#SHAREorg

SHARE
Technology · Connections · Results

Using Solid-State Devices to Solve a DB2 INSERT Problem

Terry L. Berman
DST Systems, Inc.

March 12, 2012
Session 10842

Agenda

Case 1 – Batch INSERT SLA

- Problem, prior remedies

Solid State Devices (SSD)

- Technology “sweet spot”
- Deployment
- Case 1 results

Case 2 – CICS mass INSERTs

- The business case

Futures …

Measurement

- Analyze performance issue for bottlenecks
- Designate the SSD candidates
- Measure, report results

www.dstsystems.com

DST SYSTEMS

SHARE Technology · Connections · Results

Retirement Solutions | Health Solutions

Mainframe

z196

z10's

DB2 for z/OS 9 → 10

Database Support Infrastructure

IBM 8x00

DASD

Automations

Repositories

Information delivery

sas

Case 1 – Single Batch INSERT Bottleneck

Insert a large number of rows in the shortest possible time

common business scenario

- Retirement Solutions
- Batch calculate Dividends (month-end)
- … to be inserted into tables TRN, ATRN
- Rest of Batch has to wait
- SLA's tied to various Batch deliverables

TRN 25 min

ATRN 80 min

Largest customer 1.2M INSERTs

250 / sec

Parallelism – (Divide and Conquer)

- Multiple jobs
  - Subdivide data
  - 8 simultaneous jobs, 8 initiators
  - Less complex vs. multi-tasking in same AS (LE)

OK for now, but can scale to larger customers in future?

- DB2 for z/OS lacks a parallel INSERT capability
  - As DB2 for LUW provides

INSERT processing (costs)

INDEX

TABLESPACE

- Maintain the B-TREE
- Enforce uniqueness
- Split if necessary

- Put new rows (into Cluster order)

- Space search (spacemap pages)
  - Conserve space if possible
  - Extend object if not

- Partitioned
- Segmented, UTS PBG
- UTS PBR

- Log changes under UOW

- Data Sharing integrity

- Changed pages “trickle out” asynchronously

DB2 Instrumentation (IFI Traces) z/OS

SMF

ACCTG	STAT	PERFM	SMF 42-6
Each DB2 thread	Subsystem metrics, per minute	Specialized traces	Dataset I/O

… exploited systematically

PDB

DB2 tables

ACCTG breaks down elapsed times

CLASS 1	CLASS 2	CLASS 3
Clock	In DB2	Suspend (30 causes)

CPU 25%

waiting

Sync_Read_Wt 60%

Async_Read_Wt

Lock / Latch
Commit (log sync)
Log Write
Page Latch (BP)
Data Sharing

TRN / ATRN
INSERT

other INSERTs

Sync_Read Breakdown – Bufferpool Stats

Tablespace		Indexes
56%	Sync_Read %	44%

- Maintaining Clustering
  - SELECT gain vs. **INSERT pain**
  - Due diligence
  - REORG (~24x7) to recluster
- APPEND removes cleanly

450 / sec

Share same BP

Which index(es) need the I/O relief?

IFCID 199 – Dataset Stats – Sync_Read

- STAT CLASS(8) – on DB2 startup (ZPARM)
- -DIS BP LSTATS information (held in Buffer Manager)
- Each open dataset, at least 1 I/O / Second

Timestamp (STCK)	**Each Minute**
Database, Space, Piece/Part	**Dataset (“object”)**
# Sync_Read	✸ Health check
Sync_Read Response	ms granularity

✸ Busy subsystem main index BP 5/4/2011

IFCID 002 Subsystem BP Stats	**IFCID 199**
62,739,166	62,686,478

99.9% attributable by object

… an I/O monitor built into the engine

Which index(es) need the I/O relief?

IFCID 199

IX Name		Key len		Clust
X0	P	12	NPI	77%
X1	D	50	Clust Part	~100%
X2	D	38	NPI	89%
X3	D	28	NPI	97%

% Sync_Read	
~0	"Dumb-IDs" in sequence
90%	**SKIP SEQUENTIAL**
~0	"Dumb-IDs" in sequence
10%	Dividend adjacency

What IS the troublesome X1 insert pattern?

