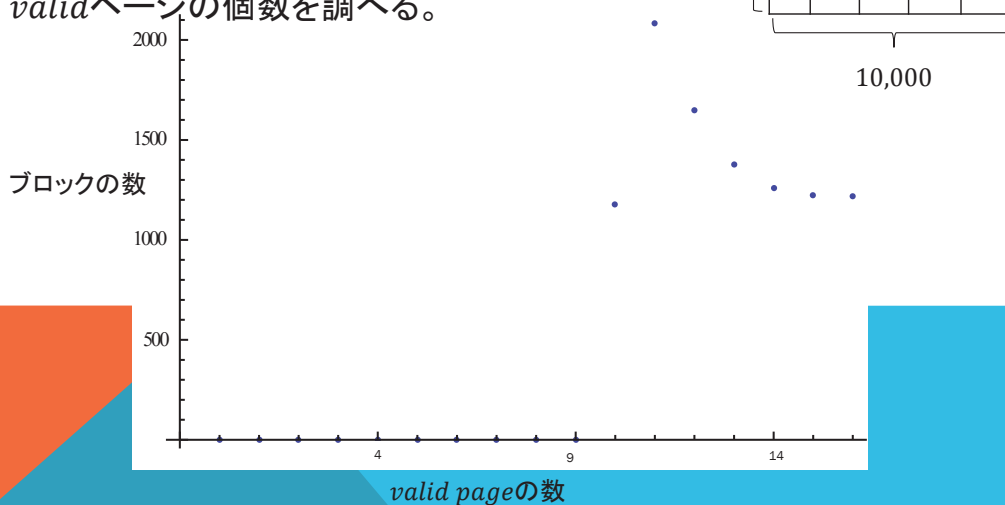
valid page の数

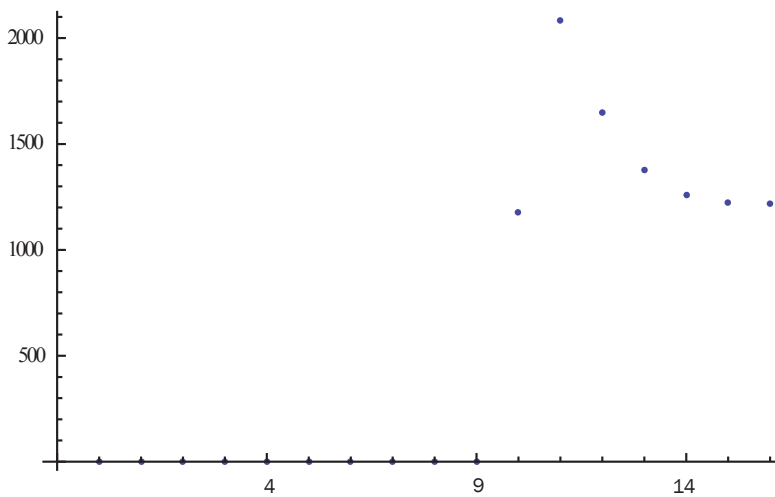
Werner Bux, Ilias Iliadisらによる先行研究(2010)

ブロックの数: 10,000、ページの数: 16

valid ページの数: 128,000

としたとき、定常状態での各ブロックに存在する *valid* ページの個数を調べる。





先行研究では、定常状態ではある c^* が存在し、*Garbage Collection*を行うとき高確率で*valid page*の数が c^* 以下のブロックは選ばれないことが示された。この場合 $c^* = 9$ である。

アルゴリズムの詳細

先行研究では、*Page Update*は一様に行っていた。
本研究では、モデルを一般化するため*Page Update*される確率が異なる*valid* ページが存在するモデルを考えた。

考えるモデル

A	X	B	B	...	A	UW
A	B	B	X	...	X	UW
⋮	⋮	⋮	⋮	⋮	⋮	⋮
X	A	B	A	...	X	UW

*valid page*が A *valid*(A)と B *valid*(B)の2種類ある以外は先行研究と同じである。

アルゴリズムの詳細

•Page Update

$1 / (1 + \mu)$ の確率でA valid を、 $\mu / (1 + \mu)$ の確率でB validを選択し、選ばれた方のvalid pageをランダムに選択しunwritten page に書き込む。
(元のvalid pageは invalid pageに変わる。)

