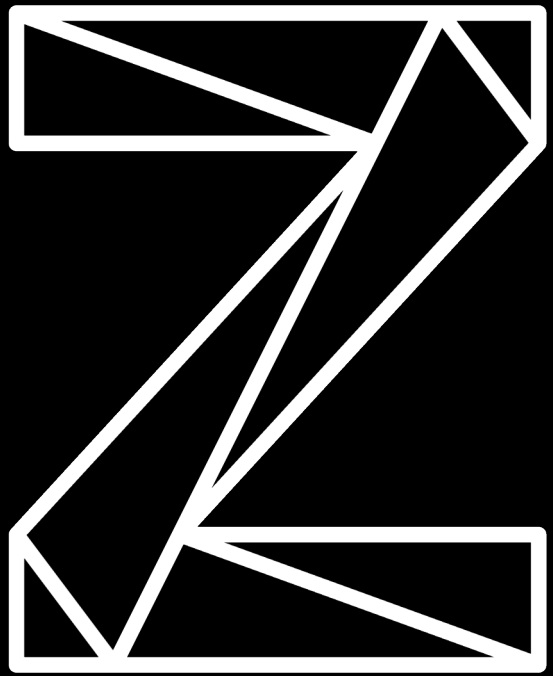
IBM Z Table Accelerator

Get the most out of your mainframe

Virtual West Coast Z Council
December 2nd 2021

Andrew Bowker
IBM Z Table Accelerator Product Manager
Andrew.bowker@ibm.com

Larry Strickland
Chief Product Officer, DataKinetics Ltd.
lstrickland@dkl.com
Authorized IBM Business Partner



Today’s presenters

Andrew Bowker

Product Manager

IBM

Larry Strickland

Chief Product Officer

DATAKINETICS
Z PERFORMANCE & OPTIMIZATION

Before we dive in

Let’s understand “fit” and “pain”

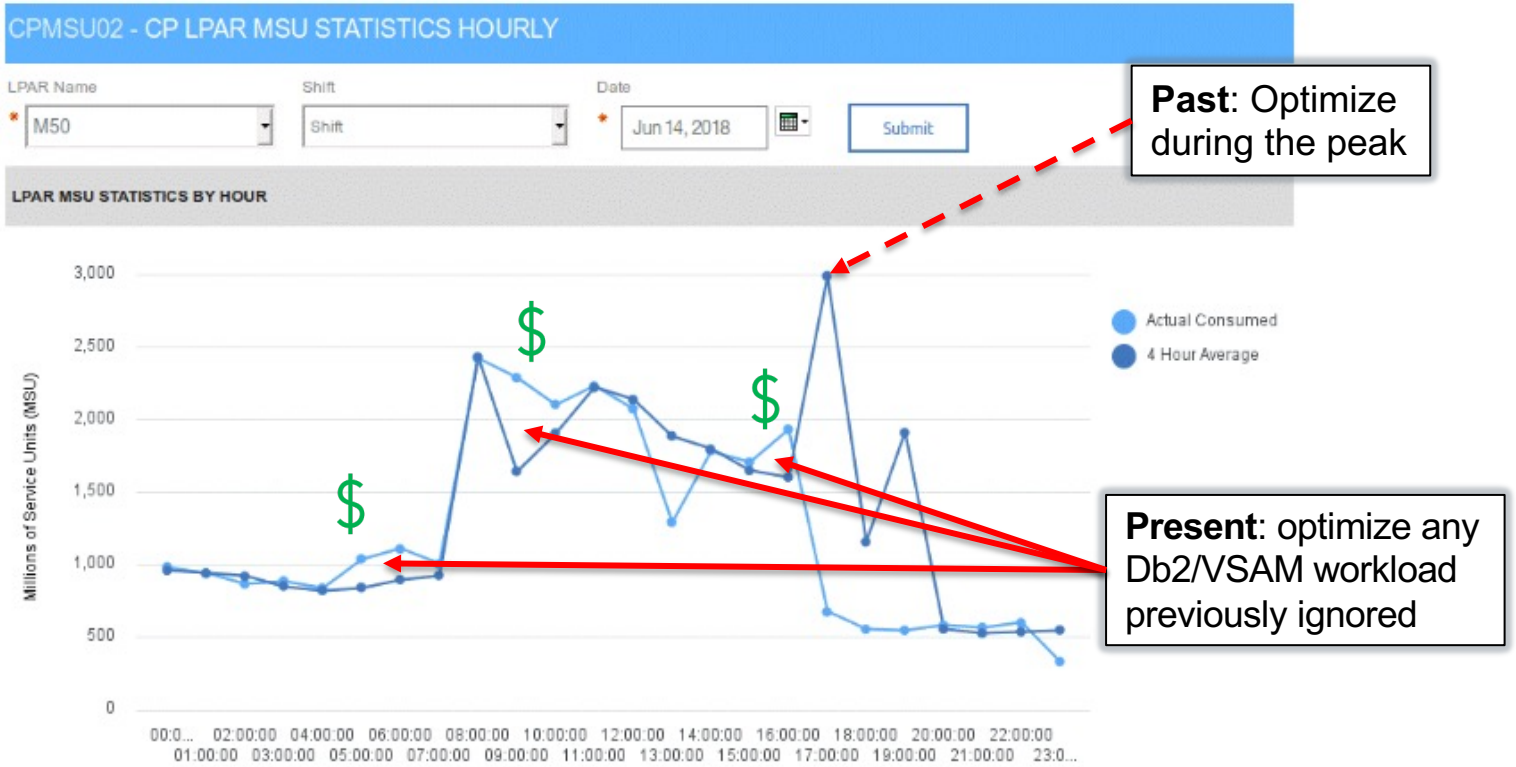
- [x] 1. Db2 and/or VSAM data sources?
- [x] 2. Highly (frequently) transactional workload?
- [x] 3. Goal to reduce cost?
- [x] 4. Fix batch window contention?
- [x] 5. Reduce online response times?
- [x] 6. Reduce consumption on Tailored Fit Pricing?

Agenda

- Tailored Fit Pricing unlocks opportunity
- Introduce IBM Z Table Accelerator
- 4 In-memory optimization examples:
  - Example 1: Insurance Co. - Db2 in R4HA
  - Example 2: Credit Card batch processing
  - Example 3: Temporary COBOL data
  - Example 4: Bank Co. - VSAM
- Finding candidates
- Click-through demo

New optimizations uncovered with Tailored Fit Pricing

- **Past**: Optimization only matters during R4HA peak
- **Present**: Optimization opportunities exist for all 720 hours in a month



IBM Z Table Accelerator

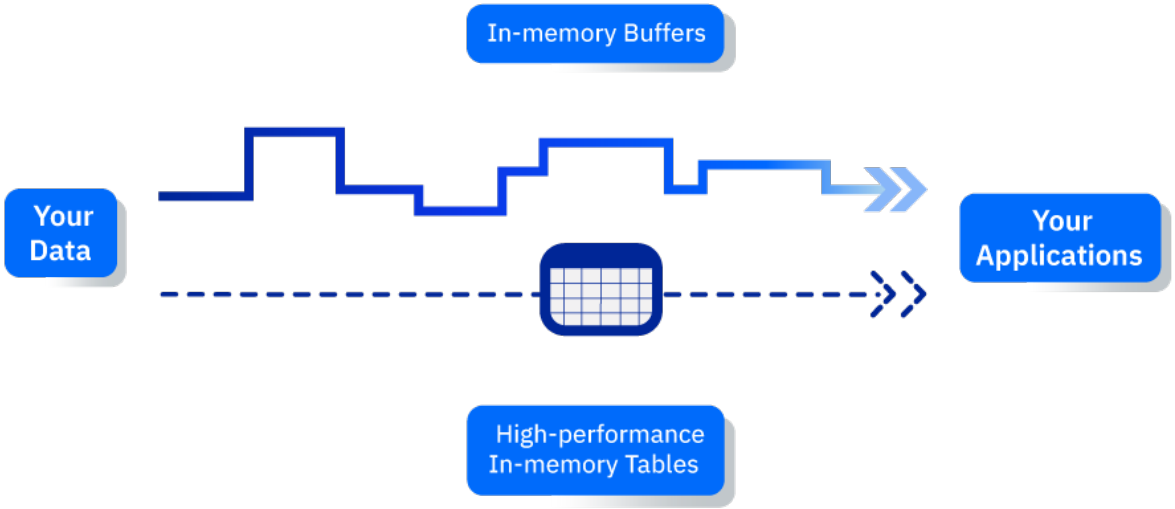
Reduce Resource Consumption

Reduce Elapsed Time

Reduce Operational Cost

Improve System Capacity

Reallocate Funding to Innovation



Example 1: Insurance Company with large Db2 workload in the peak window

Challenge:

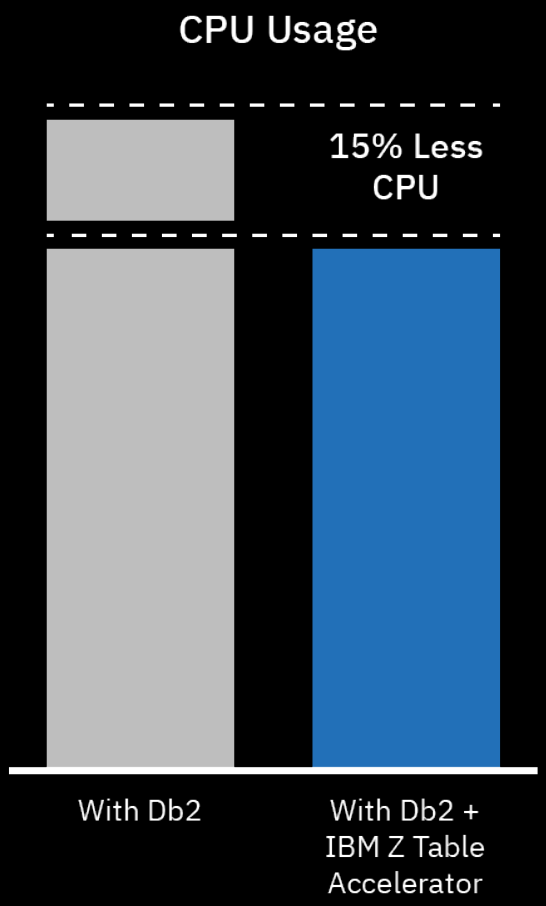
- One Db2 workload made up 25% of R4HA peak