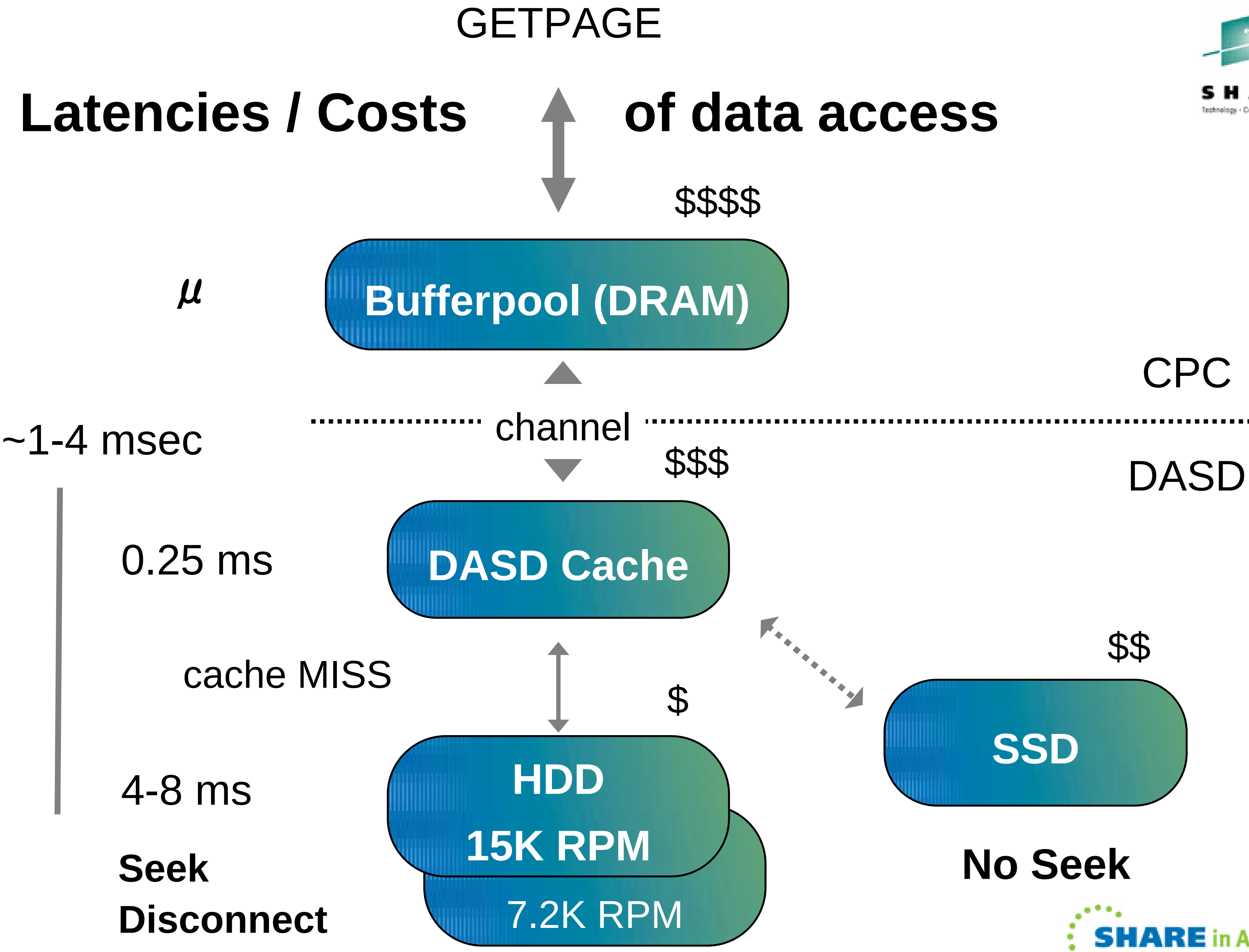
PERFM TRACE *(used by Buffer Pool tools)*

IFCID		Qualify by job?	Qualify by object?	Volume/ overhead
6, 7	Sync I/O Start , End	Yes	NO	
9,10	Async I/O Start, End	NO	NO	High
198	GETPAGE	Yes	NO	Very high

The DB2 Log ***No application remedy***

DSN1LOGP SYSPRINT post-process

Latencies / Costs of data access



DASD Cache Hit %

4 - 8 ms MISS

- Critical in I/O Response of Sync_Read

0.25 ms HIT

- Cache algorithm uses adjacency patterns
  - Track = 12 (4K) DB2 Pages
- Less good at busy times of high I/O activity
  - Cannot designated “favored” objects

I/O Response = Sync_Read_Wt / Sync_Read

For TRN / ATRN Inserts (at a Busy time)

= 3-4 msec

Solid State Drives – Flash Memory “Cells”

SLC	Single-Level Cell	1 bit	More expensive (/ bit)	100,000 rewrites lifetime
MLC	Multi-Level Cell	2 bits	Cheaper	10,000 rewrites

