SBFAST: A Subblock-Based Flash Translation Layer Using Fully-Associative Sector Translation

Design and Simulation Results

Zheng-Yu YANG

July 23 2013

SSD Erase-before-write issue

Incoming I/O request
W(16 17 18 19 20)

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2		26		
3	15	27		
4	16			
5	17			53
6	18			54
7	19			55
8				56
9				
10	22		46	58
11	23		47	59

EraseBlk=2ms
ReadPage=25us
WritePage=200us

Re-Write

Step2 Assemble
Update mapTable

DataBlk93
12
13
15
16
17
18
19
20
22
23

1. BAST

Data Block

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2		26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8			44	56
9			45	
10	22		46	58
11	23		47	59

Cache !

Log Block
(Block-Associated)

LogBlk0	LogBlk1	LogBlk2	LogBlk3	LogBlk4
16	5	36	58	25
17	9	37	59	26
18	5	38	60	25
19	4	39		26
20	1	40		26
20	2	41		
20	3	42		
21	4	43		
16	5	44		
14		45		
16		46		
17		47		

1. BAST

Full Merge

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	
10	22		46	58
11	23		47	59

LogBlk0	LogBlk1	LogBlk2	LogBlk3	LogBlk4
16	5	36	58	25
17	9	37	59	26
18	5	38	60	25
19	4	39		26
18	1	40		26
19	2	41		
22	3	42		
23	4	43		
16	5	44		
14		45		
16		46		
17		47		

1. BAST – Full Merge

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	
10	22		46	58
11	23		47	59

Step3 Recycle DataBlk
(2 Erases)
Assign FreeLogBlk

LogBlk0
16
17
18
19
18
19
22
20
16
14
16
17

Step2 Assemble
Update mapTable

DataBlk93
12
13
14
15
16
17
18
19
20
21
22
23

Step1 Get FreeBlk

1. BAST – Switch Merge

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	
10	22		46	58
11	23		47	59

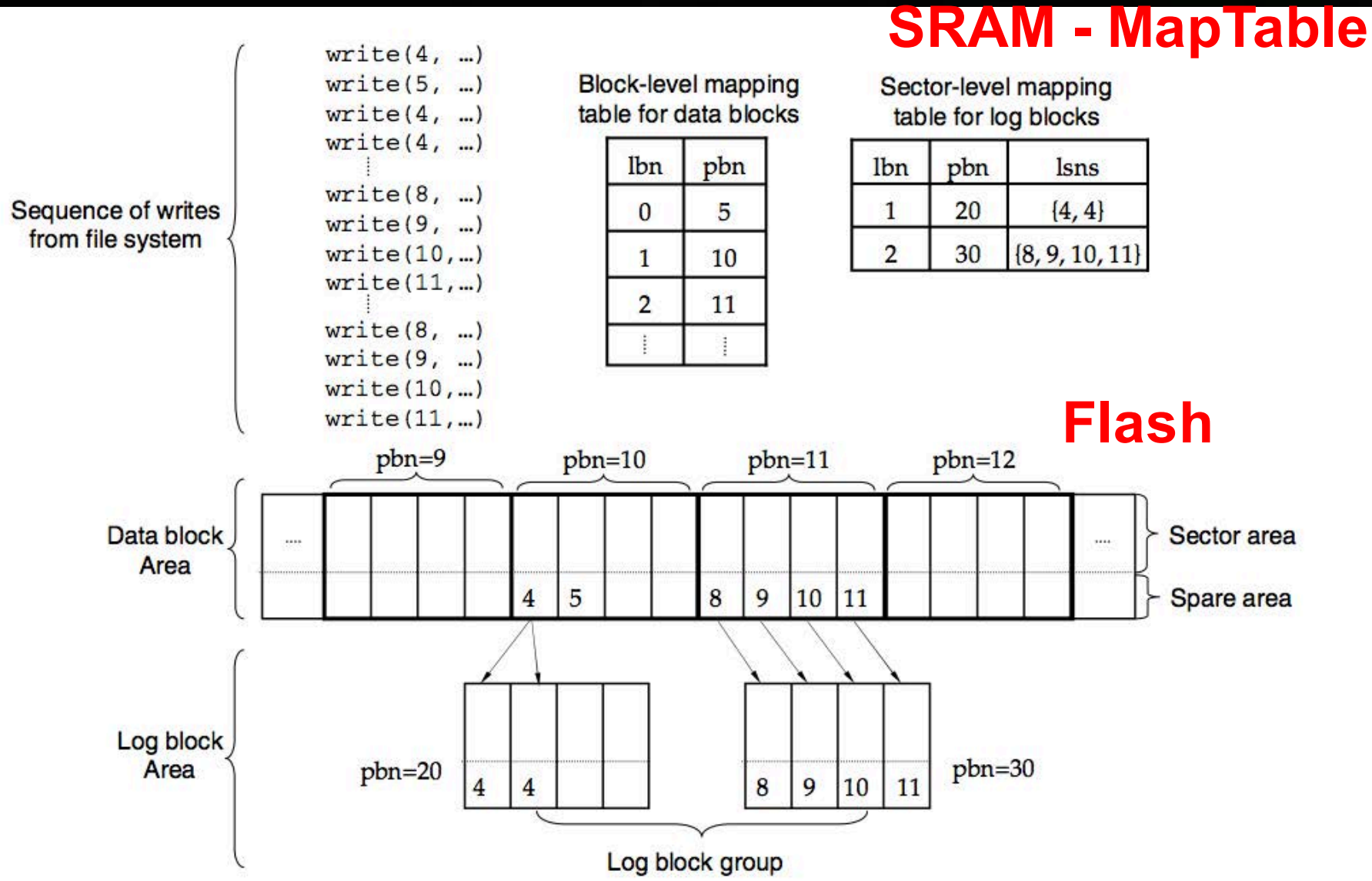
Step1 Update MapTable

LogBlk0	LogBlk1	LogBlk2	LogBlk3	LogBlk4
16	5	36	58	25
17	9	37	59	26
18	5	38	60	25
19	4	39		26
18	1	40		26
19	2	41		
22	3	42		
23	4	43		
16	5	44		
14		45		
16		46		
17		47		

Step3 Recycle DataBlk
Erase DataBlk

Step2 Assign LogBlk

1. BAST - MapTable



1. BAST - Issues

Besides RND write issue,
BAST also has block-thrashing issue:

DATA			
PBN0	PBN1	PBN2	PBN3
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

LOG	
LBN0	LBN2

Space Locality

W(0,4,8,12,4,8,12,0,0,4,12,8)

DATA			
PBN0	PBN1	PBN2	PBN3
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

LOG	
LBN0	LBN2

Temporal Locality

W(0,1,2,3,3,2,0,1,0,0,1,2)

1. FAST - FullAssoLogBlk

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	57
10	22		46	58
11	23		47	59

SLB0	RLB0	RLB1	RLB2	RLB3
0	32	56	36	104
1	65	57	37	32
2	163	58	38	
3	362	59	39	
4	963	36	40	
5	17	37	41	
6	18	54	54	
7	19	55	55	
8	40	56	56	
9	41	110	57	
10	42	58	46	
11	43	59	47	

1 Sequential LogBlk
N Rand LogBlks
Capture SeqPgs
Increase SwitchMerge

1. FAST - PartialMerge

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	57
10	22		46	58
11	23		47	59

Step1 Copy Pages
From DataBlk to SLB

SLB0	RLB0	RLB1	RLB2	RLB3
0	32	56	36	104
1	65	57	37	32
2	163	58	38	
3	362	59	39	
4	963	36	40	
5	17	37	41	
6	18	54	54	
	19	55	55	
	40	56	56	
	41	110	57	
	42	58	46	
	43	59	47	

1. FAST - PartialMerge

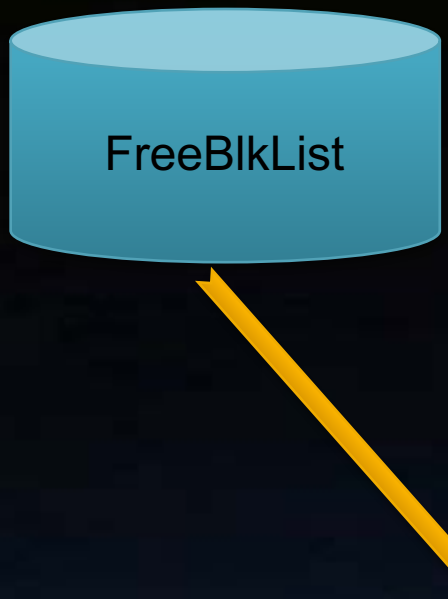
Step2 Recycle DataBlk
Erase DataBlk

DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	57
10	22		46	58
11	23		47	59

SLB0	RLB0	RLB1	RLB2	RLB3
0	32	56	36	104
1	65	57	37	32
2	163	58	38	
3	362	59	39	
4	963	36	40	
5	17	37	41	
6	18	54	54	
7	19	55	55	
8	40	56	56	
9	41	110	57	
10	42	58	46	
11	43	59	47	