Our Solution:

- Redirect problematic SQL statements to IBM Z Table Accelerator

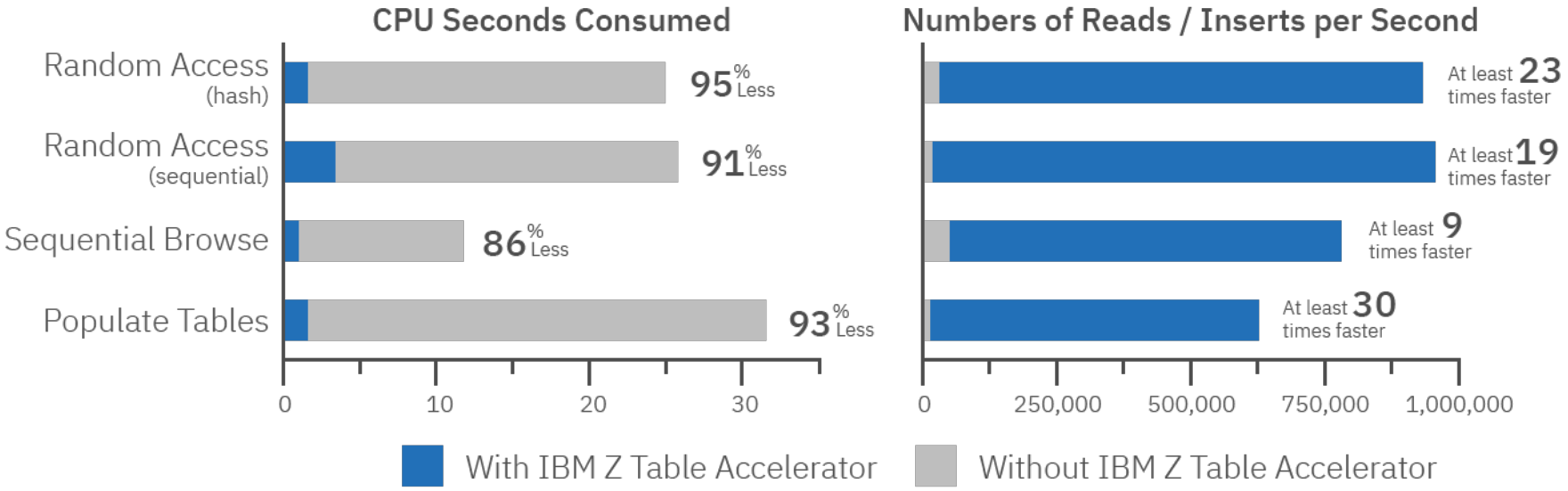
Results:

- 60% reduction in Db2 CPU
- 15% reduction in overall CPU consumption during R4HA peak

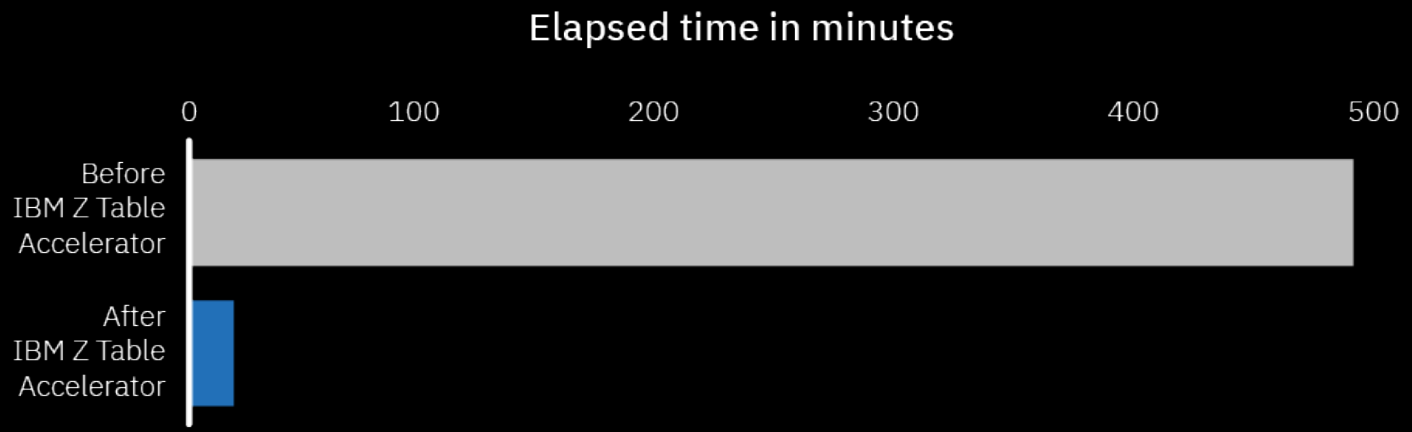


Actual IBM benchmark results for Db2

- Comparison of Db2 vs Db2 accelerated by IBM Z Table Accelerator
- No changes to Db2 systems; no changes to application logic



Example 2: Credit Card Db2 batch processing



Challenge:

- Reconciliation batch processing taking too long

Our Solution:

- Move a table describing the credit card options into IBM Z Table Accelerator in memory
- Each transaction required data from that table

Results:

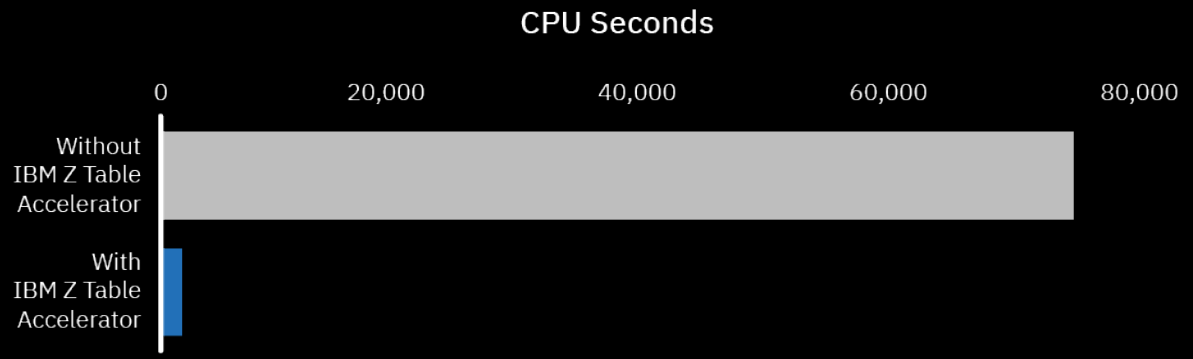
- 97% reduction in elapsed time
- Batch job that took 8 hours to complete now takes 15 min

Where is IBM Z Table Accelerator applicable?

Target is very frequently accessed mostly static reference data in Db2 tables and/or VSAM files for highly transactional workloads.

Use case	Benefit
Reference Data Tables - Batch	Reducing batch window contention and CPU resource
Reference Data Tables - Online	Reducing access time and CPU resource
Temporary Data Tables	Create, search, sort, read tables from various sources
Programmable Flexibility	Allows fast rules updates – in minutes/hours rather than weeks/months.

Example 3: Banking customer with temporary COBOL data



Challenge:

- A COBOL program was using an internal table and a binary search
- The search code was called 1.25 million times and had 4 searches in it
- Took over an hour of CPU to execute

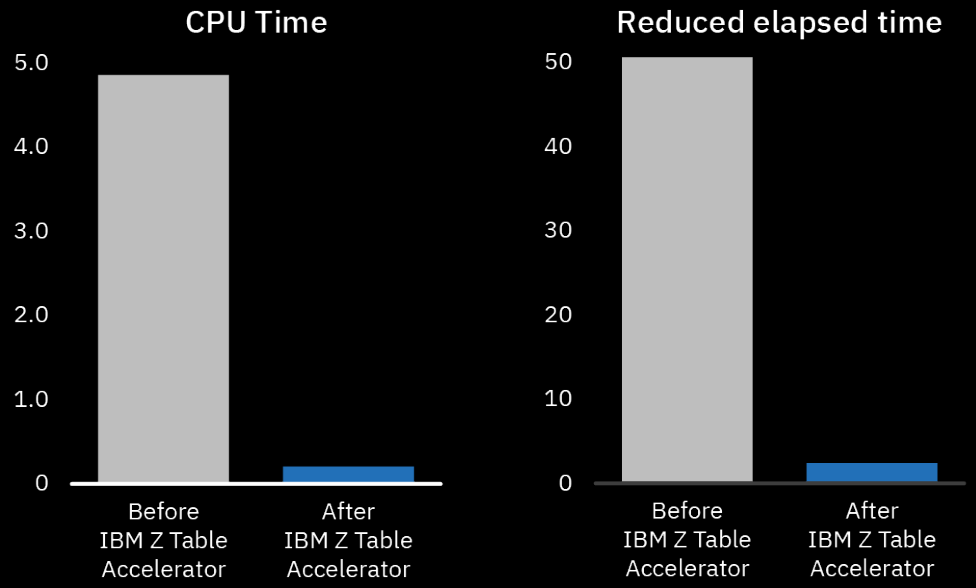
Our Solution:

- Replace the 4 searches with calls to IBM Z Table Accelerator

Results:

- 98% reduction in CPU seconds
- Now takes less than a minute to execute

Example 4: VSAM batch processing



Challenge:

- VSAM file high open/close and reads, with subsequent high CPU usage

Our Solution:

- Remove cost of open/close for the VSAM file
- Remove cost of Reads

Results:

- 93% Application CPU use Reduction
- 98% Application Elapsed Reduction
- >24 hours of CPU use reduction over a day.

Is this a good fit - Revisited?