例

X	X	A	B	A	UW
B	A	X	X	X	UW
X	A	B	UW	X	UW
X	B	A	UW	B	UW

アルゴリズムの詳細

•Page Update

$1 / (1 + \mu)$ の確率でA valid を、 $\mu / (1 + \mu)$ の確率でB validを選択し、選ばれた方のvalid pageをランダムに選択しunwritten page に書き込む。
(元のvalid pageは invalid pageに変わる。)

例

X	X	A	B	A	UW
B	A	X	X	X	UW
X	X	B	A	X	UW
X	B	A	UW	B	UW

アルゴリズムの詳細

・Garbage collection

free block 以外で*valid page*が最も少ないブロック(複数ある場合はその中からランダムに選択する)をさがし、そのブロックの*valid page*を*free block*に移す。選ばれたブロック全体を*unwritten*の状態にすることで新たな*free block*とする。

例

X	X	A	B	A	UW
B	A	X	X	X	UW
X	X	B	A	X	UW
X	B	A	A	B	UW
(0,1)	(1,1)	(2,1)	(2,1)	(1,1)	free (A valid pageの数, B valid pageの数)
1	3	2	3	1	free valid pageの数

アルゴリズムの詳細

・Garbage collection

free block 以外で*valid page*が最も少ないブロック(複数ある場合はその中からランダムに選択する)をさがし、そのブロックの*valid page*を*free block*に移す。選ばれたブロック全体を*unwritten*の状態にすることで新たな*free block*とする。

例

UW	X	A	B	A	B
UW	A	X	X	X	UW
UW	X	B	A	X	UW
UW	B	A	A	B	UW
free	(1,1)	(2,1)	(2,1)	(1,1)	(0,1) (A valid pageの数, B valid pageの数)
free	3	2	3	1	1 valid pageの数