1. FAST - PartialMerge

Step3 Update MapTable
Assign FreeBlk to SLB



DataBlk0	DataBlk1	DataBlk2	DataBlk3	DataBlk4
0	12	24	36	
1	13	25	37	
2	14	26	38	
3	15	27	39	
4	16		40	
5	17		41	53
6	18		42	54
7	19		43	55
8	20		44	56
9	21		45	57
10	22		46	58
11	23		47	59

SLB0	RLB0	RLB1	RLB2	RLB3
	32	56	36	104
	65	57	37	32
	163	58	38	
	362	59	39	
	963	36	40	
	17	37	41	
	18	54	54	
	19	55	55	
	40	56	56	
	41	110	57	
	42	58	46	
	43	59	47	

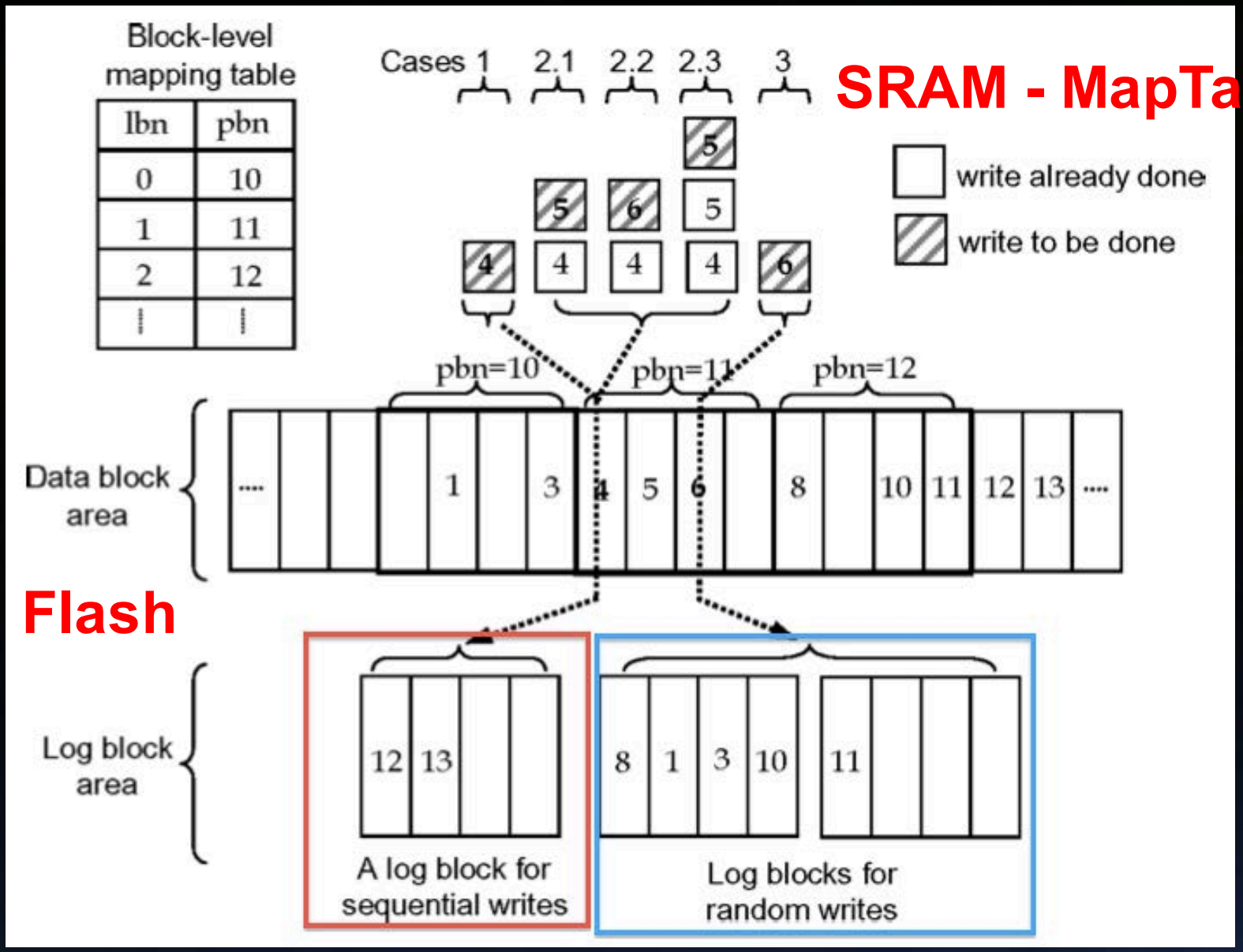
1. FAST – Cost Analysis

FullMerge -> Partial/SwitchMerge

FullMerge = 2 Erase + 64 Copy
ParitalMerge = 1 Erase + K Copy
SwitchMerge = 1 Erase

Assume a block has 64 pages.
K is in the range of [1,63]

2. FAST


Block-level mapping table
lbn
pbn
0
10
1
11
2
12
Cases 1
2.1
2.2
2.3
3
SRAM - MapTable
write already done
write to be done
pbn=10
pbn=11
pbn=12
Data block area
Flash
Log block area
A log block for sequential writes
Log blocks for random writes

1. FAST - Issues

1. **Does not consider multiple sequential streams.**

Thrashing issues still exist: if multiple sequential streams simultaneously come, they will be interleaved together.

StreamA:0,1,2,3,4
StreamB:12,13,14,15,16
StreamC:24,25,26,27,28

0 1 12 24 2 3 13 14 15 25 26 4 16 27 28

SLB0	RLB0	RLB1	RLB2	RLB3
0	32	56	36	104
1	65	57	37	32
2	163	58	38	
3	362	59	39	
4	963	36	40	
5	17	37	41	
6	18	54	54	
7	19	55	55	
8	40	56	56	
9	41	110	57	
10	42	58	46	
11	43	59	47	

1. FAST - Issues

2. Only treats block header as a VIP role:

- Not detecting sequential or not.
- BlkHdr is only necessary condition of SwitchMerge.
- One block can have more than 256 pages these days.
- If sequential stream begins from any place of a block, FAST cannot capture it.

SLB0	RLB0	RLB1	RLB2	RLB3
?	32	56	36	104
?	65	57	37	32
2	163	58	38	
3	362	59	39	
4	963	36	40	
5	17	37	41	
6	18	54	54	
7	19	55	55	
8	40	56	56	
9	41	110	57	
10	42	58	46	
11	43	59	47	

3. SBFAST

Motivation to solve the two issues in FAST.

Solution to issue1 Introducing multiple SLBs.
These SLBS can hold different streams or multiple same stream.

SLB0	SLB1	SLB2	RLB0	RLB1	RLB2
0	12	0	36	104	
1	13	1	37	32	
2	14	2	38		
3	15	3	39		
4	16		40		
5	17		41		
6	18		54		
7	19		55		
8	20		56		
9	21		57		
10			46		
11			47		

3. SBFAST

Solution to issue2 Introducing multiple subblocks.
No buffers -> no idea of the true Seq/Rand info.
Partite SLB into subblocks to increase the possibility to capture the sequential streams that does not begin from the block header.

SLB0	SLB1	SLB2	RLB0	RLB1	RLB2
	12	0	36	104	
	13	1	37	32	
	14	2	38		
	15	3	39		
4	16		40		
5	17		41		
6	18		54		
7	19		55		
8	20		56		
9	21		57		
10			46		
11			47		

3. SBFAST - Algorithm

Case 1. Input page IS (sub)block header

Case1.1 SLBs are all free, append it to SLB

Case1.2 SLBs are not all free, its hdr is in SLBs

Insert input page to that position.

(partial merge its hdr's newest blk if needed)

Case1.3 SLBs are not all free, its hdr is not in SLBs

If SLBs have a free blk then append

(merge a victim blk if needed)

Case 2. Input page IS NOT (sub)block header

Case2.1 Its hdr is in SLBs

Append it or trigger a partial/full merge in SLB

Case2.2 Its hdr is not in SLBs

Insert it into RLBs (full merge a victim blk if needed)

3. SBFAST - Algorithm

Partial Merge

Case 1.3.2.1 Write(12,..),(13,..),(14,..),(15,..),(16,..)

Before

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN6	LBN7	LBN8	LBN9
	0		INVALID		
	1		21		
	2		75		
	3		INVALID		
INVALID	INVALID	4	39		
INVALID	INVALID	5	127		
INVALID	INVALID	6	11		
7			INVALID		
INVALID	INVALID	8			
INVALID	INVALID	9			
	10				

Partial Merge

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN6	LBN7	LBN8	LBN9
	0 ->	0	INVALID		
	1 ->	1	21		
	2 ->	2	75		
	3 ->	3	INVALID		
INVALID	INVALID	4	39		
INVALID	INVALID	5	127		
INVALID	INVALID	6	<- 11		
7 ->			INVALID		
INVALID	INVALID	8			
INVALID	INVALID	9			
	10 ->	10			
		11			

if victimBlockInSEQ is a newestBlock
partialMergeNewestSameLBNBlockInSEQ(victimBlockInSEQ)
1 Erase for Data Block

After

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN65	LBN7	LBN8	LBN9
	INVALID	**12**	INVALID		
	INVALID	**13**	21		
	INVALID	**14**	75		
	INVALID	**15**	INVALID		
INVALID	INVALID	**16**	39		
INVALID	INVALID		127		
INVALID	INVALID		INVALID		
INVALID			INVALID		
INVALID	INVALID				
INVALID	INVALID				
	INVALID				

3. SBFAST - Algorithm

Full Merge in RLBs

Case 2.2.1 Write(3997,..)