If you can check these boxes…

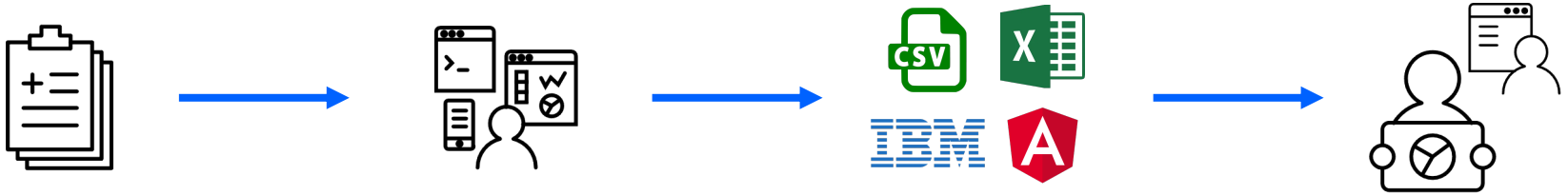
- ☑ 1. Db2 and/or VSAM data sources
- ☑ 2. Highly transactional workload
- ☑ 3. Goal to reduce cost, fix batch window contention or reduce online response times
- ☑ 4. Applications developed in-house

Then work with an IBM representative to …

Collect data from your environment

Analyze the best candidates

Next Steps



We'll send instructions

You collect some data

We'll analyze the data

We'll show you the best opportunities

Risk Free!

Finding Candidates

Finding the best candidates

DB NAME	TS NAME	SCHEMA	TBL NAME	UPD TIMESTAMP	#ACCESSES	#WRITES	SIZE(KB)	PRTS
D0350C0C	SBANC098	C0C	MS_BUFFER_REGISTRY	2016-03-02-21.35.45	7139099910	3345069	4320	1
D0350C0C	SBANC490	C0C	BATCH_JOB_EXECUTIO	2016-03-02-05.01.32	9012804070	792161	10080	1
D0387C1B	RTS9048	C1B	RULEPACKAGE	2016-02-01-01.32.10	4149470380	61070	7200	1
D0388C2A	TSTFTM01	C2A	OBJ_BASE	2016-03-02-21.35.45	3654218820	2743170	143280	1
D1419A65	S1419	DA65	T141901	2016-03-02-21.35.45	10868704000	32698	4320	1
D1484D2W	S4053	DB2W	T7997_CTL_ACT_REMT	2016-03-02-07.01.46	7498673940	3249003	10800	1
D1484D2W	SF721	DB2W	TE111_AMX_INI_CM_P	2016-03-02-21.35.45	736736632	1259310	7920	1
D1558AP1	S5962	APPC1	T5962_FS_ALIAS_ASC	2016-03-02-21.35.45	1.31756E+11	2080948	612000	1
D1961A65	S1961	DA65	T196101	2015-05-14-17.05.25	1355457040	1257066	639360	1
D2075A65	S2075	DA65	T207501	2016-03-02-21.35.45	2247222820	4798	2160	1
D2120M0B	S2120	DB2W	T2120_SEAS_RECOMM	2016-03-02-16.05.31	3154674230	26397712	313920	1
D2138M0B								
D2139M0B								

Data Set Name	EXCPS	INSERTS	DELETS	UPDATES	RETRIEVALS	RECORDS CHANGE	# RECORDS	LRECL	GB
PRDFE.#LPS.APOR.DATA	24,883,173	1,123	940	41,308	121,193,241	205	52,946	600	0.029586
PRDI9.#P1BHI.N1ZXFDR.DATA	28,821,49,	0	0	0	114,137,907	6	231,896	4089	0.883101
PRDXW.#ENTRDX.VPF.DK	2,319,082	95,220	163,653	70	104,925,834	-68,433	585,381	136	0.074144
PRDI9.#P1BHI.PE01RCR.X00001.DATA	26,092,817	10,561	2	0	92,958,864	10,559	2,952,090	18	0.049488
PRDI9.#P1BHI.PG1SIDF.X00021.DATA	32,675,941	97,947	73,731	0	88,488,668	24,216	6,452,284	38	0.228348
PRDI9.#P1BHI.E10FRPM.A00015.DATA	3,180,727	0	0	0	68,269,467	82	130,461	8185	0.994488
PRDI9.#P1BHI.E104CIF.A00004.DATA	5,942,502	0	0	0	63,679,458	4	173,605	4089	0.661119
PDRI9.#P1BHI.N1ZXFDRX.DATA	24,330,441	17,670	19,696	0	61,776,087	-2,026	4,364,814	40	0.162602
PDRI9.#P1BHI.E103CIF.A00003.DATA	4,433,407	0	0	0	60,130,646	5	124,992	4089	0.475992
PRDI9.#P1BHI.N105AFS.A00005.DATA	12,132,415	0	0	0	54,528,900	4	156,271	4089	0.595108
PRDE6.OS.PAYEE.INDEX.CICS.DATA	79,206	0	0	0	46,879,875	0	147,422	100	0.01373
PRDI9.#P1BHI.PG0WAIF.X00032.DATA	11,813,441	86,676	0	0	42,542,171	86,676	2,882,518	34	0.091275
PRDI9.#P1BHI.N113PAC.A00039.DATA	4,467,981	0	0	0	35,375,809	84	87,734	10233	0.836125
PRDI9.#P1BH									
PRDI9.#P1BH									

UID	March 3 to March 8			March 8 to March 11			March 11 to March 31			March 3 to March 31		
	Time	Read Rate	Write Rate	Time	Read Rate	Write Rate	Time	Read Rate	Write Rate	Time	Read Rate	Write Rate
DEODE102.PGTS0070.PEORE102.PR4_RULE_VW	450536	1308.26	0.000109	238321	4329.061	0.000533	1756273	3941.427	0.000476	2443130	3493.342	0.000414
D3986MUS.S3986A.MUS.T398601	454159	819.8802	0	257932	2161.681	0	1722037	1343.594	0	2434128	1332.568	0
D7099MTC.SH627.MTC.TF485_WRKR_CRT_SIG	151553	37.31155	0.125877	560541	0.00467	0.000168	1566068	1880.805	0.478344	2278162	1295.398	0.337242
DD927MK3.SD927.MK3.TD927_PTV_NM	68665	0.000189	0.000175	677666	0.257257	7.38E-06	1635545	1491.696	-0.11282	2381876	1024.365	-0.07745
D3643MBA.PGTSP039.MBA.PR_SYS_CACHE_DEP	463186	403.2476	0.002096	268735	1060.842	0.002192	1700413	881.9157	0.002304	2432334	810.5322	0.002252
D2E50M0B.S2E50.DB2W.T2E50_COMMUN_DATA	464988	1.312772	0	268735	-695.705	2.315091	1696803	1200.451	0.000575	2430526	761.39	0.256373
D2075A65.S2075.DA65.T207501	463186	686.0769	0.010877	268736	689.4758	0.039976	1701175	676.5437	0.108188	2433097	679.7869	0.082129
D1419A65.S1419.DA65.T141901	463186	678.9726	0.00025	268735	682.5051	0.000301	1701176	669.4204	0.000287	2433097	672.8841	0.000282
DEODE102.PGTS0170.PEODE102.PR_SYS_STATUSDETAI	463186	476.531	0.033593	268736	523.5691	0.033312	1701175	654.8009	0.033082	2433097	606.3692	0.033205
D4259MUS.SZUPT.MUS.TA985_CA_SMRY	463186	348.1611	0.012019	268736	319.9028	0	1701175	693.703	0.409171	2433097	586.6363	0.288373

- Using DB2 Stats Query and/or VSAM SMF64
- Collect Stats on Multiple Dates
- Process to find read/write rates

Finding VSAM Optimization Candidates

1. Collect SMF64
    - Extract days of interest SMF64 records to data set
    - Terse dataset
    - FTP to IBM

2. Extract Data to .csv

3. Analysis of SMF64 Data for low hanging fruit
    - High reads (c.f. number of records)
    - Low inserts/updates
    - Small file (<2 GB)
    - Bonus High Open/Close rate

4. Review Candidates

Collect SMF Data

Extract the SMF 64 Records to a data set

Run this job to extract SMF 64 records to a data set. Can be one day or 2 day's data. Preferably during a busy period, such as month end or other busy period.

```
//STEP1    EXEC PGM=IFASMFDP
//IN       DD DISP=SHR,DSN="SMF data sets"
//OUT      DD DSN=xxxxx.SMFDUMP.SMF64,DISP=(,CATLG),
//   UNIT=DISK,SPACE=(CYL,(400,100),RLSE)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  INDD(IN,OPTIONS(DUMP))
  OUTDD(OUT,TYPE(64))
```

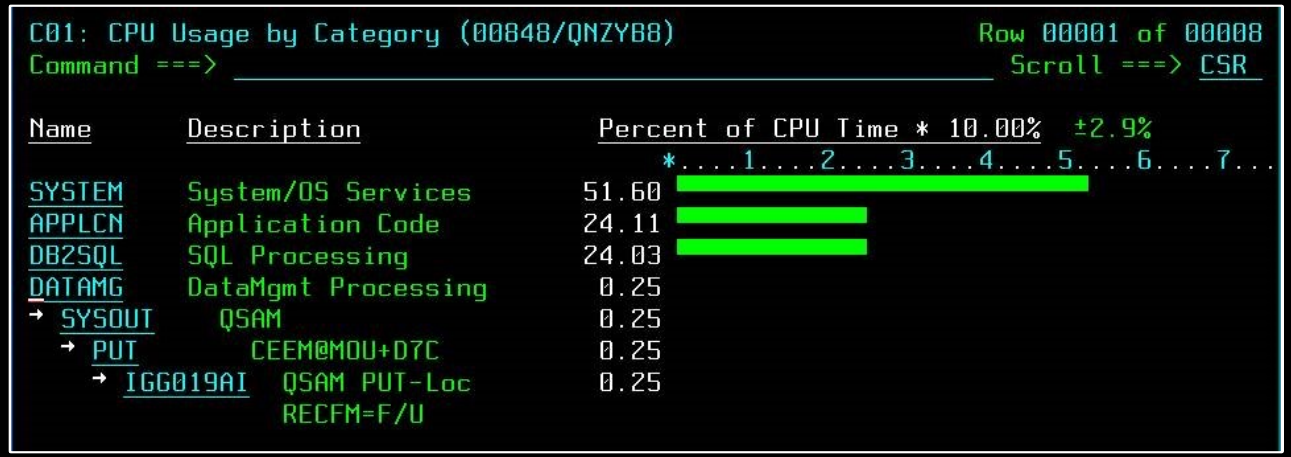
TERSE the data set.