- **More reliable** vs. HDD
  - No moving parts
  - 10 years lifetime vs. 5 years
- **Less durable** vs. HDD
  - Limited number of rewrites
  - Wear out, must be replaced

- 50% power
- Less noise
- **7 – 10x the cost** of HDD

IBM DS8x00 implementation of SSD

DS8000 : Introducing Solid State Drives REDP-4522-00 June 4, 2009

SLC only

More reliable than MLC

RAID 5

RAID 6 redundancy not req'd

RAID 10 performance not req'd

Error Detection / Correction

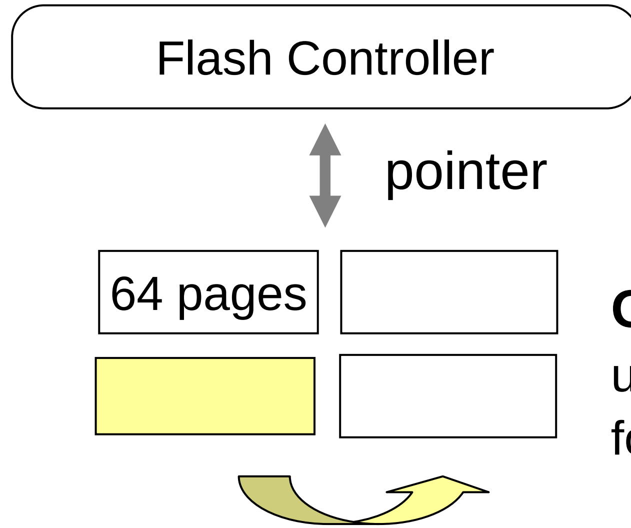
Assures accurate data

Bad blocks removed

Wear-Leveling

Even out use

Move static data to high-wear blocks



Over-provisioned

up to 75% add. cells

for redundancy

Each write uses a new block

Copies all data

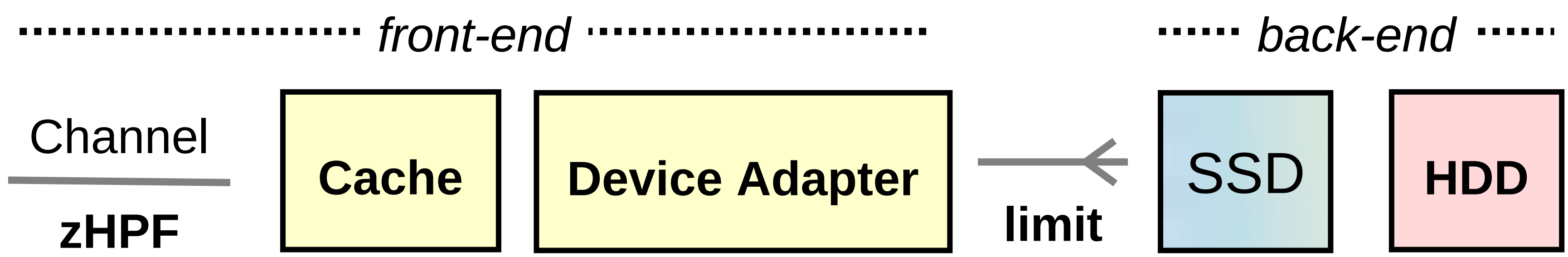
No hot spots

0.8 msec response (added "enterprise" microcode)