Before

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN65	LBN7	LBN8	LBN9
0	INVALID		INVALID	27	INVALID
1	INVALID		21	25	INVALID
2	INVALID		75	551	INVALID
	INVALID		INVALID	2141	171
INVALID	INVALID	4	39	14	315
INVALID	INVALID		127	3	231
INVALID	INVALID	6	INVALID	147	81
INVALID		7	99	91	621
INVALID	INVALID	8	INVALID	997	3121
INVALID	INVALID	9	INVALID	67	147
	INVALID		INVALID	71113	17413
			79	7143	31

Full Merge

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN65	LBN7	LBN8	LBN9
0	INVALID		INVALID	27	INVALID
1	INVALID		21	25	INVALID
2	INVALID		75	551	INVALID
	INVALID	3	INVALID	2141	171
INVALID	INVALID	4	39	14	315
INVALID	INVALID		127	<- 3	231
INVALID	INVALID	6	INVALID	147	81
INVALID		7	99	91	621
INVALID	INVALID	8	INVALID	997	3121
INVALID	INVALID	9	INVALID	67	147
	INVALID		INVALID	71113	17413
			79	7143	31

victimBlock

(1) fullMergeInRND for each associated block and
(2) Copy pages to SEQ_Log_Blocks if possible
Erase = 10 Data + 1 Log
10 is the associated block number

After

SEQ_Log_Blocks | **RND_Log_Blocks**

LBN0	LBN2	LBN65	LBN7	LBN12	LBN9
0	INVALID		INVALID	**3997**	INVALID
1	INVALID		21		INVALID
2	INVALID		75		INVALID
	INVALID	3	INVALID		171
INVALID	INVALID	4	39		315
INVALID	INVALID		127		231
INVALID	INVALID	6	INVALID		81
INVALID		7	99		621
INVALID	INVALID	8	INVALID		3121
INVALID	INVALID	9	INVALID		147
	INVALID		INVALID		17413
			79		31

3. SBFAST - Evaluation

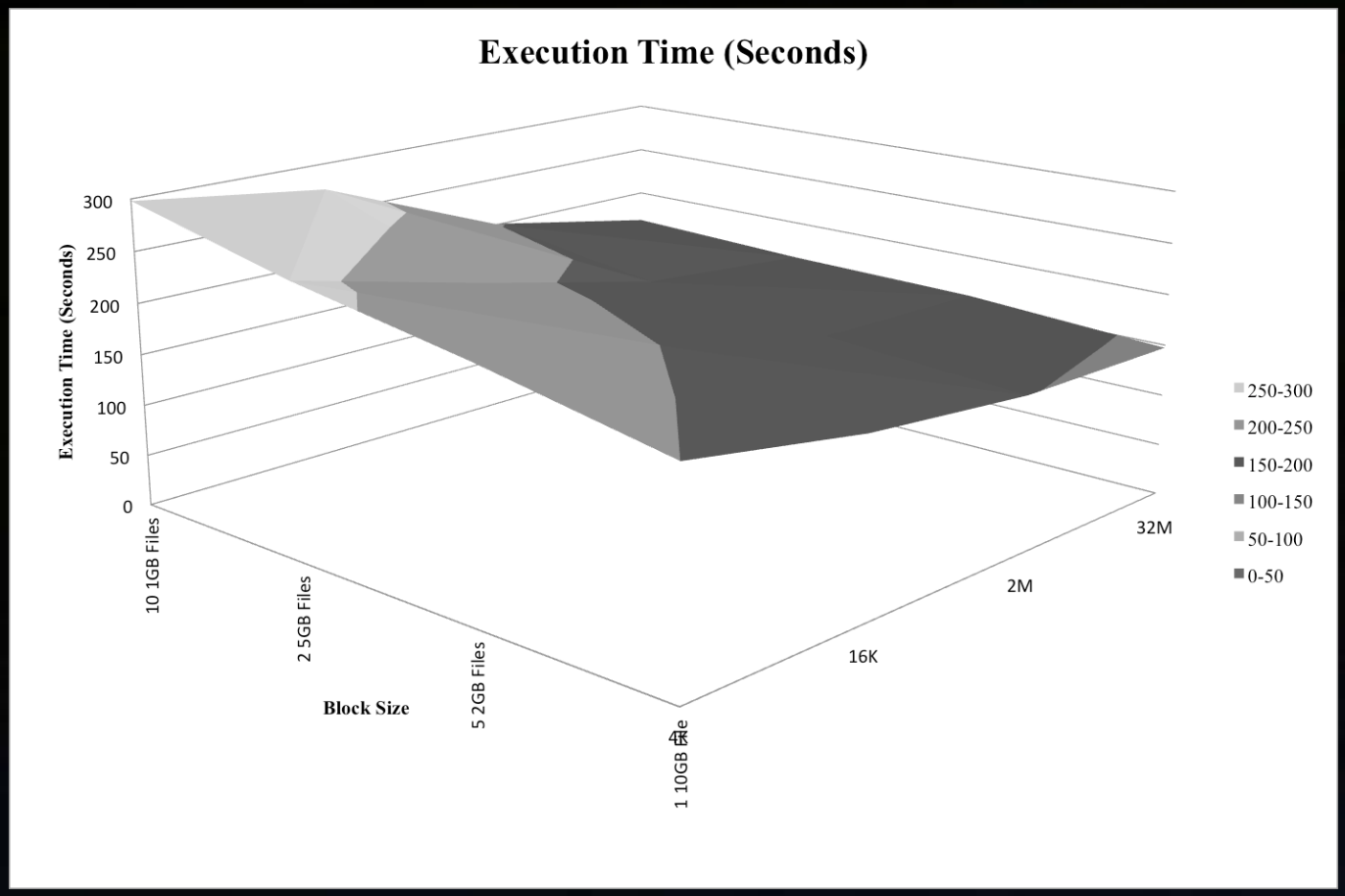
Part A: Parameter Tuning

1. **Multiple SLBs and Stream?**
2. **Subblock size?**

Part B: Workloads Performance

1. **Seq/Rand Ratio (SRR)**
2. **Sub Stream Length**
3. **Block Header Ratio**
4. **Sub Block Header Ratio**
5. **Scuffling Degree**
6. **TPC-E**
7. **MSR-WEB0**
8. **RealTrace**