Run this job to terse the file created in previous job.

```
//*************************************************************
//* JCL TO UNTERSE A FILE COMPRESSED WITH IBM'S TRSMAIN
//*************************************************************
//STEP01   EXEC PGM=TRSMAIN,PARM='PACK'
//SYSPRINT DD SYSOUT=*
//INFILE   DD DISP=SHR,DSN= xxxxx.SMFDUMP.SMF64
//OUTFILE  DD DSN= xxxxx.SMFDUMP.SMF64.TRS,
//            DISP=(,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,20),RLSE)
```

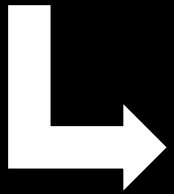

Application Profile

Application Profile (eg APA) - can provided detailed view



Program Modification

```
LOOKUP-BANK.
    MOVE IN-CCARD-BIN TO BIN-CODE.
    READ BINCODE.
    EVALUATE FS-CODE
        WHEN "00"
            MOVE BIN-BANK TO OUT-BANK
        WHEN "23"
            MOVE "**** INVALID BIN CODE ****" TO OUT-BANK
        WHEN OTHER
            PERFORM VSAM-CODE-DISPLAY
    END-EVALUATE.
```



```
LOOKUP-BANK.
    MOVE 'BINCODE' TO TA-TABLE
    MOVE 'FK' TO TA-COMMAND
    CALL "DKJTCALL" USING TA-PARM TA-COMMAND-AREA BIN-REC.
    IF TA-ERROR = 0 THEN
        IF TA-WAS-FOUND
            MOVE BIN-BANK TO OUT-BANK
        ELSE
            MOVE "**** INVALID BIN CODE ****" TO OUT-BANK
        END-IF
    ELSE
        GO TO ZTA-ERROR
    END-IF.
```


Results

Use Case	CPU save %	Elapsed Time Save %	Notes
1	75%	78%	Single job
2	89%	34%	Multiple jobs – same VSAM file
3	54%	81%	Jobs with Easytrieve
4	0%	48%	Single job
5	95%	99%	Very high open/close rate (1000's/second) - total was 24 hours of CPU saved in a day

Finding Db2 Optimization Candidates

1. Collect dB2 statistics data
   Collect multiple samples / day over a period of interest
   (essential multiple are collected due to cumulative nature of collected statistics)

2. Send to IBM

3. Visualization of Data
   High read rate
   Low inserts/updates rate
   Small table size (<2 GB)

4. Review Candidates

Collecting Dat - Meta data only

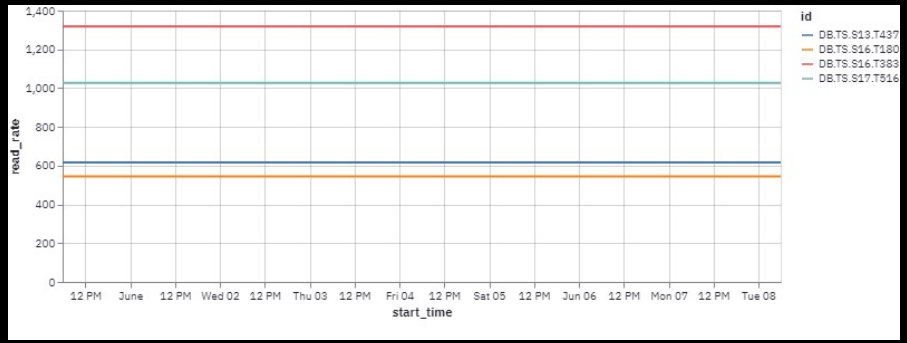
See "s2.txt"

sample1.txt | s2.txt

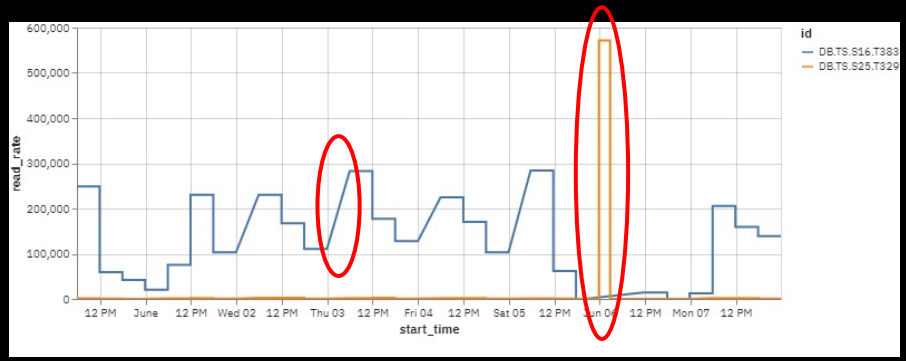
```
DB NAME      TS NAME       SCHEMA   TBLNAME UPD TIMESTAMP             #ACCESSES
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S10      T94      2021-05-31-00.00.40     23911    0    7200     1
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S10      T94      2021-05-31-00.00.40     23911    0    7200     1
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S10      T94      2021-05-31-00.00.40     23911    0    7200     1
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S10      T94      2021-05-31-00.00.40     23911    0    7200     1
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S10      T94      2021-05-31-00.00.40     23911    0    7200     1
DB   TS   S10      T108     2021-05-31-00.00.40     894 0    7200     1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
DB   TS   S17      T574     2021-05-31-00.01.30     314401  2874     7200     1
DB   TS   S17      T266     2021-05-31-00.01.30     111 0    1898640 1
DB   TS   S16      T571     2021-05-31-00.01.30     2638058 120 720 1
```

Names changed to protect the innocent

Why multiple samples per day?



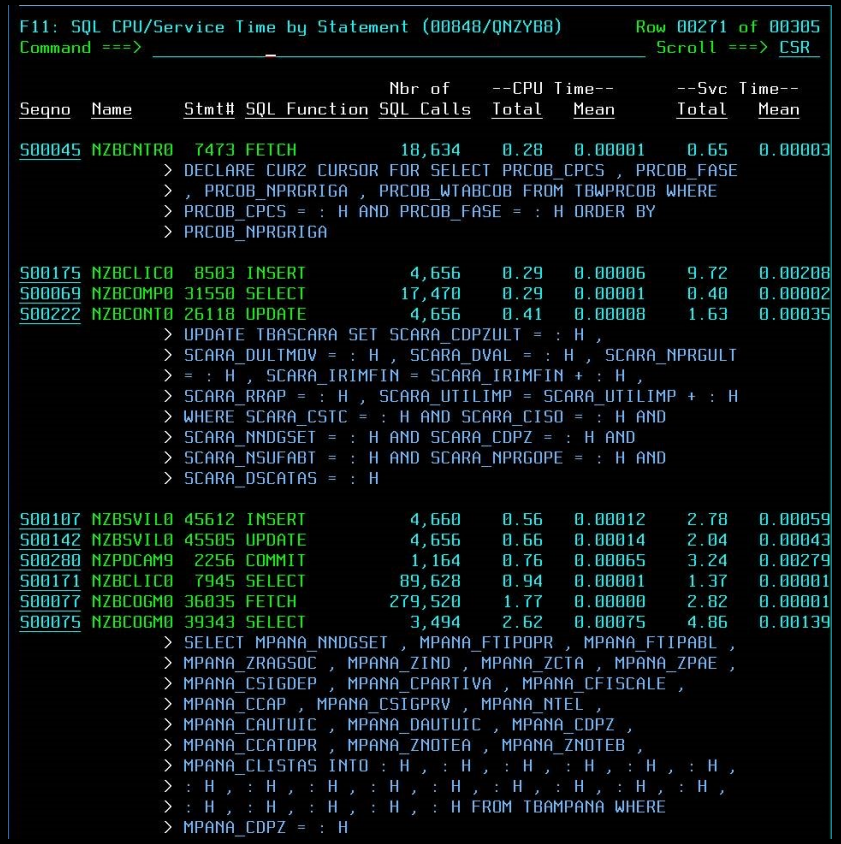
1. Miss spikes



2. Miss reorgs

Application Profile

Application Profile Report
- Details by SQL Statement
- Note – Db2 processing time only



```
F11: SQL CPU/Service Time by Statement (00848/QNZYB8)        Row 00271 of 00305
Command ===> _____                                          Scroll ===> CSR

                                        Nbr of     --CPU Time--       --Svc Time--
Seqno  Name     Stmt# SQL Function     SQL Calls   Total    Mean       Total    Mean

S00045 NZBCNTR0  7473 FETCH             18,634     0.28   0.00001     0.65   0.00003
             > DECLARE CUR2 CURSOR FOR SELECT PRCOB_CPCS , PRCOB_FASE
             > , PRCOB_NPRGRIGA , PRCOB_WTABCOB FROM TBWPRCOB WHERE
             > PRCOB_CPCS = : H AND PRCOB_FASE = : H ORDER BY
             > PRCOB_NPRGRIGA

S00175 NZBCLIC0  8503 INSERT           4,656     0.29   0.00006     9.72   0.00208
S00069 NZBCOMP0 31550 SELECT          17,470     0.29   0.00001     0.40   0.00002
S00222 NZBCONT0 26118 UPDATE           4,656     0.41   0.00008     1.63   0.00035
             > UPDATE TBASCARA SET SCARA_CDPZULT = : H ,
             > SCARA_DULTMOV = : H , SCARA_DVAL = : H , SCARA_NPRGULT
             > = : H , SCARA_IRIMFIN = SCARA_IRIMFIN + : H ,
             > SCARA_RRAP = : H , SCARA_UTILIMP = SCARA_UTILIMP + : H
             > WHERE SCARA_CSTC = : H AND SCARA_CISO = : H AND
             > SCARA_NNDGSET = : H AND SCARA_CDPZ = : H AND
             > SCARA_NSUFABT = : H AND SCARA_NPRGOPE = : H AND
             > SCARA_DSCATAS = : H

S00107 NZBSVIL0 45612 INSERT           4,660     0.56   0.00012     2.78   0.00059
S00142 NZBSVIL0 45505 UPDATE           4,656     0.66   0.00014     2.04   0.00043
S00280 NZPDCAM9  2256 COMMIT           1,164     0.76   0.00065     3.24   0.00279
S00171 NZBCLIC0  7945 SELECT          89,628     0.94   0.00001     1.37   0.00001
S00077 NZBCOGM0 36035 FETCH          279,520     1.77   0.00000     2.82   0.00001
S00075 NZBCOGM0 39343 SELECT           3,494     2.62   0.00075     4.86   0.00139
             > SELECT MPANA_NNDGSET , MPANA_FTIPOPR , MPANA_FTIPABL ,
             > MPANA_ZRAGSOC , MPANA_ZIND , MPANA_ZCTA , MPANA_ZPAE ,
             > MPANA_CSIGDEP , MPANA_CPARTIVA , MPANA_CFISCALE ,
             > MPANA_CCAP , MPANA_CSIGPRV , MPANA_NTEL ,
             > MPANA_CAUTUIC , MPANA_DAUTUIC , MPANA_CDPZ ,
             > MPANA_CCATOPR , MPANA_ZNOTEA , MPANA_ZNOTEB ,
             > MPANA_CLISTAS INTO : H , : H , : H , : H , : H , : H ,
             > : H , : H , : H , : H , : H , : H , : H , : H , : H ,
             > : H , : H , : H , : H , : H FROM TBAMPANA WHERE
             > MPANA_CDPZ = : H
```