SSD sweet spot – tuning I/O path

Ready to Access DB2 for z/OS Data on Solid-State Drives

Jeffrey Berger, Paolo Bruni – REDP-4537-00 Dec 7, 2009

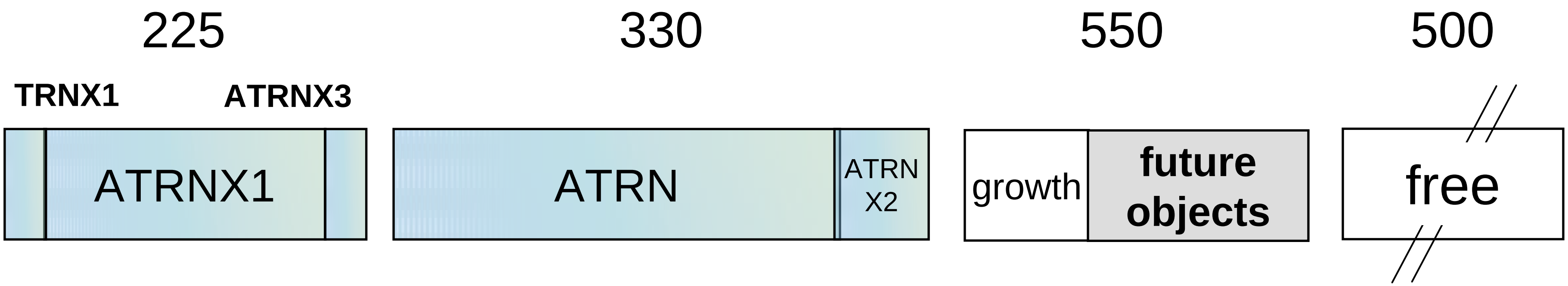


	Bottleneck	SSD benefit?
Random I/O	back-end seek	**NO SEEK** ✓
Sequential Prefetch	front-end throughput	no advantage vs. HDD
Dynamic Prefetch	front-end throughput	no advantage vs. HDD
List Prefetch	“	minor benefit

Adaptive
Multistream
Prefetch

Hyper-PAVs

Deploying SSD in ~1600GB “ranks”



Phase 1
DIV Inserts

ATRNX1	90%
ATRNX3	10%
TRNX1	few%

Phase 2
Batch Peaks

ATRNX2	7947K
ATRNX1	7291
ATRN	4256
TRNX1	2529

ATRNX1	8670K
ATRN	2628
TRNX1	1990
ATRNX3	995

Sync_Read
(IFCID 199)
surveys

REORG “shadows”

- Tablespace Partition(s)
- Index Partition(s)
- NPSI

Enough for worst-case simultaneous REORGs

Deploying SSD – SMS configuration

SHARE
Technology · Connections · Results

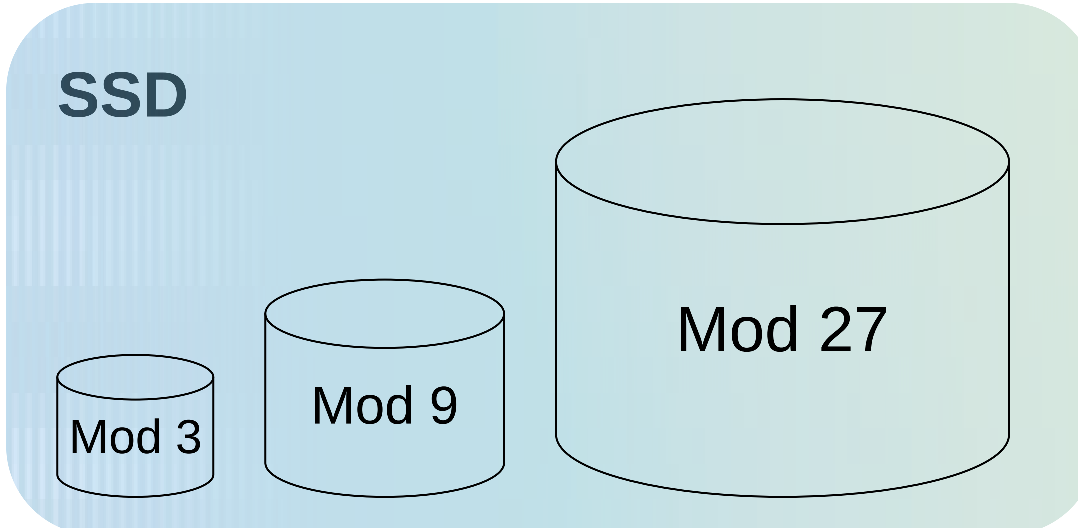
Virtual 3390

VOLSERs

(just like HDD)

D SMS,STORGRP(sg),LISTVOL

DCOLLECT



STORGRP

Automatic

Class

Selection

rules

STORCLAS

SSDsc

"SSDdsn"*

Inflexible

FILTLIST

DSNAME matching rules

Deploying SSD – Object by Object

```
CREATE STOGROUP SSDsg
   VOLUMES('*') VCAT vcatname
   STORCLAS SSDsc
```

SSD STOGROUP

ALTER object(s)

```
-STOP DB(db) SPACE(space)
  ALTER TABLESPACE db.ts  [PART p] STOGROUP SSDsg
  ALTER INDEX ixcr.ixname [PART p] STOGROUP SSDsg
-START DB(db) SPACE(space)
```

Specify PRIQTY, SECQTY to "fit"