1)モデルの説明

2)アルゴリズム

3)数値実験

4)結果の考察

実験内容

1. ブロックの数(b): 5,000

ページの数(c): 16

A valid pageの数(J): $40,000 \times \rho$

B valid pageの数(K): $40,000 \times \rho$

としたときの μ と ρ の値を変えることで

各ブロックのvalid ページの数がどのような変化するか調べる。

2. ブロックの数(b): 5,000

ページの数(c): 16

A valid pageの数(J): $24,000 \times \rho$

B valid pageの数(K): $56,000 \times \rho$

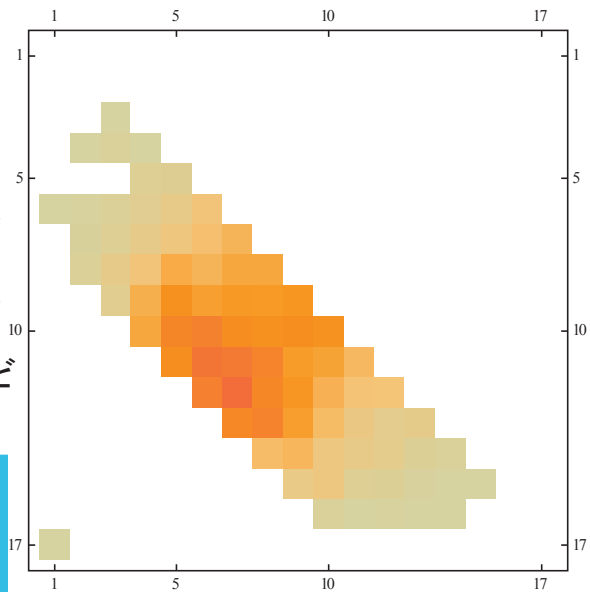
としたときの μ と ρ の値を変えることで

各ブロックのvalid ページの数がどのような変化するか調べる。

得られた結果の一例

1. $\rho = 0.8$,
 $\mu = 1$ の場合

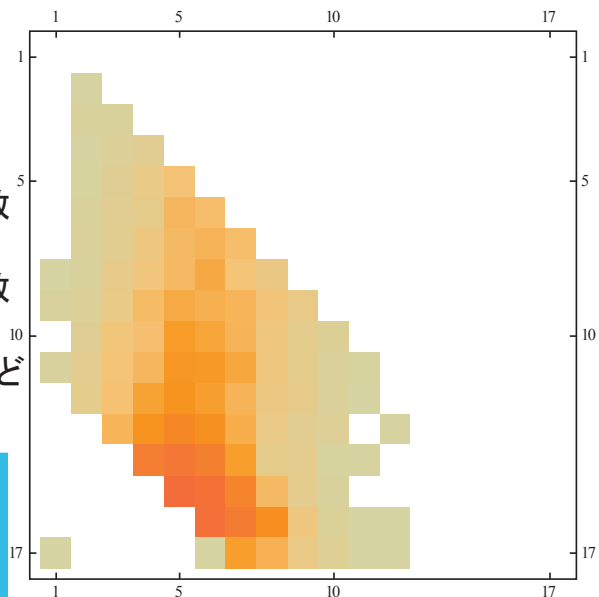
縦軸が α valid pageの数
 (上に行くほど多い)に、
 横軸が β valid pageの数
 (右に行くほど多い)に
 対応し、色が濃いものほど
 数が多い。
 この場合 $c^* = 9$ である。



得られた結果の一例

1. $\rho = 0.6$,
 $\mu = 0.2$ の場合

縦軸が α valid pageの数
 (上に行くほど多い)に、
 横軸が β valid pageの数
 (右に行くほど多い)に
 対応し、色が濃いものほど
 数が多い。
 この場合 $c^* = 5$ である。

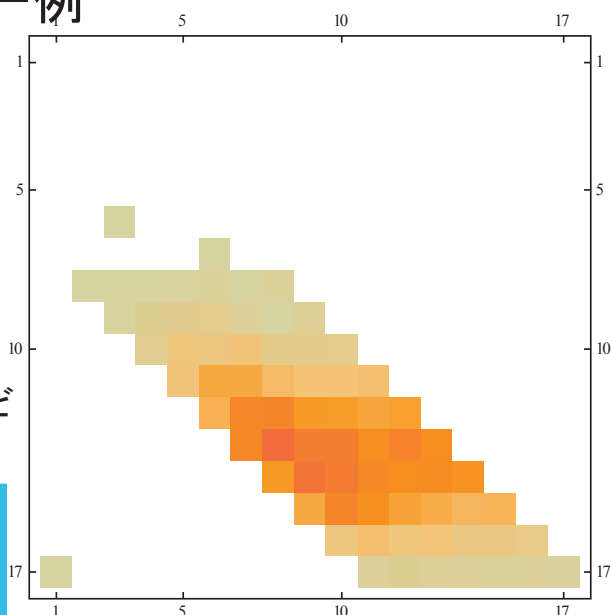


得られた結果の一例

2. $\rho = 0.8$,
 $\mu = 5$ の場合

縦軸が α valid pageの数
 (上に行くほど多い)に、
 横軸が β valid pageの数
 (右に行くほど多い)に
 対応し、色が濃いものほど
 数が多い。

この場合 $c^* = 9$ である。



得られた結果のまとめ

μ と ρ を変化させた時の c^* の値を以下の表にまとめた

1.

$\rho \setminus \mu \times J/K$	0.1	0.2	0.5	1	2	5	10
0.6	5	5	4	4	4	5	5
0.7	7	7	6	6	6	7	7
0.8	9	9	9	9	9	9	9
0.9	12	12	12	12	12	12	12

2.

$\rho \setminus \mu \times J/K$	0.1	0.2	0.5	1	2	5	10
0.6	5	5	4	4	4	4	5
0.7	7	7	6	6	6	7	7
0.8	9	9	9	9	9	9	9
0.9	12	12	12	12	12	12	12

1)モデルの説明

2)アルゴリズム

3)数値実験

4)結果の考察



結果の考察

- ・ c^* の値は μ の値(どちらの *valid page* が選ばれやすいか)にはあまり依存していない。
 - ・ c^* の値は ρ の値(どのくらいブロックを使用しているか)に依存している。
 - ・ c^* の値は各 *valid page* の個数には依存していない。
- 