Application Changes

```
EXEC SQL DECLARE CUR01 CURSOR FOR
    SELECT A.ACCT_NBR
    FROM DKLDB001.USB_ACCOUNT A,
        DKLDB001.USB_PRODUCT P
    WHERE A.CLNT_ID = :W-CLNT-ID
        AND A.BNK_NBR = :W-BNK-NBR
        AND A.AGT_NBR = :W-AGT-NBR
        AND A.PRODUCT_ID = P.PRODUCT_ID
        AND A.BNK_NBR = P.BNK_NBR
        AND P.CARD_TYP_CDE = :W-CARD-TYP-CDE
        FOR FETCH ONLY
END-EXEC.
```

```
EXEC SQL DECLARE CUR01 CURSOR FOR
    SELECT ACCT_NBR, CLNT_ID, BNK_NBR, PRODUCT_ID
    FROM DKLDB001.USB_ACCOUNT
    WHERE CLNT_ID = :L-CLNT-ID
        AND BNK_NBR = :L-BNK-NBR
        AND AGT_NBR = :L-AGT-NBR
    FOR FETCH ONLY
END-EXEC.
************************************************
MOVE PRODUCT-ID      TO  ITZA-PRODUCT-ID.
MOVE L-BNK-NBR       TO  ITZA-BNK-NBR.
MOVE L-CARD-TYP-CDE  TO  ITZA-CARD-TYP-CDE.
CALL 'ITZA'     USING  W-ITZA-PARM
                       W-ITZA-COMMAND-AREA
                       ITZA-PRODUCT-REC.
```


Results

Use Case	CPU save %	Elapsed Time Save %	Notes
1	96%	98%	Application to retrieve bank branch details for inter-branch transfers (20 tables involved)
2	86%	97%	Reconciliation Batch Job, credit card processing
3	60%	Not measured	Healthcare batch job (representing 15% of R4HA!)
4	15%	Not measured	CICS transaction – management of web session data inter-region

Demo / Example

VSAM example at a large bank:

- IBM Z Performance and Capacity Analytics
- IBM Z Table Accelerator
- IBM Application Performance Analyzer

Example scenario

Application Optimization at a large national bank

Zach identifies the top MSU-consuming applications with **IBM Z Performance and Capacity Analytics**

Systems Programmer

Verify

Detect

Stan and Sagar use **IBM Z Table Accelerator** to optimize the applications. Result: Better performance & lower cost

Fix

Middleware Systems Programmer

Isolate

Sagar analyzes a subset of DB2 and VSAM applications

Enterprise Architect

Diagnose

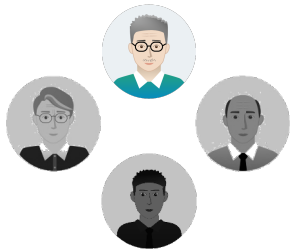
Julian uses application tuning tools to understand the business case

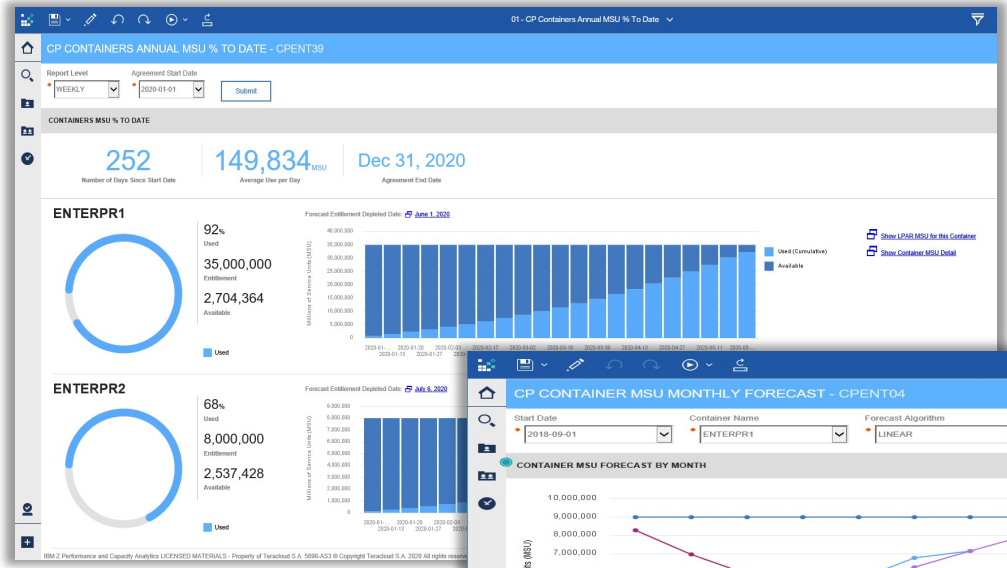
Subsystem Specialist

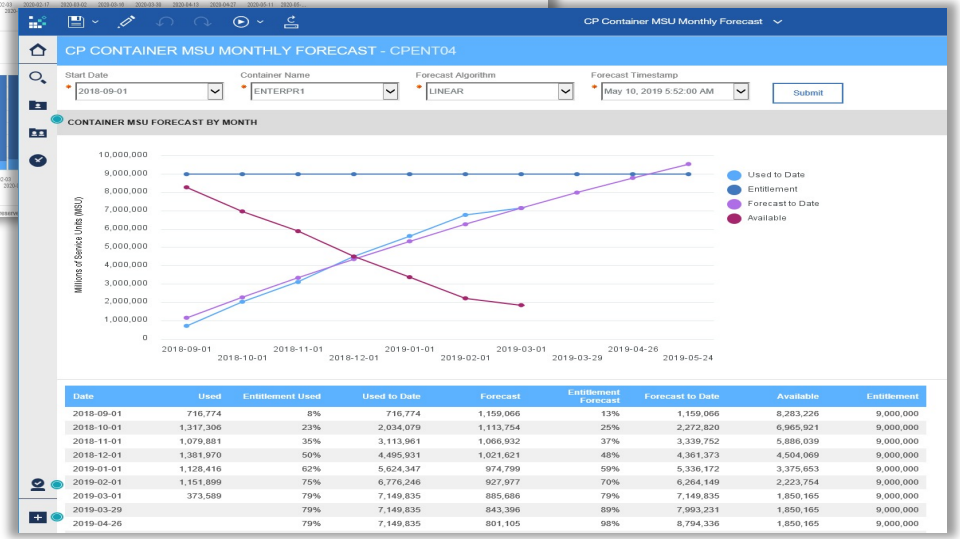
Example > Application Optimization at a large national bank

Zach tracks top MSU consumers in **IBM Z Performance and Capacity Analytics**

What can he do to address this trend? Is there any optimization that could be done?



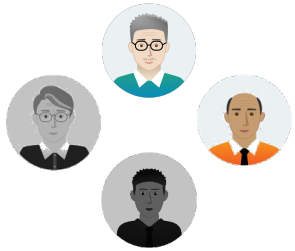




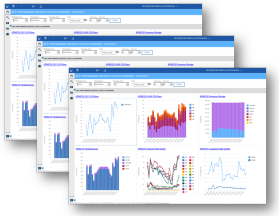
Date	Used	Entitlement Used	Used to Date	Forecast	Entitlement Forecast	Forecast to Date	Available	Entitlement
2018-09-01	716,774	8%	716,774	1,159,066	13%	1,159,066	8,283,226	9,000,000
2018-10-01	1,317,306	23%	2,034,079	1,113,754	25%	2,272,820	6,965,921	9,000,000
2018-11-01	1,079,881	35%	3,113,961	1,066,932	37%	3,339,752	5,886,039	9,000,000
2018-12-01	1,381,970	50%	4,495,931	1,021,621	48%	4,361,373	4,504,069	9,000,000
2019-01-01	1,128,416	62%	5,624,347	974,799	59%	5,336,172	3,375,653	9,000,000
2019-02-01	1,151,899	75%	6,776,246	927,977	70%	6,264,149	2,223,754	9,000,000
2019-03-01	373,589	79%	7,149,835	885,686	79%	7,149,835	1,850,165	9,000,000
2019-03-29		79%	7,149,835	843,396	89%	7,993,231	1,850,165	9,000,000
2019-04-26		79%	7,149,835	801,105	98%	8,794,336	1,850,165	9,000,000

Example > Application Optimization at a large national bank

Zach passes this info along to the enterprise architect to see if any optimizations can be done.



Zach, Systems Programmer



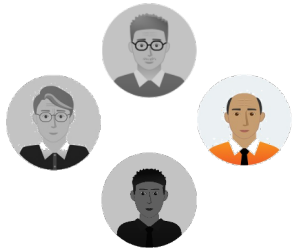
Sagar, Enterprise Architect

Example > Application Optimization at a large national bank

Sagar, the enterprise architect obtains the data collection package scripts

He collects meta data from Db2 and VSAM workloads during normal operations

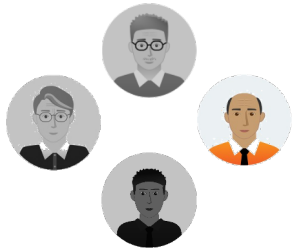
Once collected, he sends to IBM for analysis



Example > Application Optimization at a large national bank

He looks for applications with high read-rates, inserts, and deletes

Sagar identifies a BIN-code reconciliation application as a part of VSAM batch processing.