REORG Shrlevel Change

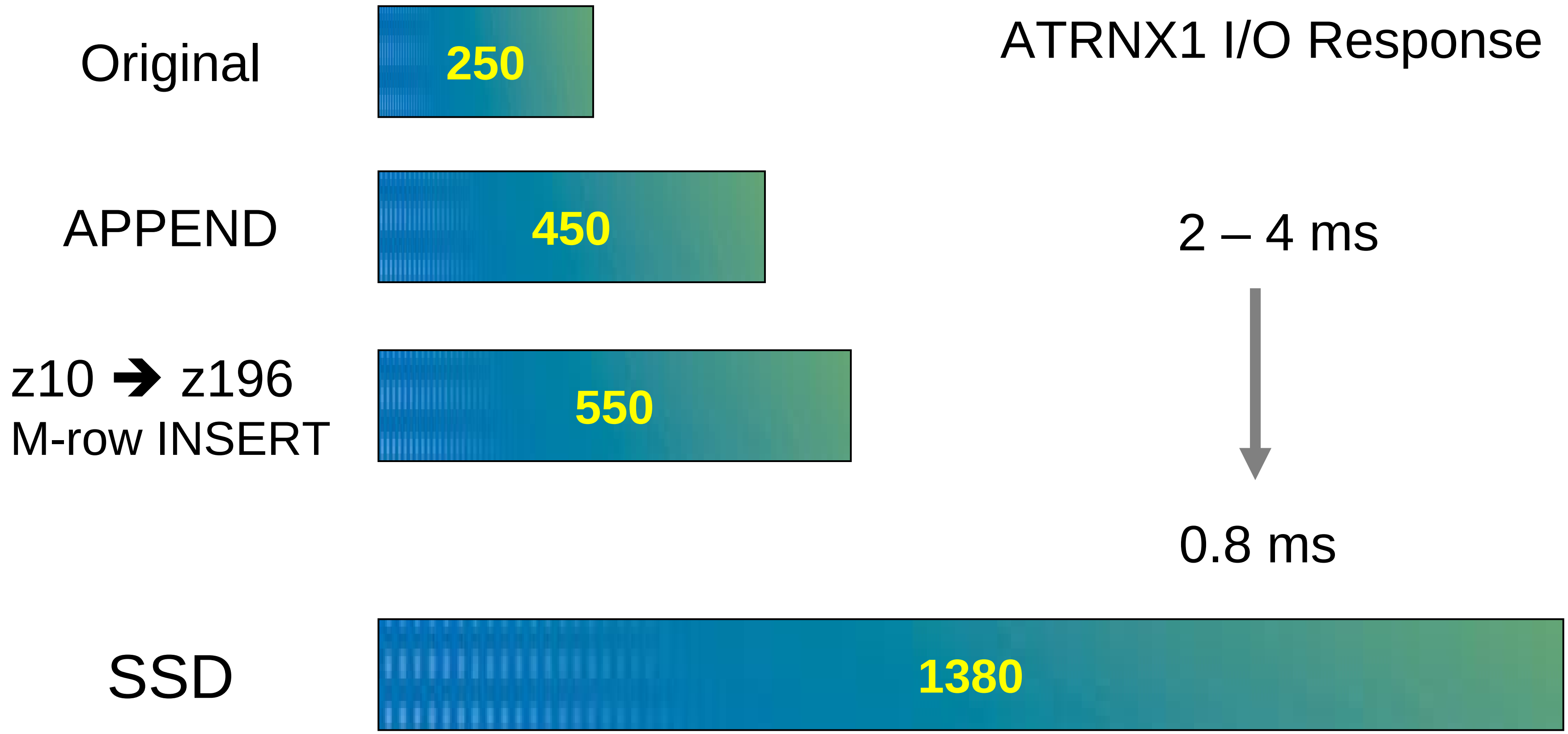
Shadow (⇨ object) allocated on SsdStorclas