3. SBFAST – Evaluation PAE1

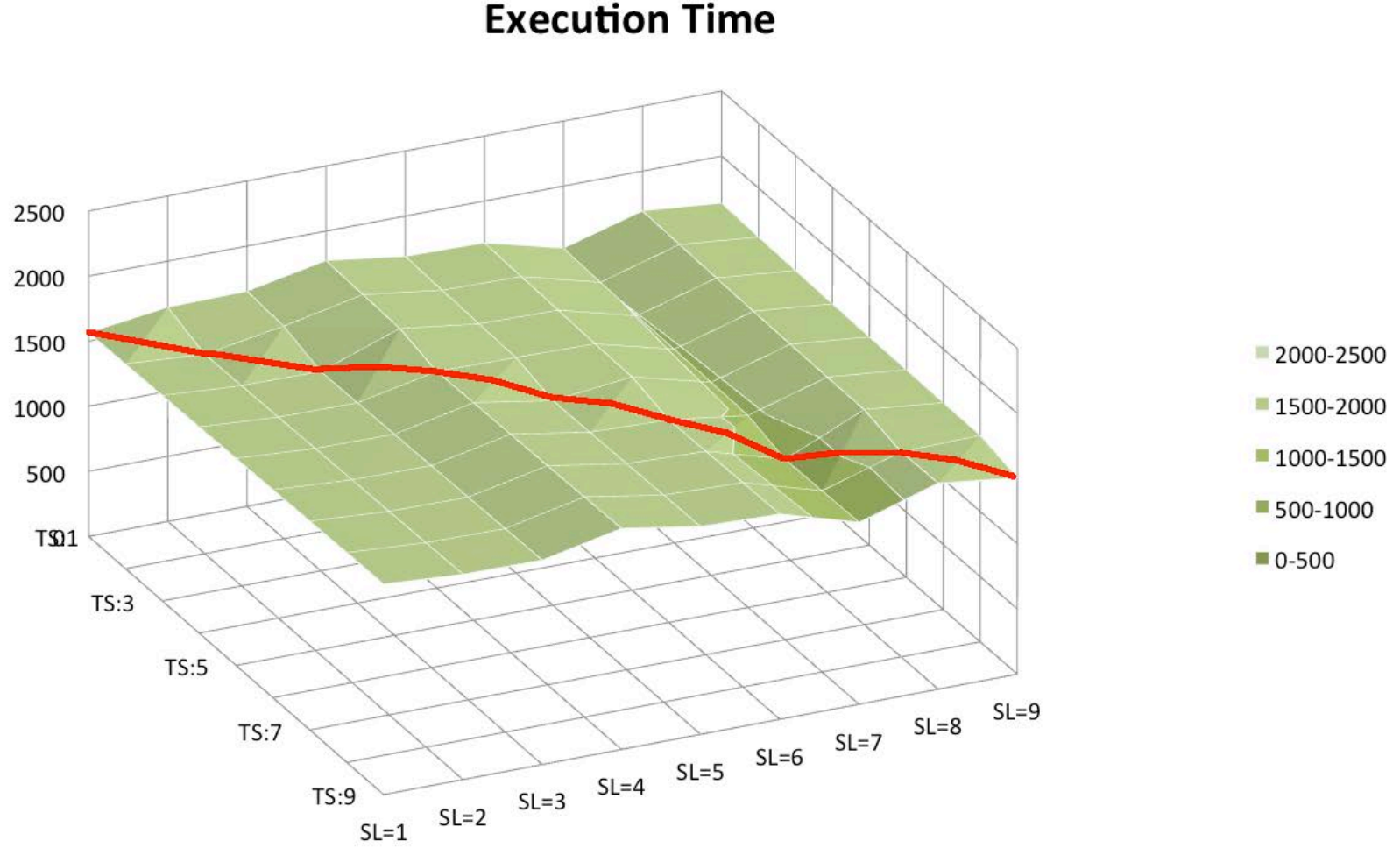


	Execution Time (Sec)			
Block Size	10 1GB Files	2 5GB Files	5 2GB Files	1 10GB File
4K	298	261	233	199.97
16K	274	216.9228	199.330848	168.32
2M	201.36	176.36	162.0576	150.2664
32M	168.39	161.79	158.88	147.32

3. SBFAST – Evaluation PAE1

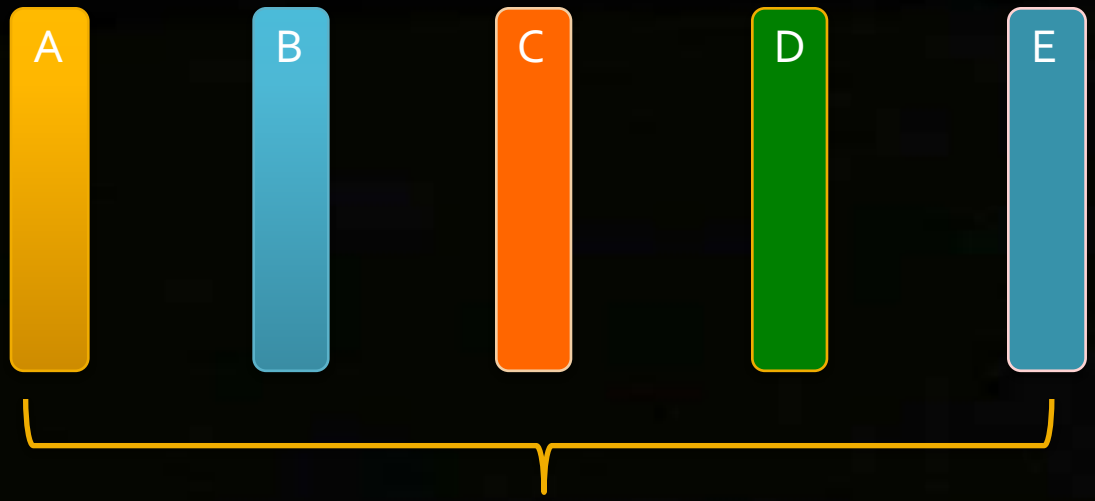
Part One: Parameter

1. **Multiple SLBs and Stream? When # of SeqStream = # of SLBs, best.**



TRACE_SEQ#									
Cost	TS:1	TS:2	TS:3	TS:4	TS:5	TS:6	TS:7	TS:8	TS:9
SL=1	1564.25	1642.9	1650.98	1771.32778	1687.8875	1677.38803	1520.53848	1692.07533	1636.0512
SL=2	1569.86	1530.1	1630.5	1757.08393	1675.47513	1665.28108	1505.52158	1680.72875	1623.3975
SL=3	1575.17	1535.8	1529.28	1744.06713	1664.73183	1653.39633	1495.85083	1671.03858	1613.6864
SL=4	1581.73	1541.7	1535.88	1661.341	1653.3856	1643.13678	1486.0624	1662.61273	1603.8224
SL=5	1587.85	1548.7	1542.55	1669.03158	1569.9038	1636.2861	1478.97303	1657.11175	1597.5085
SL=6	1593.77	1555.4	1548.02	1674.99635	1574.38113	1554.4378	1472.31295	1651.54638	1592.4465
SL=7	1600.77	1560.8	1554.3	1680.15098	1580.43688	1559.90425	1383.9365	1649.0097	1586.5709
SL=8	1607.18	1567.1	1560.9	1687.89525	1587.1095	1566.9842	1389.24095	1572.8543	1583.3114
SL=9	1613.23	1573.4	1567.24	1694.46098	1593.83473	1573.43375	1395.38703	1579.6303	1506.3716

3. SBFAST – Evaluation PAE1

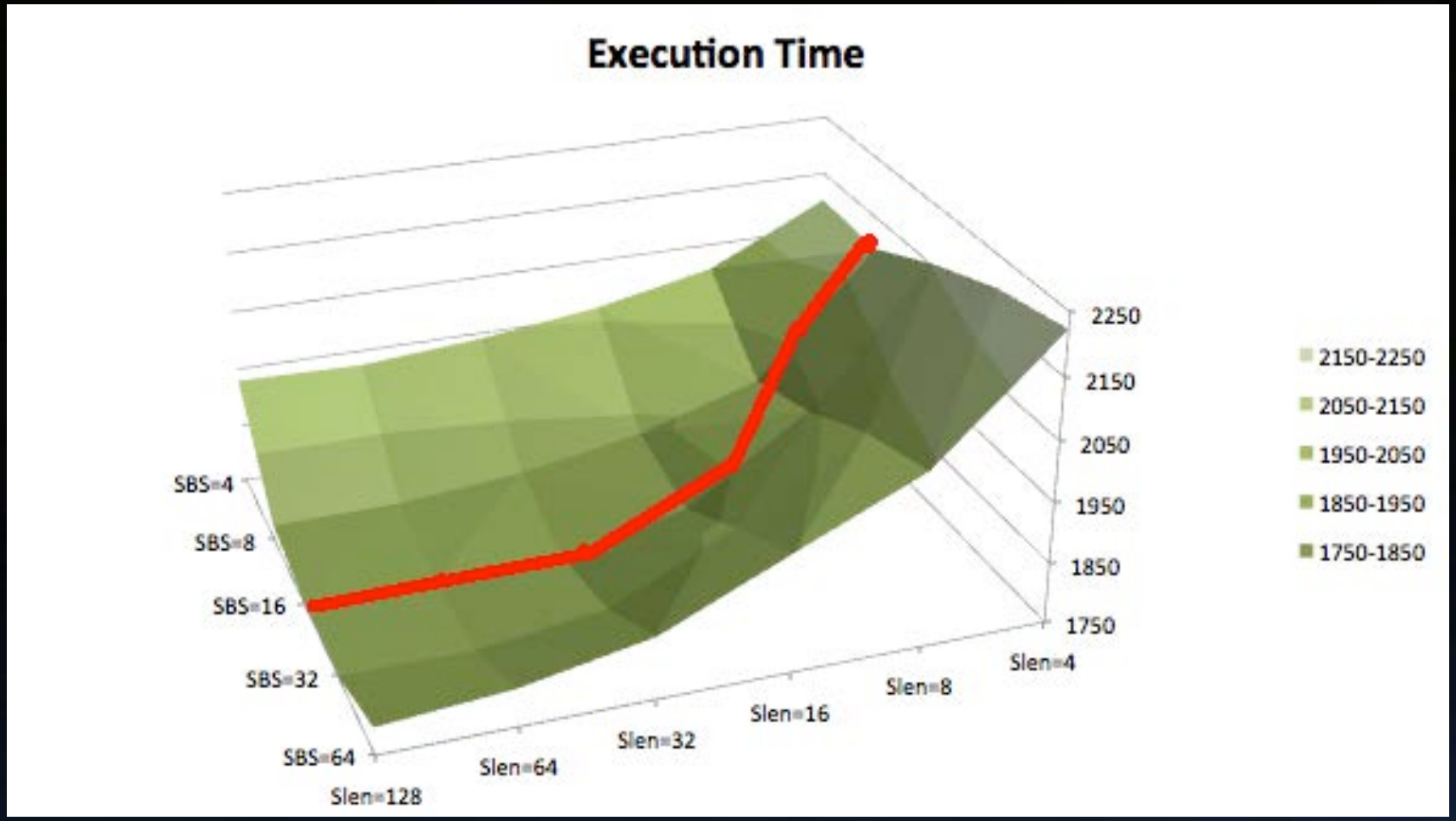


SLB0	SLB1	SLB2	SLB3	SLB4	RLB0	RLB1	RLB2	RLB3	RLB4
	48		12	72	31	104			
	49		13	73	37	3125			
	50		14	74	38				
			15		39				
4		124	16		43				
5		125	17	77	41				
6		126	18	78	531				
7		127	19		55				
8	56	128	20		53				
9			21		57				
10					43				
11					47				

3. SBFAST – Evaluation PAE2

Part One: Parameter

2. Subblock size?



3. SBFAST – Evaluation PAE2

Why the "matching is the best" principle does not work?