	End Time	Reads	Updates	Start Time	Reads	Updates	Time	Read Rate (/second)	Update Rate (/second)	Ave. MIPS (low)	Ave MIPS (High)	Size (GB)
LC	2020-07-14-17.38.16	4.94989E+11	9271217	2020-07-15-05.38.18	4.9793E+11	9304693	43202	68129	0.7749	195	681	1.545
DC	2020-07-14-17.38.16	4724071150	2240782	2020-07-15-05.38.18	4739186520	2250968	43202	350	0.2358	1	3	1.099
PC	2020-07-14-17.38.16	6188252870	4069361	2020-07-15-05.38.18	6209383930	4088423	43202	489	0.4412	1	5	1.030
DT	2020-07-14-17.38.16	433974382	2039462	2020-07-15-05.38.18	434493584	2044614	43202	12	0.1193	0	0	0.858
Z	2020-07-14-17.38.16	1454506810	1462774	2020-07-15-05.38.18	1456747330	1469510	43202	52	0.1559	0	1	1.305
M	2020-07-14-17.38.16	100437293	843493	2020-07-15-05.38.18	100442956	843722	43202	0	0.0053	0	0	0.275
Z	2020-07-14-17.38.16	1141779240	1930567	2020-07-15-05.38.18	1143902800	1942139	43202	49	0.2679	0	0	0.330
C	2020-07-14-17.38.16	884586718	1935292	2020-07-15-03.38.18	888290487	1942819	36002	103	0.2091	0	1	0.447

Input: Sequential Dataset 1M records

DATE	TRANID	CREDIT-CARD-NO
20201030	0600829586	548652 1700507212..
20210609	4866975324	443438 7464089420..
20200823	2001954658	544748 5986576348..
20201124	0227228291	497063 3615477450..
20201011	7189460642	436700 7686442531..

Output: Sequential Dataset 1M records

DATE	TRANID	CREDIT-CARD-NO	BANK-NAME
20201030	0600829586	548652 1700507212..	Banco de Chile Master Card Credit Card
20210609	4866975324	443438 7464089420..	Credit Union Australia - Visa Debit Card
20200823	2001954658	544748 5986576348..	Chase SLATE MasterCard Credit Card
20201124	0227228291	497063 3615477450..	La banque postale visa (France)
20201011	7189460642	436700 7686442531..	China Construction Bank Credit Card

Example > Application Optimization at a large national bank

Julian, a VSAM specialist runs a series of tuning reports to gain better understanding of potential MSU savings.

Julian creates a business case of estimated savings

```
C01: CPU Usage by Category (00848/QNZYB8)                Row 00001 of 00008
Command ===>                                              Scroll ===> CSR

Name      Description                Percent of CPU Time * 10.00%  ±2.9%
                                       *....1....2....3....4....5....6....7...
SYSTEM    System/OS Services      51.60
APPLCN    Application Code        24.11
DB2SQL    SQL P
DATAMG    DataM
→ SYSOUT    QSA
  → PUT       C
    → IGG019AI
```

```
W01: WAIT Time by Task/Category (00848/QNZYB8)           Row 00001 of 00020
Command ===>                                              Scroll ===> CSR

Name          Description                Percent of Time in WAIT * 10.00%  ±1.3
                                            *....1....2....3....4....5....6...
DSNECP10-009  TCB=009AC2E0            22.59
→ APPLCN        Application Code      30.51
→ SYSTEM        System/OS Services    29.90
→ DB2SQL        SQL
→ NOSYMB        No
→ DATAMG        Dat
  → SYSOUT        Q
    → PUT
      → IEAVESLL
        → LLUOBT
```

```
D04: Dataset Attributes (00848/QNZYB8)                   Row 00136 of 00176
Command ===>                                              Scroll ===> CSR
VSAM file ZWB0TALS OPENed at 15:08:53.75 Wednesday Jan 13 2021

   DDNAME                ZWB0TALS
   Open Intent           KEY,DFR,DIR,SEQ,IN,LSR POOL=1
   Dataset Name          SFS.DOP0ZW.VS000.VS6340.ZWB0TABK.D.ME
   Management Class      PUSNN0N
   Storage Class         MDNNNNR
   Device Type           3390
   % Free Bytes in CI    0%                          Initial      Last
   Volume Serial         DSAFCL   CI Splits          0            0
   CI Size               4,096    CA Splits          0            0
   Record Size (LRECL)   2,056    Logical Records    242,445      242,445
   Number of Extents     2        Deleted Records    0            0
   SHAREOPTIONS          (1 3)    Insrted Records    0            0
   Organization          KSDS     Retrved Records    686,709      700,701
   CIs per CA            180      Updated Records    0            0
   Free CIs per CA       9        Bytes Free Space   2,695,168    2,695,168
   Free Bytes per CI     0        Number of EXCPs    2,587        2,587
   % Free CIs in CA      5%
   Strings               16       String Waits       0
   DATA Buffers          1280     String Waits HWM   0
   INDEX Buffers         1280
```

Example > Application Optimization at a large national bank

Julian informs the team that the VSAM application takes 3.3 minutes to execute and consumes ~.09 minutes of CPU time.

IBM estimates 96%+ savings with **IBM Z Table Accelerator**

They do a Proof of Concept

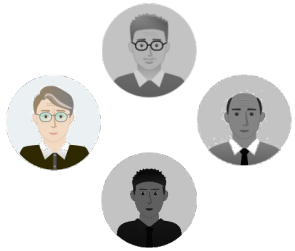
```
17.08.04 JOB20093  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG   PAGE   SWAP    VIO SWAPS
17.08.04 JOB20093  -LKUPVSM            STEP01    0000      6    .00    .00     .0     46   0      0      0      0     0
17.11.22 JOB20093  -LKUPVSM            STEP02      00  1430K    .09    .01    3.30  1461K   0      0      0      0     0
2
17.11.22 JOB20093  -LKUPVSM            STEP02    0000  1430K    .09    .01     3.3  1461K   0      0      0      0     0
17.11.22 JOB20093  -LKUPVSM  ENDED.  NAME-LOOKUP02             TOTAL CPU TIME=    .09  TOTAL ELAPSED TIME=  3.30
17.11.22 JOB20093  -LKUPVSM  ENDED.  NAME-LOOKUP02             TOTAL CPU TIME=    .09  TOTAL ELAPSED TIME=   3.3
17.11.22 JOB20093  $HASP395 LKUPVSM  ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  09 MAY 2020 JOB EXECUTION DATE
           22 CARDS READ
          104 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
           12 SYSOUT SPOOL KBYTES
         3.30 MINUTES EXECUTION TIME
```

Example > Application Optimization at a large national bank

With the help of Stan, the middleware systems programmer, Sagar authorizes an application change.

No logic is changed, only the file calling protocol

Instead of calling the BIN code table from VSAM, the file is placed in memory and the application will call the file directly from **IBM Z Table Accelerator**



```
LOOKUP-BANK.
    MOVE IN-CCARD-BIN TO BIN-CODE.
    READ BINCODE.
    EVALUATE FS-CODE
        WHEN "00"
            MOVE BIN-BANK                          TO OUT-BANK
        WHEN "23"
            MOVE "**** INVALID BIN CODE ****" TO OUT-BANK
        WHEN OTHER
            PERFORM VSAM-CODE-DISPLAY
    END-EVALUATE.
```

VSAM Code

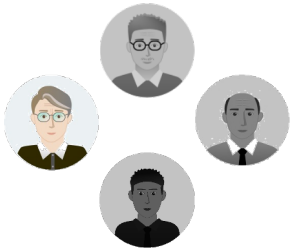
IBM Z Table Accelerator Code

```
LOOKUP-BANK.
    MOVE 'BINCODE' TO TA-TABLE
    MOVE 'FK'  TO TA-COMMAND
    CALL "DKJTCALL" USING TA-PARM TA-COMMAND-AREA BIN-REC.
    IF TA-ERROR = 0 THEN
        IF TA-WAS-FOUND
            MOVE BIN-BANK                          TO OUT-BANK
        ELSE
            MOVE "**** INVALID BIN CODE ****" TO OUT-BANK
        END-IF
    ELSE
        GO TO ZTA-ERROR
    END-IF.
```

Example > Application Optimization at a large national bank

After the KSDS is placed in-memory, the batch VSAM application runs again.

A **99.9%** elapsed time improvement and **90x** improvement to CPU consumption with **IBM Z Table Accelerator!**