DFSMSdss

Move

Faster, but requires STOP downtime

INSERT ATRN throughput

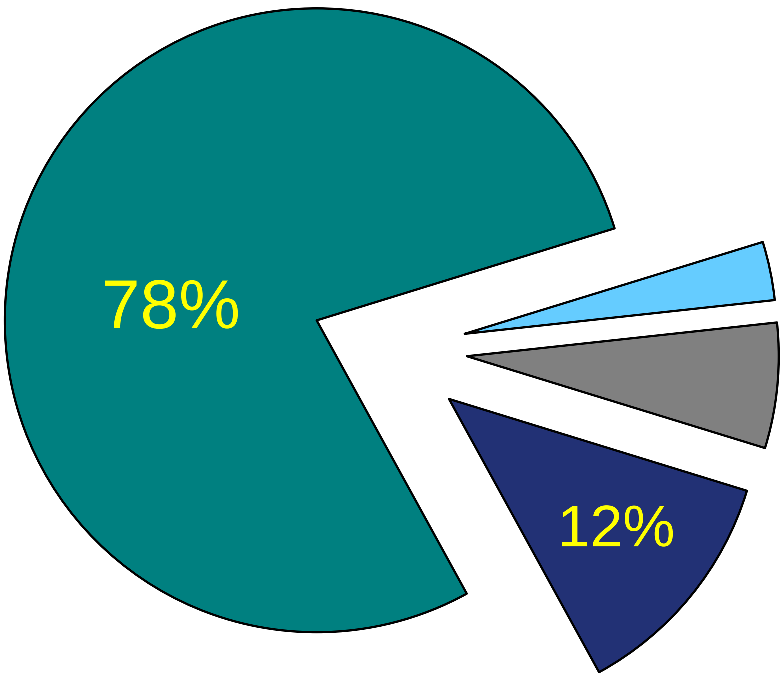


Case 2 – Mass INSERTs in OLTP

- Representative day – 4.6M transactions, 22.8M INSERTs
- Goal – Reduce DB2 Elapsed / (CICS) Transaction
- Especially given – an upcoming conversion of tables to DB2 from an older legacy database

DB2 Elapsed_Time Analysis (ACCT Class 3)

Sync_Read_Wt



Async_Read_Wt

Other | Lock / Latch
Data Sharing
Log Write
Syslgrng
. . .

CPU

More typically

65%

small

SMF 42-6 – DFSMS DASD Dataset I/O

MXG

	SMFTIME	(60 minute interval)
S42JDJNM	JOB	= 'db2xDBM1'
S42DSN S42DSVOL	**DSNAME** VOLSER	VSAM LLDS (TS/IX)
S42DSIOR	RESPTIME	RESPONSE TIME (MS)
S42DSCND S42DSWCN S42DSSHTS S42DSWHI calc	CACHCAND WRITCAND CACHHITS WRITHITS **RDHITPCT**	# CACHE CANDIDATES # WRITE CANDIDATES # CACHE HITS # WRITE HITS READ CACHE HIT %
S42DSION S42DSIOD	IOCOUNT AVGDISMS	# I/Os AVG I/O DISCONNECT
S42DSRDT S42DSRDD	**S42DSRDT** **S42DSRDD**	# READS AVG READ DISCONNECT

Isolate READ
(WRITE irrelevant)

Read + Write

Read =
Total - Write

Read + Write

OA25559 z/OS 1.8

Candidates – FLASHDA Service Aid

http://www-03.ibm.com/systems/z/os/zos/downloads/flashda.html

- VOLUMES with high disconnect — SMF 74-5
  - ✖ Move to SSD with zDMF and TDMF®
  - Ineffective, since datasets move with REORGs