	Execution Time					
	SubStreamLen->					
Time	Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
SBS=4	2102.23953	2006.08555	1964.9133	1940.1727	1924.62445	1929.41208
SBS=8	2095.8487	1891.44743	1829.69675	1794.73355	1776.45308	1774.63523
SBS=16	2153.17643	1933.96043	1799.2782	1746.06875	1723.71195	1717.77535
SBS=32	2198.73693	1997.79383	1874.804	1778.4235	1739.45013	1734.9771
SBS=64	2223.75138	2037.69488	1945.91955	1852.68268	1812.25273	1795.0482

3. SBFAST – Evaluation PAE2

Let's think about benefit and penalty of partition:

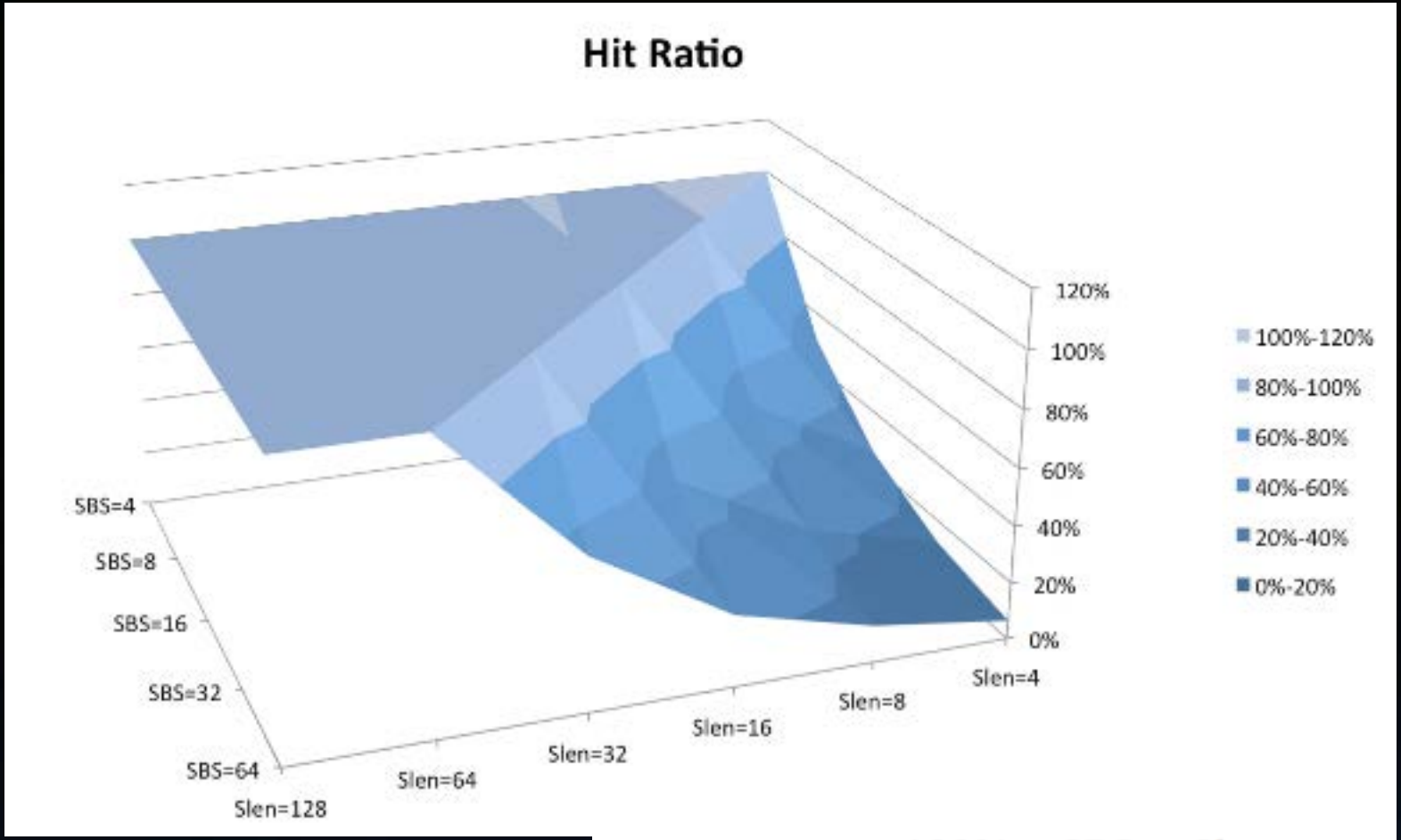
<u>Benefit 1: Hit Ratio</u>

The ratio of # of captured sequential sub streams to # of all sub streams, the higher the better:

$$Hit\ Quality = \frac{AvgCapSubStreamLen}{SubStreamLen}$$

		Benefit 1: Hit Ratio					
		SubStreamLen->					
SubBlockSize		Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
	SBS=4	99.99%	99.99%	100.00%	99.89%	99.88%	99.88%
	SBS=8	50.52%	99.97%	100.00%	99.88%	99.89%	99.89%
	SBS=16	25.41%	50.04%	100.00%	99.89%	99.89%	99.91%
	SBS=32	13.01%	24.57%	49.72%	99.94%	99.87%	99.94%
	SBS=64	6.77%	12.97%	25.09%	53.06%	99.94%	100.00%

3. SBFAST – Evaluation PAE2



$$Hit\ Ratio = \begin{cases} 100\%, & SBS \geq Slen \\ \dfrac{Slen}{SBS}, & SBS < Slen \end{cases}$$

3. SBFAST – Evaluation PAE2

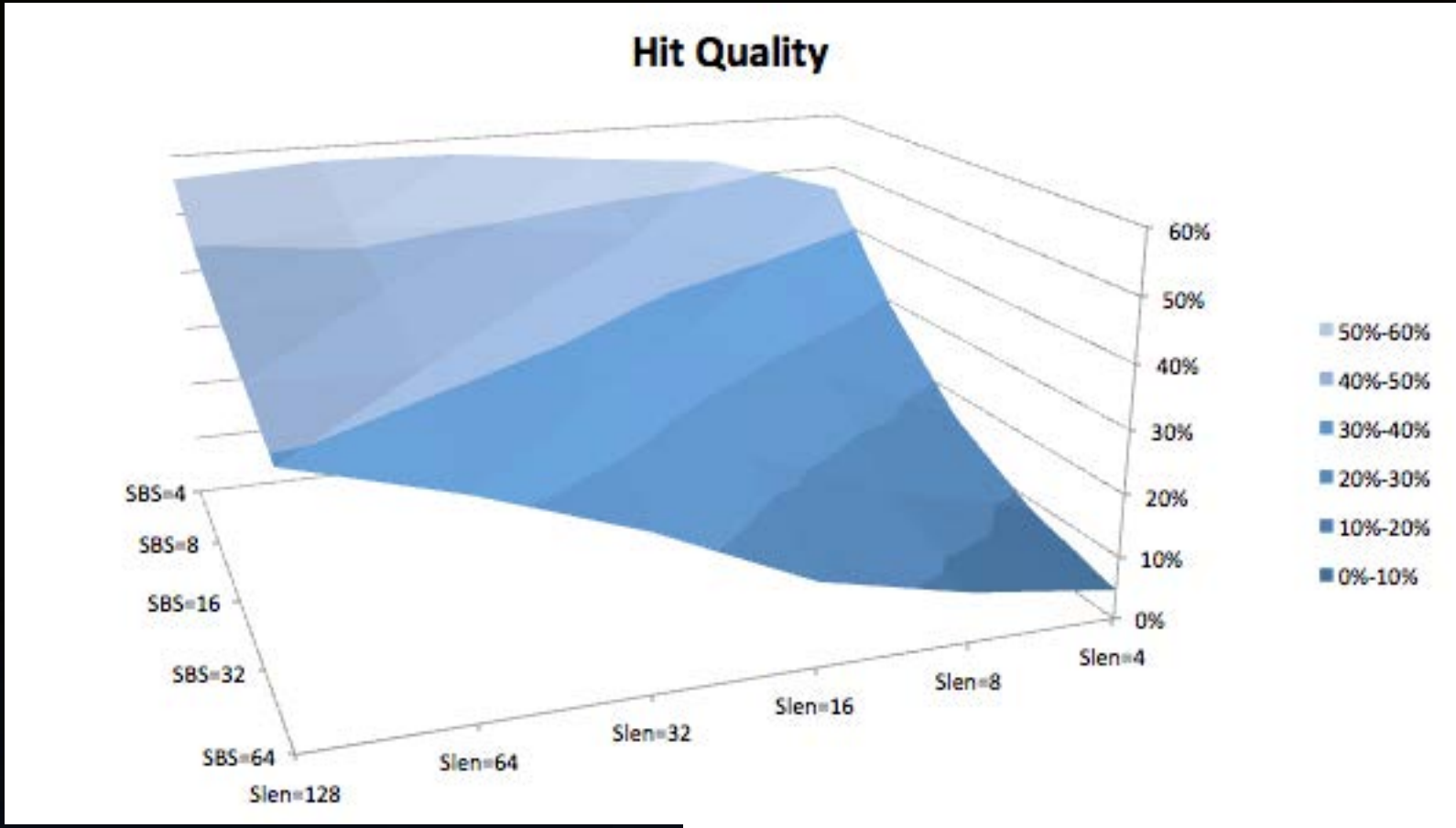
Benefit 2: Hit Quality

Hit ratio is the ratio of average captured sub streams length to sub stream length, the higher the better:

$$Hit\ Quality = \frac{AvgCapSubStreamLen}{SubStreamLen}$$

		Benefit 1: Hit Ratio					
		SubStreamLen->					
SubBlockSize		Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
	SBS=4	99.99%	99.99%	100.00%	99.89%	99.88%	99.88%
	SBS=8	50.52%	99.97%	100.00%	99.88%	99.89%	99.89%
	SBS=16	25.41%	50.04%	100.00%	99.89%	99.89%	99.91%
	SBS=32	13.01%	24.57%	49.72%	99.94%	99.87%	99.94%
	SBS=64	6.77%	12.97%	25.09%	53.06%	99.94%	100.00%

3. SBFAST – Evaluation PAE2



$$y = aSBS^3 + bSBS^2 + cSBS + d$$

3. SBFAST – Evaluation PAE2

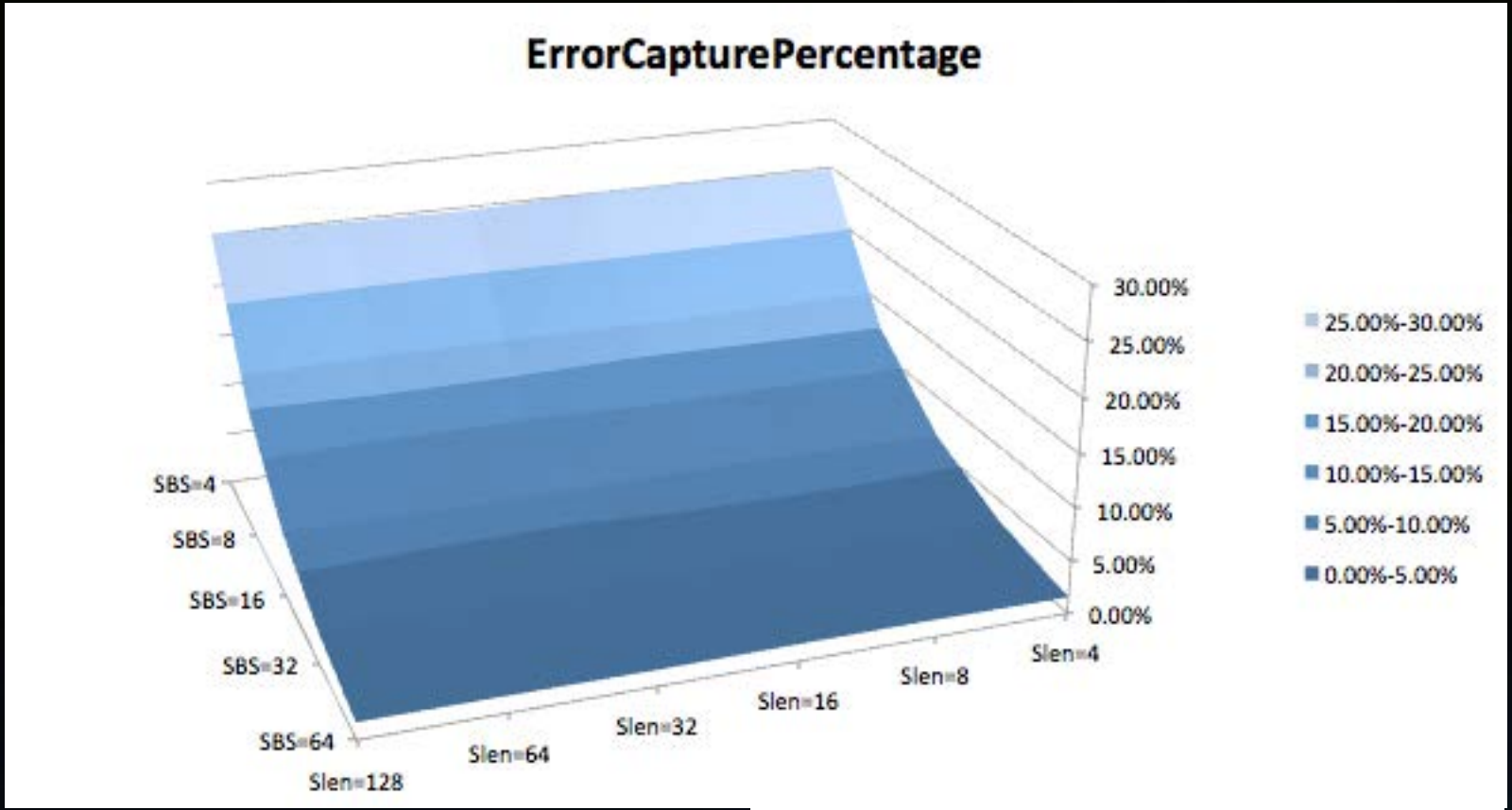
Penalty

The ratio of # of captured random pages in SLBs to the total # of random pages in the trace, the lower the better:

$$ErrorCapturePercentage = \frac{PagesCapturedFromRNDSubStream}{RandomPages\#}$$

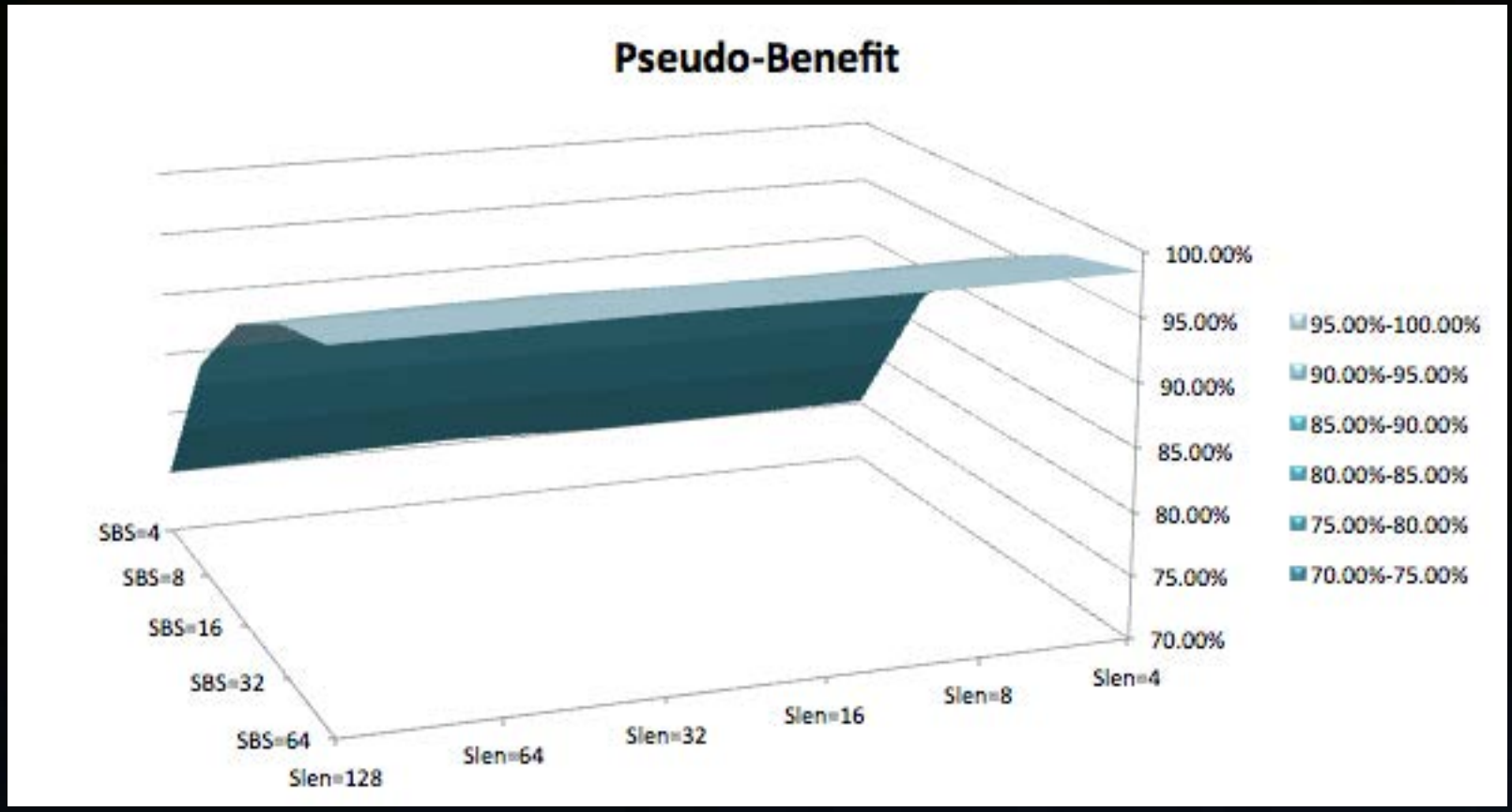
		Penalty: ErrorCapturePercentage					
		SubStreamLen->					
SubBlockSize		Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
	SBS=4	24.79%	24.97%	25.01%	24.65%	24.80%	25.04%
	SBS=8	12.39%	12.49%	12.65%	12.40%	12.34%	12.63%
	SBS=16	6.17%	6.31%	6.32%	6.06%	6.12%	6.32%
	SBS=32	3.08%	3.21%	3.16%	3.04%	3.12%	3.19%
	SBS=64	1.55%	1.65%	1.60%	1.52%	1.56%	1.58%