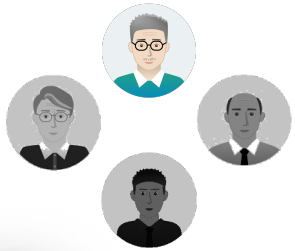


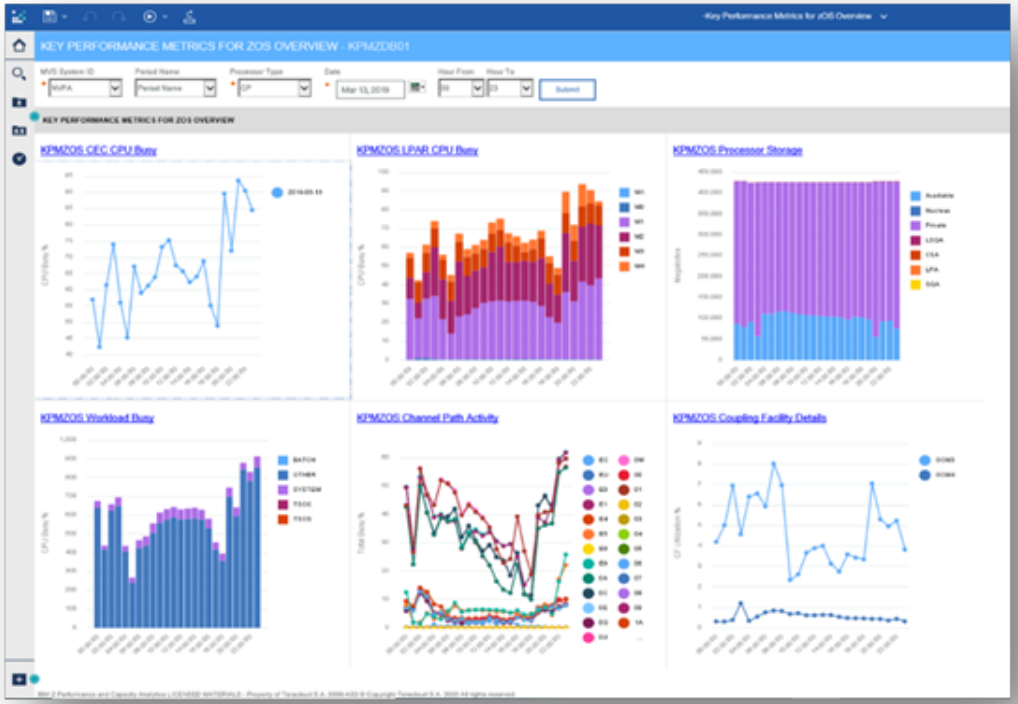
```
17.15.50 JOB20094  +DKJ00202I Initializing IBM Z Table Accelerator (IZTA)                                V110
17.15.50 JOB20094  +DKJ00204I - Executing in Step=STEP02
17.15.50 JOB20094  +DKJ00244I MAXNMTAB set to 012800
17.15.51 JOB20094  -ZLKUPVSM          STEP02      00   5250    .00    .00    .01  25118   0      0      0      0     0
2
17.15.51 JOB20094  -ZLKUPVSM          STEP02    0000   5250    .00    .00     .0  25118   0      0      0      0     0
17.15.51 JOB20094  -ZLKUPVSM ENDED.  NAME-LOOKUP02            TOTAL CPU TIME=   .00  TOTAL ELAPSED TIME=   .01
17.15.51 JOB20094  -ZLKUPVSM ENDED.  NAME-LOOKUP02            TOTAL CPU TIME=   .00  TOTAL ELAPSED TIME=    .0
17.15.51 JOB20094  $HASP395 ZLKUPVSM ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  09 MAY 2020 JOB EXECUTION DATE
           34 CARDS READ
          125 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
           13 SYSOUT SPOOL KBYTES
         0.01 MINUTES EXECUTION TIME
```

Example > Application Optimization at a large national bank

Zach verifies lower MSU consumption with **IBM Z Performance and Capacity Analytics**

Additionally, he sees lower overall MSU consumption and cost savings

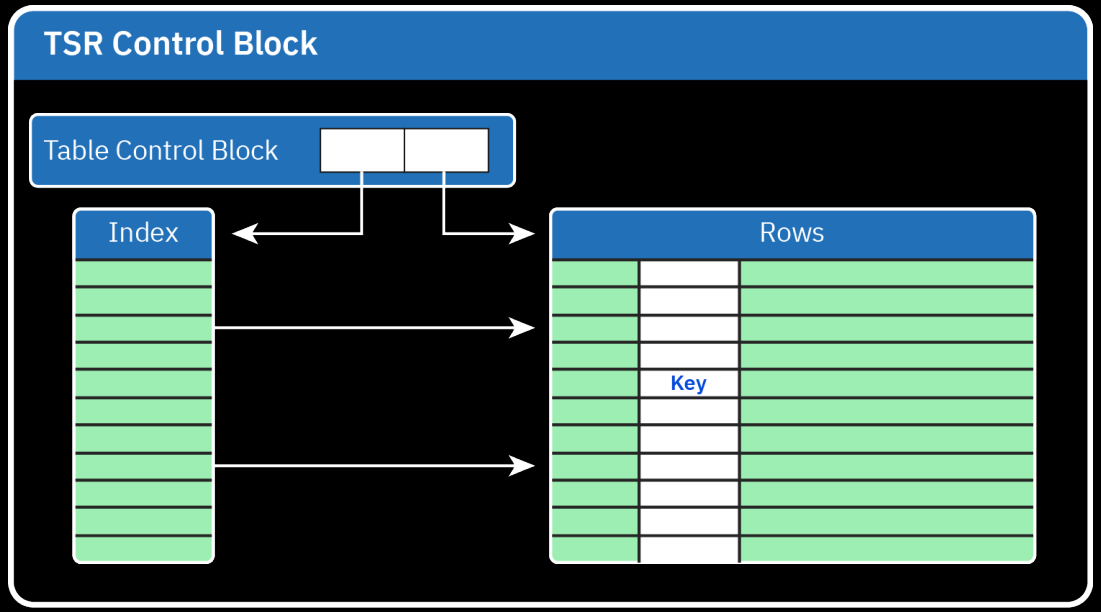




Deployment options

Tables

- Defining a Table
  - As well as Organization and Search Methods, tables are made up of Rows, Keys, Index. In-memory tables are kept in a Table Space Region.



Terminal Service Request (TSR) is in a Dataspace

A table can have multiple indexes

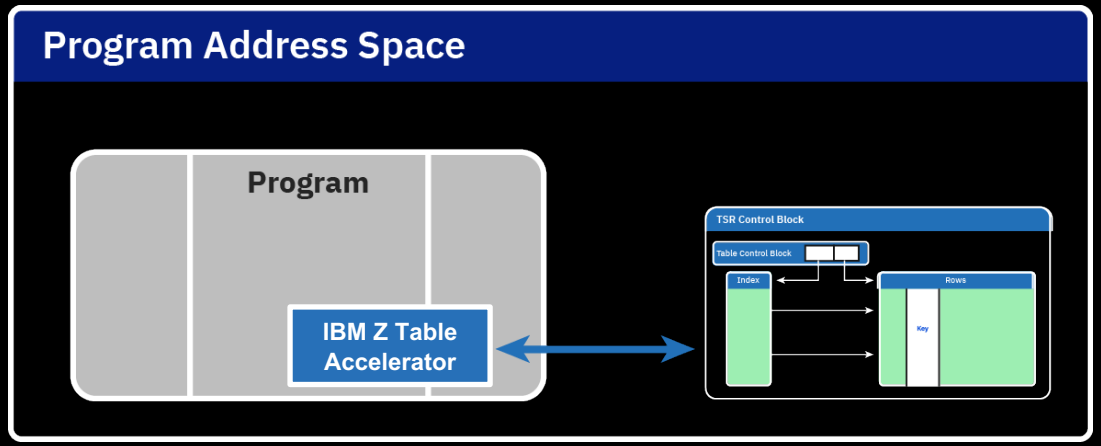
Keys can be 1 to 256 bytes

Rows can be 1 to 32k bytes

TSR has multiple Tables

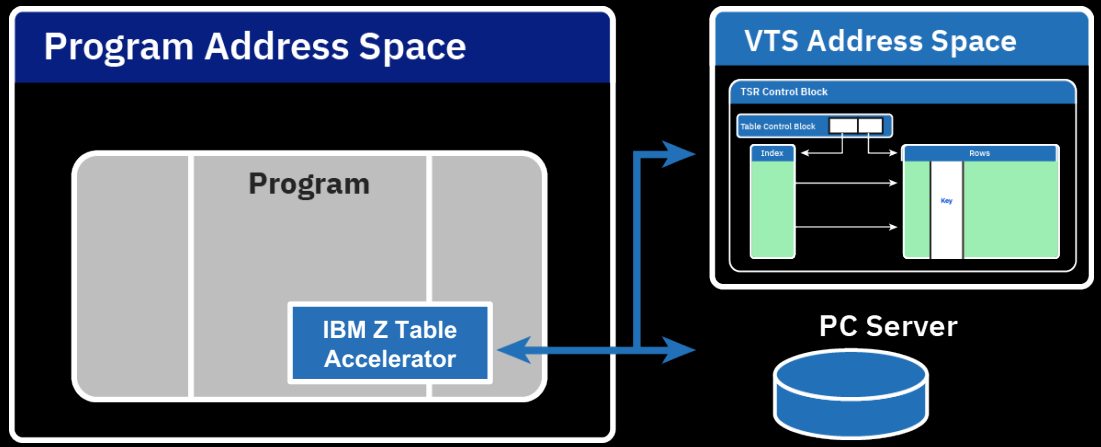
IBM Z Table Accelerator

- Accessing a Table
  - Compiled into program
  - Tables in Program Address Space
  - Math performed to find row – return to program
  - Batch and Online



IBM Z Table Accelerator — Virtual Table Share (VTS) mode

- Accessing a Table
  - Compiled into program
  - VTS PC Server starter task
  - Use PC Server to find VTS Address Space
  - Math performed to find row and return to program
  - Multiple programs, and CICS Regions



Closing Thoughts

1. **Reduce batch window contention**
2. **Reduce online response times**
3. **Reduce MIPS/MSU & associated cost**

Summary and Call To Action

IBM Z Table Accelerator provides a solution for improving mainframe application and database performance while reducing total cost of ownership

Improved transaction processing throughput drives improved workflow handling and optimizes performance and cost

Ensure IBM Z remains at the heart of your modern Enterprise IT Datacenter

Reach out to us for a deep-dive discussion on how you can leverage the current capabilities of your mainframe assets – as they are now – to increase your transaction processing capacity

Let IBM gather data from your environment to find out how impactful IBM Z Table Accelerator can be

Learn More

IBM Z Table Accelerator. Product Page

The latest updates & information about IBM Z Table Accelerator

www.ibm.com/products/z-table-accelerator

IBM Z Table Accelerator Announcement Blog

ibm.biz/WhatIsIZTA

Tailored Fit Pricing: How to manage workload in a world without capping

Learn how IBM Z Table Accelerator can get you in the best possible position to get the most from TFP

ibm.biz/IZDSCPandTFP

IBM Z Software Newsletter: Operations & Management Edition

Subscribe to the quarterly newsletter for operation, systems programmers and administrator to get the latest news, tips, blogs and trials in one place

ibm.biz/ZOperations

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*
ibm.com*
IBM logo*

*** Registered trademarks of IBM Corporation**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.
ITIL is a Registered Trade Mark of AXELOS Limited.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Zowe™, the Zowe™ logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.
Other product and service names might be trademarks of IBM or other companies.

Notes:
Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.


IBM

Top 3 Questions

How is this different from IBM Db2 Analytics Accelerator?
Short answer: IDAA deals with large/complex/analytical queries. IBM Z Table Accelerator deals with small/frequent/simple/mostly static queries.

Why not just use Db2 12's native in-memory capabilities?
Short answer: While Db2 does many things that IBM Z Table Accelerator will never do, the in-memory to in-memory comparison is still much slower.

What memory requirements does the product need itself?
Short answer: typically early implementations benefit mostly from small tables, usually < 100MB. This can grow over time, as more CPU is saved. We are aware of one account that has grown to 20,000 tables and uses 6GB of memory.