- **DATASETS** with highest read disconnect — SMF 42-6

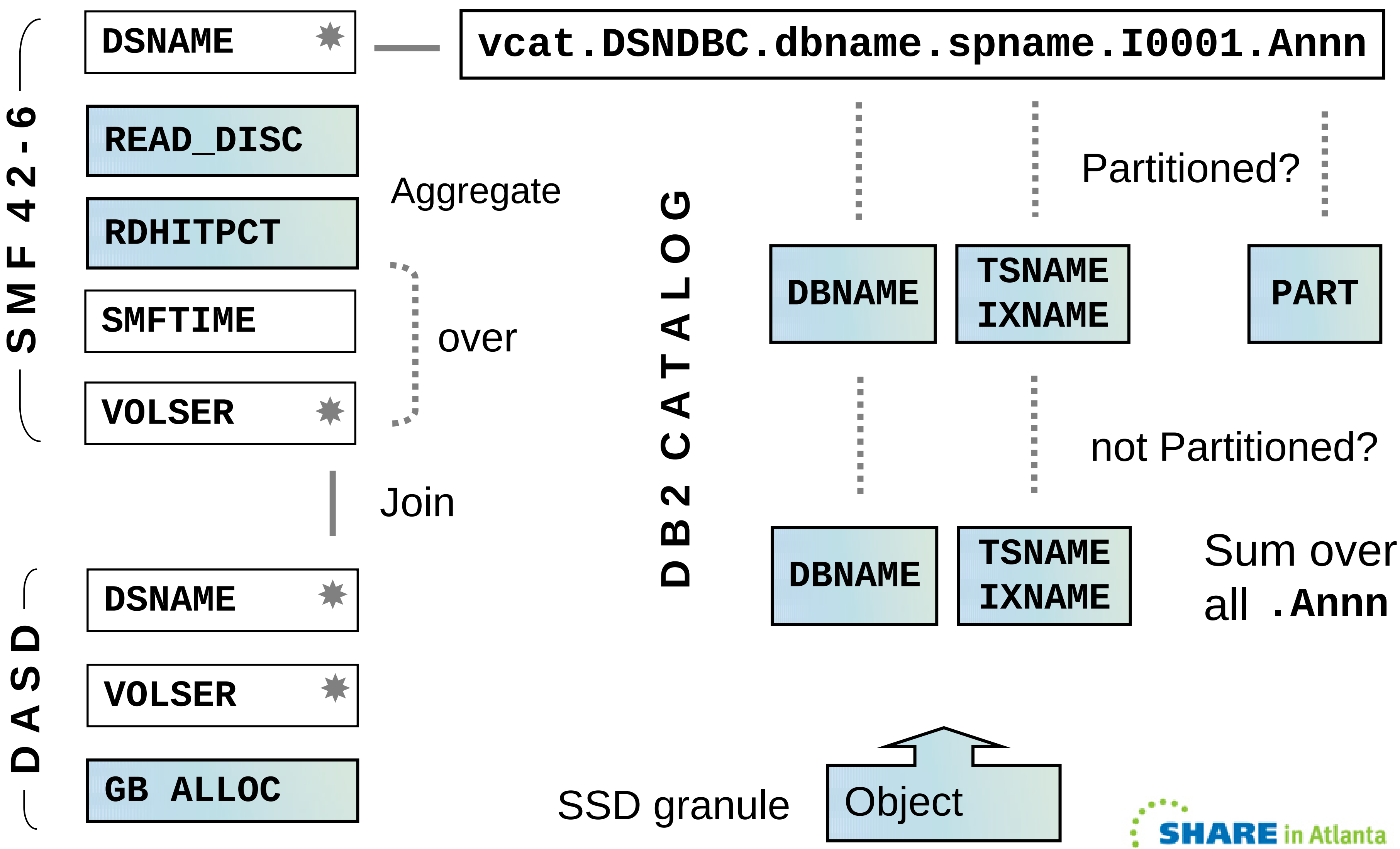
S42DSRDT	**#Reads**
S42DSRDD	**Avg. Read Disc**