3. SBFAST – Evaluation PAE2

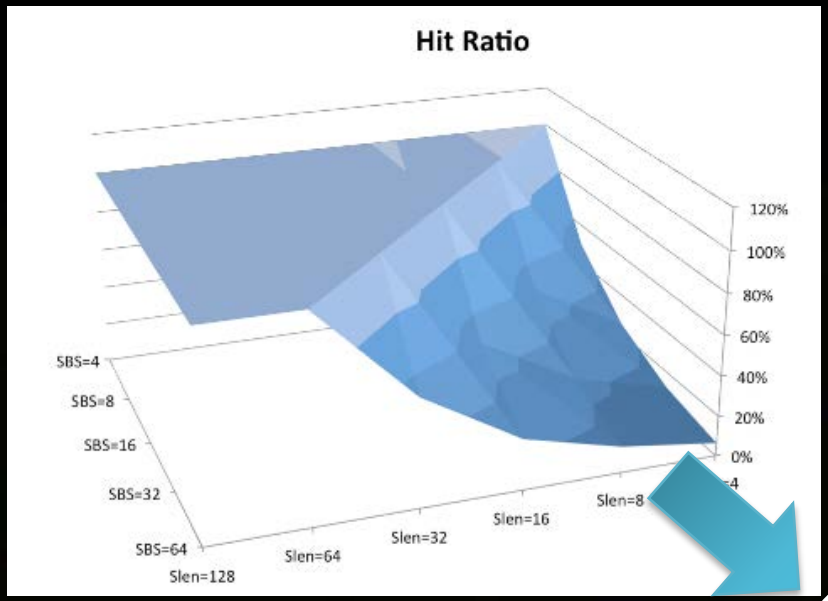


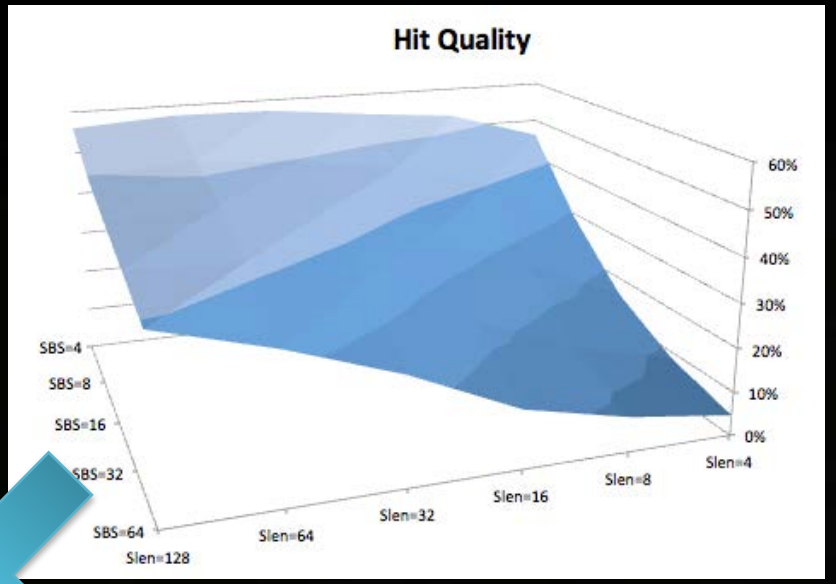
$$ErrorCapturePercentage = \frac{1}{SBS}$$

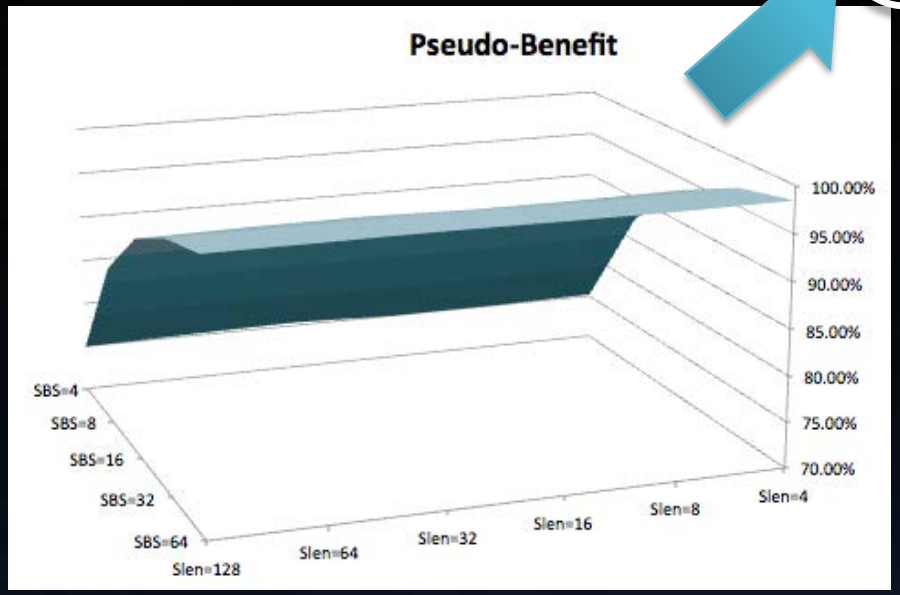
3. SBFAST – Evaluation PAE2



Convert Penalty to Pseudo Benefit







	Execution Time					
	SubStreamLen->					
Time	Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
SBS=4	2102.23953	2006.08555	1964.9133	1940.1727	1924.62445	1929.41208
SBS=8	2095.8487	1891.44743	1829.69675	1794.73355	1776.45308	1774.63523
SBS=16	2153.17643	1933.96043	1799.2782	1746.06875	1723.71195	1717.77535
SBS=32	2198.73693	1997.79383	1874.804	1778.4235	1739.45013	1734.9771
SBS=64	2223.75138	2037.69488	1945.91955	1852.68268	1812.25273	1795.0482

3. SBFAST – Evaluation PAE2

We explained why the lowest cost points stop at some points that are not the expected matching points.

But is that true, in all cases 16 is always the best point?

We tuned the SRR when Slen=64 and see whether the best points always are located at SBS=16?