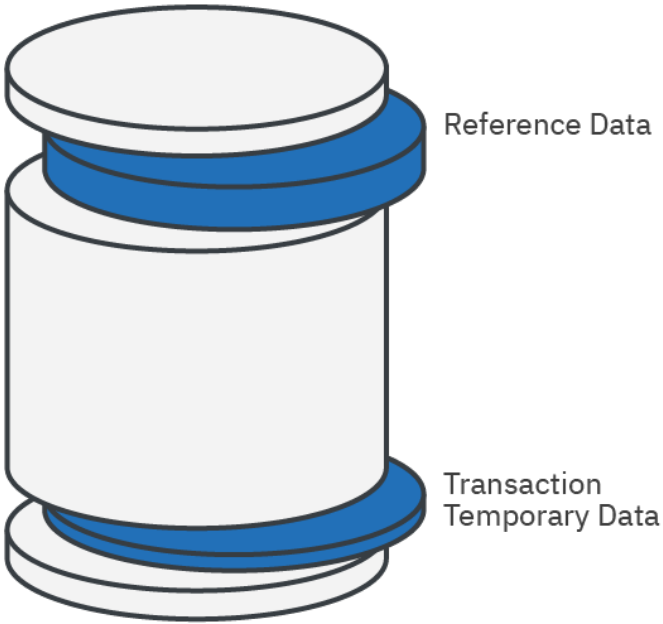
Technical Description

Not all Data is handled the same way…

Best fit for IBM Z Table Accelerator:

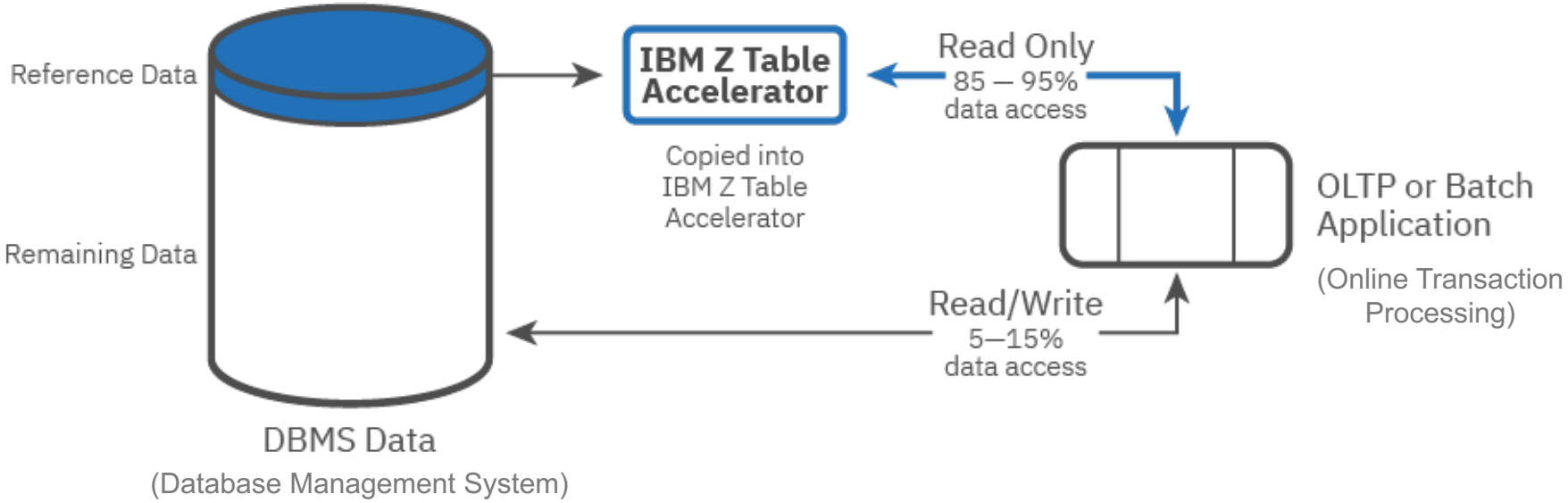
Reference data:
- 5-15% of your total data
- Changes infrequently
- Accessed often

Temporary data:
- Is created, processed and then deleted
- Generates a high volume of data accesses for the volume of data



How is this Possible?

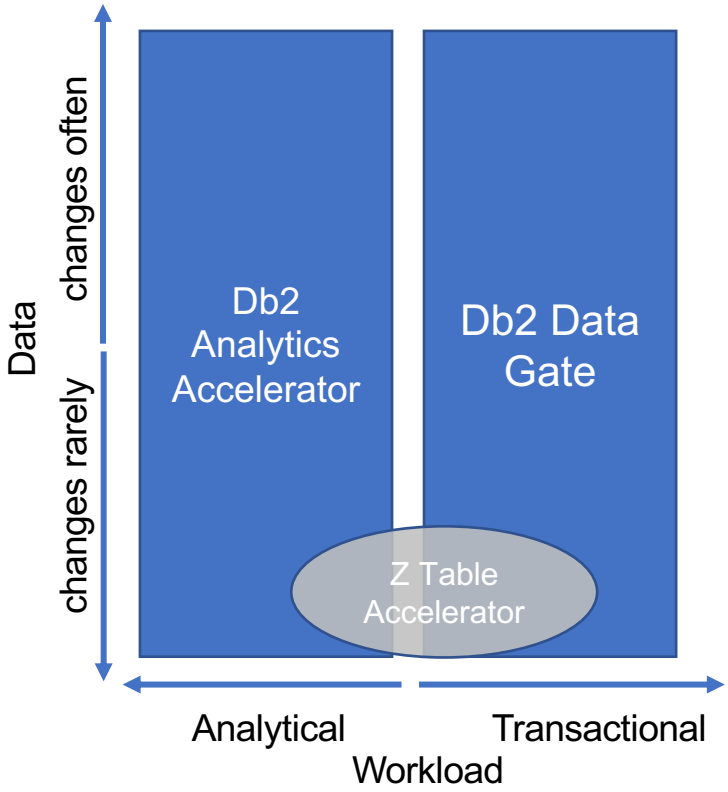
IBM Z Table Accelerator uses the shortest path to data



Technical Customer use cases & benefits of IBM Z Table Accelerator (IZTA)

Technical Use Case	Benefit
Share CICS tables with batch - Replace CICS tables and Table Space (TS) queues with IZTA tables	Allows batch applications to share CICS table data – reducing access time and I/O-related resource usage
Easier RRDS (Relative Record Data Set) access - Replace VSAM (KSDS (Key Sequence Data Set) and RRDS) files with IZTA tables.	Provides a performance boost. RRDS access is made easier by converting to IZTA. The ability to use binary searches and hash searches in IZTA improves performance.
In-memory scratch pad between transitions - Use IZTA as a temporary memory store.	A temporary in-memory scratchpad between CICS or IMS transitions requires no I/O – increases performance, reduced resource usage.
Replace CICS tables & Table Space (TS) queues w/ IZTA tables.	Allows batch applications to share CICS table data – reducing access time and I/O-related (In/Out) resource usage.
Replace ISPF (Interactive System Productivity Facility) tables	Allows ISPF application data to be shared with batch, ISPF, and CICS applications.
Replace control statement files with IZTA tables.	Move commonly used control information files into IZTA tables to reduce access time and program I/O.
Replace hard coded module tables w/ IZTA tables.	Eliminates the need to rebuild applications and control tables when a table update is required.
Unload Db2 table data into IZTA tables (use a Db2 unload utility to extract data from Db2 and then load data into a IZTA table).	Provides faster access to the data during peak Db2 system usage. Allows for dynamic sorting and alternate indexing.
Replace BDAM files with IZTA tables.	Simplifies and speeds up record access. IZTA provides the ability to create alternate indexes on the rows to provide increased flexibility and performance.
Use IZTA tables for summarizing and grouping data for reports. (Pre-processed data with SAS, and loaded results into IZTA tables.).	Use the IZTA tables to create reports. Allows data to be dynamically sorted & reorganized based on reporting requirements.
Use IZTA tables for data merge/purge operations and for printing box split operations.	Data merge/purge operations use IZTA in memory tables, providing high performance and ease of use. Printed forms box split processing requires creating and manipulating tables – by using IZTA tables, larger box split processing tables can be created and manipulated.
IZTA to propagate updates across all CICS regions.	IZTA can be used for small, volatile tables that are updated in batch or CICS, because it is easier to propagate the updates across all CICS regions.
Use IZTA for DB2 cursor scrolling	If cursor scrolling uses in-memory tables, there are no I/O delays in processing – achieve higher performance & throughput.
IZTA to continue sessions for online applications.	If sessions are continued using dynamically created in-memory tables rather than static database accesses, you'll be able to reduce both access time and I/O-related resource usage.

Differentiation



Db2 for z/OS

- Db2 for z/OS is *the* transaction database engine
- Mature, built-in acceleration technologies, such as FTB (Fast Traverse Blocks), large 1Tb contiguous in-memory buffer pools and inherent optimal access path selection for data access, which handle any variety of workloads (unlike a point-solution like Z Table Accelerator which targets a narrow special case)

Db2 Analytics Accelerator (IDAA)

- IDAA addresses complex analytical queries that require data intensive optimizations not present in transactional databases
- Implemented as Columnar-store technology provides optimal performance because in-memory tables alone don't achieve the necessary performance
- Provides complete application transparency – no application changes required

Db2 Data Gate

- Targets new high-volume transactional applications that customers don't want to deploy directly on Db2 for z/OS (for whatever reason)
- Provides a full-fledged database system able to handle ad-hoc and changing workloads without need to pre-identify hot-spot tables like Table Accelerator

IBM Z Table Accelerator

- Copies of data are stored in memory to reduce MSU consumption
- Leaves the data / app on IBM z/OS
- Code modifications required

Customers may benefit from more than one technology

- Many customers have a need for more than one type of data processing and may benefit from different solutions
- Choose the solution that suits your client's operating environment
  - Db2 Analytics Accelerator optimizes resource intensive analytical processing of Db2 for z/OS data
  - Db2 Data Gate optimizes access to Db2 for z/OS data outside of IBM Z for new applications
  - IBM Z Table optimizes access to reference data from within z/OS for CICS/Cobol/etc.