X = **READ_DISC** aggregate

  - By DSNAME / VOLSER – not what DB2 can use

Survey SMF 42-6 READ_DISC

SAS / MXG



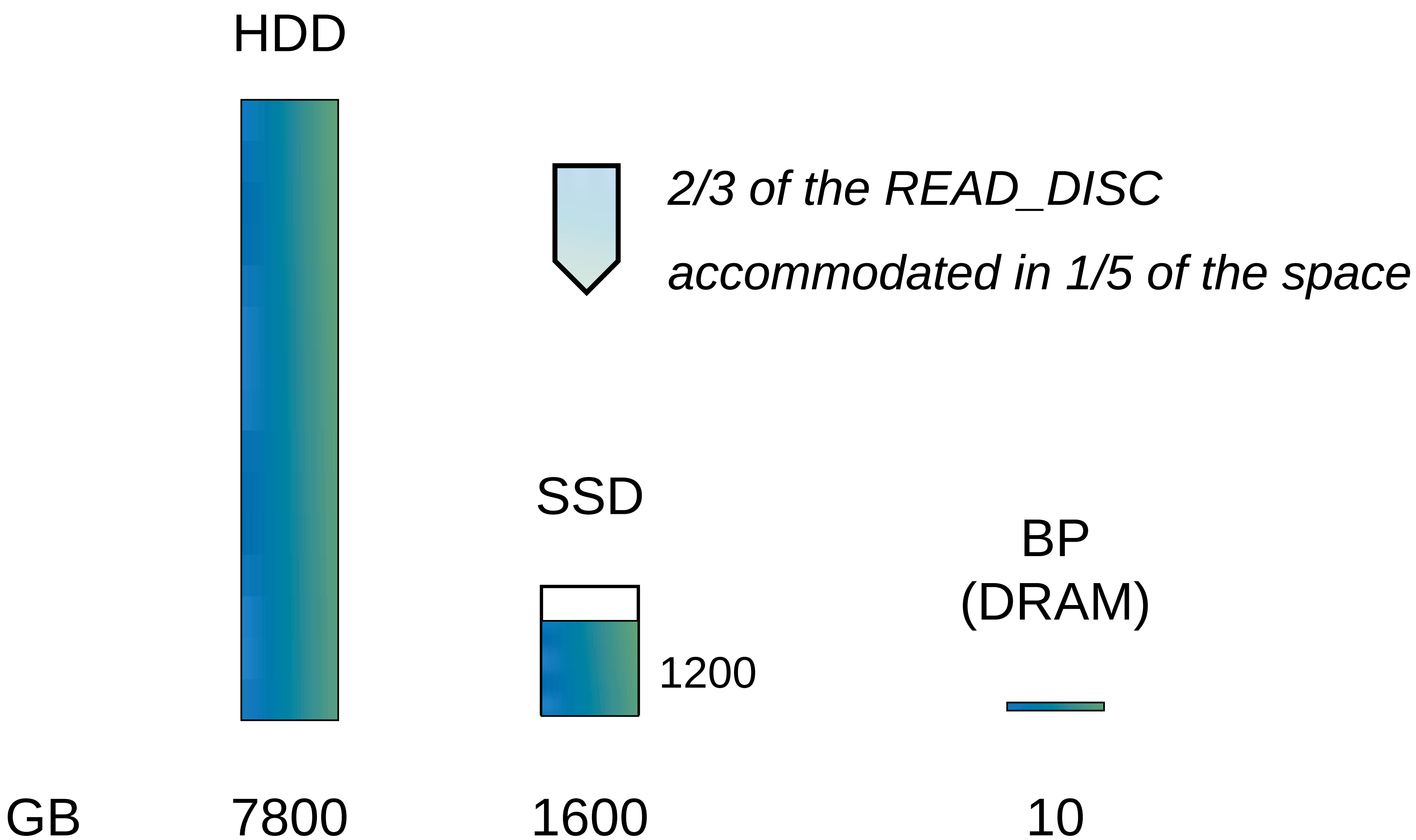
Highest READ_DISC survey – Results

TBNAME	IXNAME	PA RT	--Read Pct	Disc- Cum	Cache Read%	---GB %Space	Used--- Cum
TAB1	TAB1X2	-	3.4%	3.4%	29.8%	78.5	78.5
TAB2	TAB2X2	-	3.1%	6.5%	28.8%	16.8	95.2
TAB3	TAB3X2	-	2.8%	9.3%	27.3%	103.0	198.2
TAB3	TAB3X7	-	2.8%	12.2%	26.6%	115.2	313.4
TAB3	TAB3X3	-	2.8%	15.0%	27.2%	114.7	428.0
TAB3	TAB3X6	-	2.8%	17.8%	25.7%	103.0	531.0
TAB4	TAB4X3	-	2.7%	20.5%	25.2%	76.0	607.0
TAB4	TAB4X2	-	2.6%	23.1%	22.9%	70.4	677.4
TAB2	TAB2X1	-	2.5%	25.6%	28.4%	13.7	691.1
TAB2	TAB2X4	-	2.5%	28.1%	28.6%	11.4	702.5
TAB5	*TP*	1	1.5%	29.6%	58.7%	2.5	705.0
///							
TABN	TABNX2	-	0.2%	66.7%	36.2%	7.6	1204.6

65

Large INSERT indexes – Poor DASD Cache Hit % – High I/O response

Best use of SSD



Business case

- SSD order about to be finalized

- Reasonable forecast – 30% less DB2 elapsed
  - Sync_Read_Wt = 78% of elapsed time
  - Dominated by disconnect time (seek)
  - 2/3 of the disconnect time, 4ms response
  - … reduced to 0.8 ms

- Will measure post-install …

Futures … ?

- SSD cost continues to decline
  - Cost more closely approaches HDD
- Eventually displaces HDD entirely
- DB2 elapsed time down significantly overall
  - For Sync_Read_Wt – dominated workloads, like ours
- Shorter timeline vs. “data in memory”
  - DRAM still more expensive

SSD eliminates need to REORG?

- IBM direction to reduce REORGs
  - Have some CPU, media, baby-sitting cost
  - Cause some outages (not completely 24x7)
- DB2 10 relaxes REORG Clustering Criteria for SSD
  - SYSTABLESPACESTATS.DRIVETYPE = ‘HDD’ / ‘SSD’
  - Cluster order less important, with no Seek time
- Other REORG drivers still important
  - Index disorganization (faroff leaf splits)
  - Tablespace indirect references
  - Reclaim space
  - etc.

Using Solid-State Devices to Solve a DB2 INSERT Problem

Thank you!

Terry L. Berman

DST Systems, Inc

tlberman@dstsystems.com

Acknowledgements

Albert Weaver, DST

Ramu Nalluri, DST