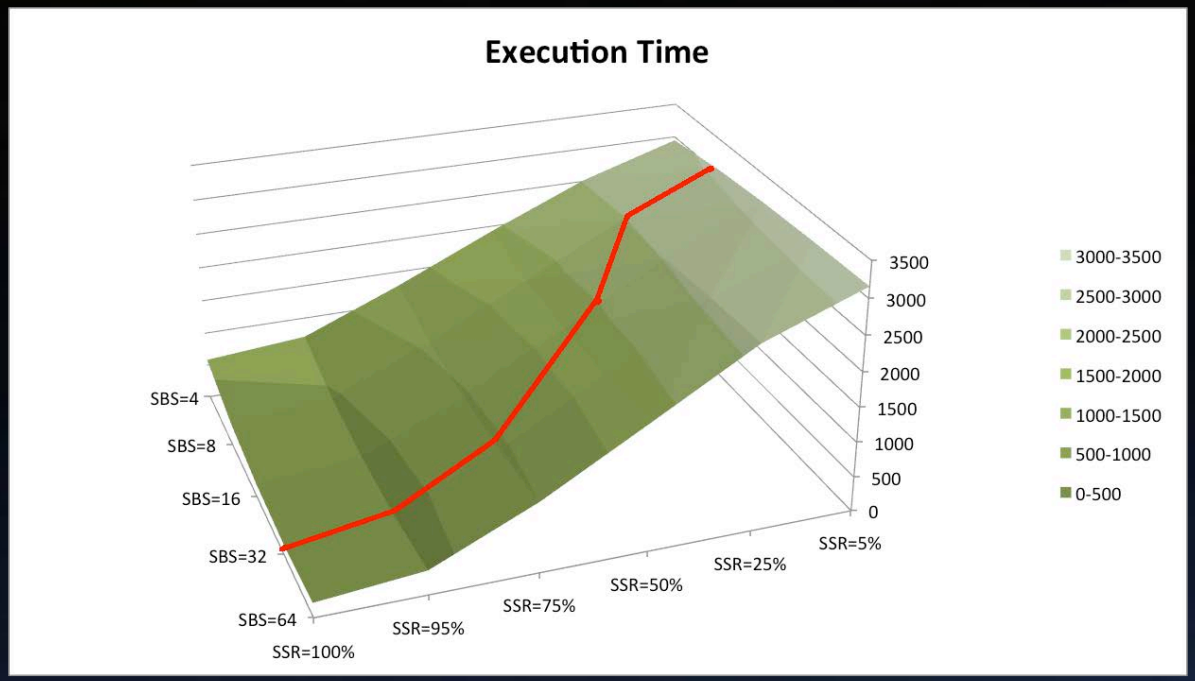
	Execution Time					
	SubStreamLen->					
Time	Slen=4	Slen=8	Slen=16	Slen=32	Slen=64	Slen=128
SBS=4	2102.23953	2006.08555	1964.9133	1940.1727	1924.62445	1929.41208
SBS=8	2095.8487	1891.44743	1829.69675	1794.73355	1776.45308	1774.63523
SBS=16	2153.17643	1933.96043	1799.2782	1746.06875	1723.71195	1717.77535
SBS=32	2198.73693	1997.79383	1874.804	1778.4235	1739.45013	1734.9771
SBS=64	2223.75138	2037.69488	1945.91955	1852.68268	1812.25273	1795.0482

	Execution Time					
	SubStreamLen->					
Time	SSR=5%	SSR=25%	SSR=50%	SSR=75%	SSR=95%	SSR=100%
SBS=4	2946.86663	2517.50943	1924.62445	1288.75728	708.63245	583.643
SBS=8	3052.79245	2475.7785	1776.45308	1060.2891	447.538975	322.28045
SBS=16	3112.27838	2512.643	1723.71195	959.19955	324.2222	197.2673
SBS=32	3143.51858	2585.91455	1739.45013	936.7041	288.338075	157.9574
SBS=64	3159.4702	2631.16763	1812.25273	1004.42555	345.911875	208.8291

Exe_Time	Slen
SBS=4	
SBS=8	
SBS=16	
SBS=32	
SBS=64	

Larger Benefit

Larger Penalty



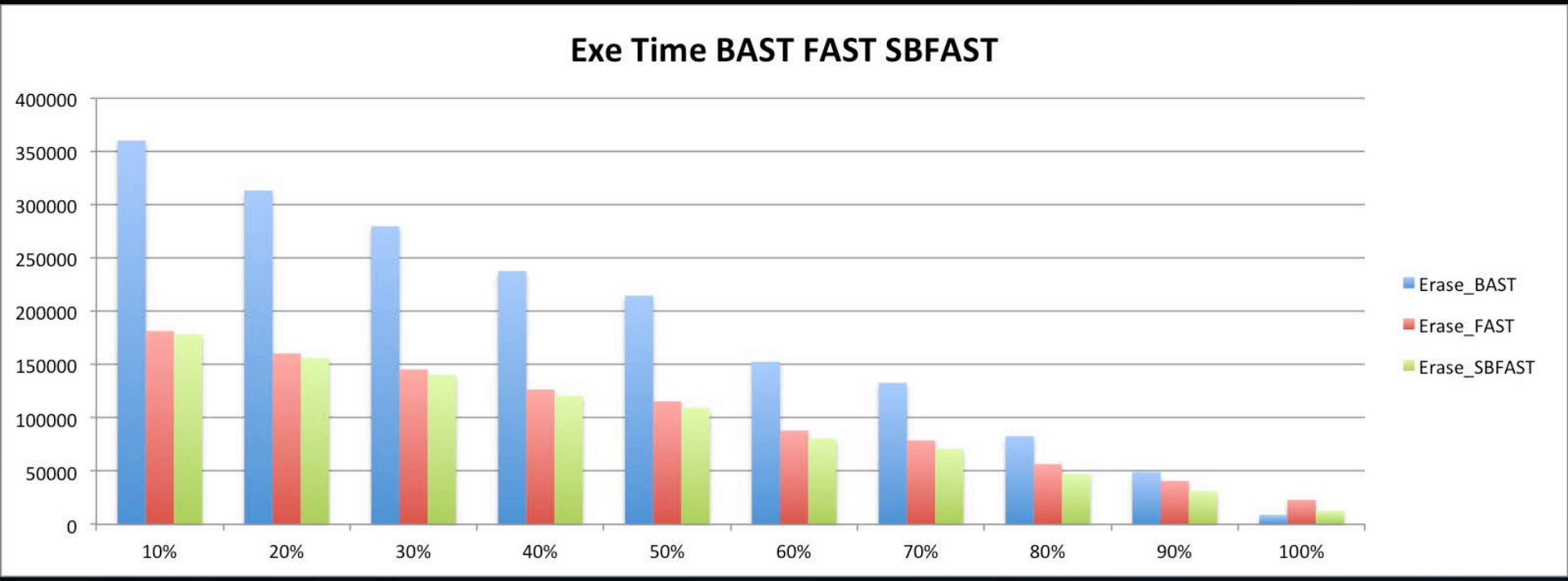
3. SBFAST - Evaluation

Part B: Workloads Performance

1. **Seq/Rand Ratio (SRR)**
2. **Sub Stream Length**
3. **Block Header Ratio**
4. **Sub Block Header Ratio**
5. **Shuffling Degree**
6. **TPC-E**
7. **MSR-WEB0**
8. **RealTrace**

3. SBFAST – Evaluation PBE1

1. Seq/Rand Ratio (SRR)



3. SBFAST – Evaluation PBE1

2. Sub Stream Length

3. SBFAST – Evaluation PBE1

3. Block Header Ratio

3. SBFAST – Evaluation PBE1

4. Sub Block Header Ratio

3. SBFAST – Evaluation PBE1

5. Shuffling Degree

3. SBFAST – Evaluation PBE1

6. TPC-E

		Execution Time								
		SubStreamLen->								
SubBlockSize	Time	SLB=1	SLB=2	SLB=3	SLB=4	SLB=5	SLB=6	SLB=7	SLB=8	SLB=9
	SBS=4	701.549475	700.771275	700.3862	694.37545	692.987925	691.98035	691.105775	692.16455	690.39745
	SBS=8	710.4679	713.130925	701.5343	699.951425	705.36495	700.167075	705.7453	704.01475	699.46355
	SBS=16	729.623	718.6179	717.209825	721.5279	711.4822	718.837475	720.045775	720.7431	715.317725
	SBS=32	741.2789	731.7171	733.77205	735.8823	730.42405	731.364025	731.26225	730.057225	725.3365
	SBS=64	747.130775	742.90395	748.35685	744.28425	745.877525	742.5479	744.588725	747.102925	736.1361
		BAST=	818.002	Saved=	15.60%		FAST=	755.120725	Saved=	8.57%

7. MSR-WEB0

	Execution Time								
	SubStreamLen->								
Time	SLB=1	SLB=2	SLB=3	SLB=4	SLB=5	SLB=6	SLB=7	SLB=8	SLB=9
SBS=4	652.10425	620.612325	602.69435	597.0381	592.039375	589.492725	597.0381	600.9239	662.069075
SBS=8	457.66295	421.844675	408.0392	401.819525	400.19335	404.0027	401.819525	473.79185	599.38155
SBS=16	250.460325	224.594125	217.78915	219.559625	226.98125	246.667975	219.559625	402.657575	515.950175
SBS=32	245.78865	224.505075	219.42095	221.17775	229.62265	248.676925	221.17775	405.74265	519.769425
SBS=64	221.46295	207.4071	205.96125	209.616325	219.780925	240.9349	209.616325	413.585825	524.595775
	BAST=	627.654	Saved=	67.19%		FAST=	313.68755	Saved=	34.34%

3. SBFAST – Evaluation PBE1

8. RealTrace

		Execution Time								
		SubStreamLen->								
SubBlockSize	Time	SLB=1	SLB=2	SLB=3	SLB=4	SLB=5	SLB=6	SLB=7	SLB=8	SLB=9
|	SBS=4	1057	1025	1006	776	775	742	753	814	938
	SBS=8	700	672	663	518	528	530	574	682	801
	SBS=16	588	570	566	443	461	449	540	650	770
	SBS=32	526	507	499	396	420	342	522	631	756
	SBS=64	453	435	438	356	388	401	503	616	745
		BAST=	856.66098	Saved=	49.71%		FAST=	565.11	Saved=	23.25%