Tailored Fit Pricing:

Technical Considerations

Dalibor Kurek
dalibor_kurek@cz.ibm.com

Enterprise Solutions Technical Sales Leader for Europe

Introduction

Two Enterprise Solution Variants:

1. Capacity

- Blended full cap pricing for everything
- **No rolling four-hour average (R4HA)**

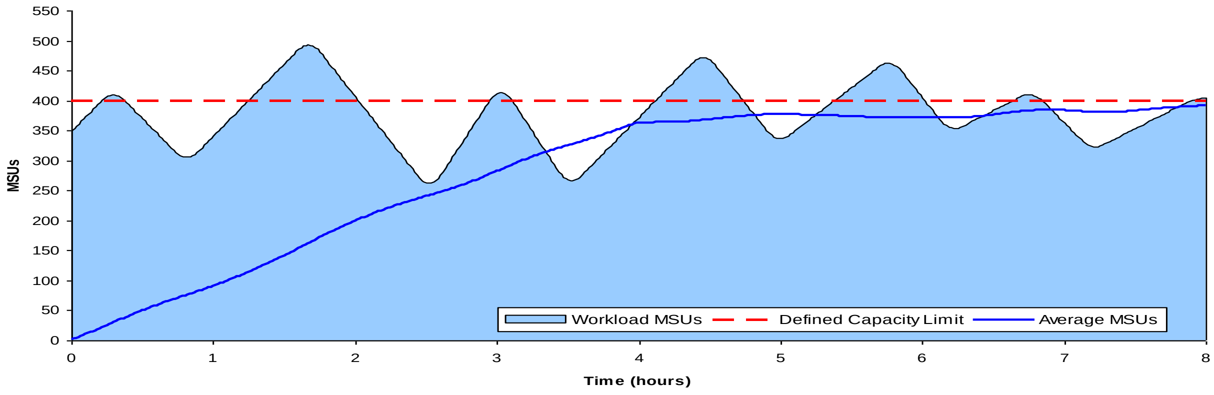
2. Consumption

- Cloud style "pay as you go" pricing on Z
- **No rolling four-hour average (R4HA)**
- Baselined on existing Million Service Units (MSU) consumed/pricing

Hardware / OS requirements:

- Z14 / ZR1 and newer
- z/OS 2.2 and newer

Rolling 4 Hour Average (R4HA)



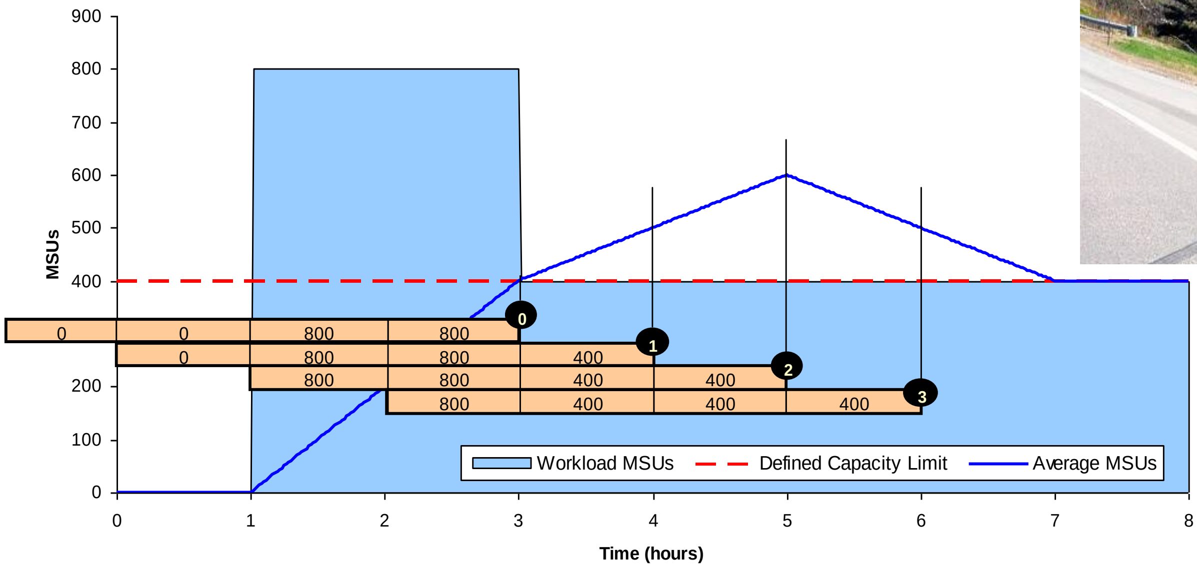
- Average consumption in LPAR in the last 4h (rolling) - SMF70LAC values
- MSU ≡ "Million Service Units per hour" (amount is defined for each z server)

- Technically calculated as an array of 48 intervals of 5 min = 4h

Capping was introduced to manage the R4HA…

- Sysplex / multi system outage
  - e.g. for LOCKs or RESERVEs not being freed timely
- System outage
  - e.g. for resources not being freed timely
  - Storage shortages
  - Work (e.g. Service Request Blocks (SRB)) backed up, common storage shortage
- Important work displaced
- Service levels missed
- Contention and increased promotion by System Resources Manager (SRM) dispatcher
  - Might be ok if displaced work is truly independent from important work – no shared resources
- Less important work displaced
- Goals missed
- Increased response times
- Increased Central Processor Unit (CPU) delays

Rolling 4 Hour Average Calculation



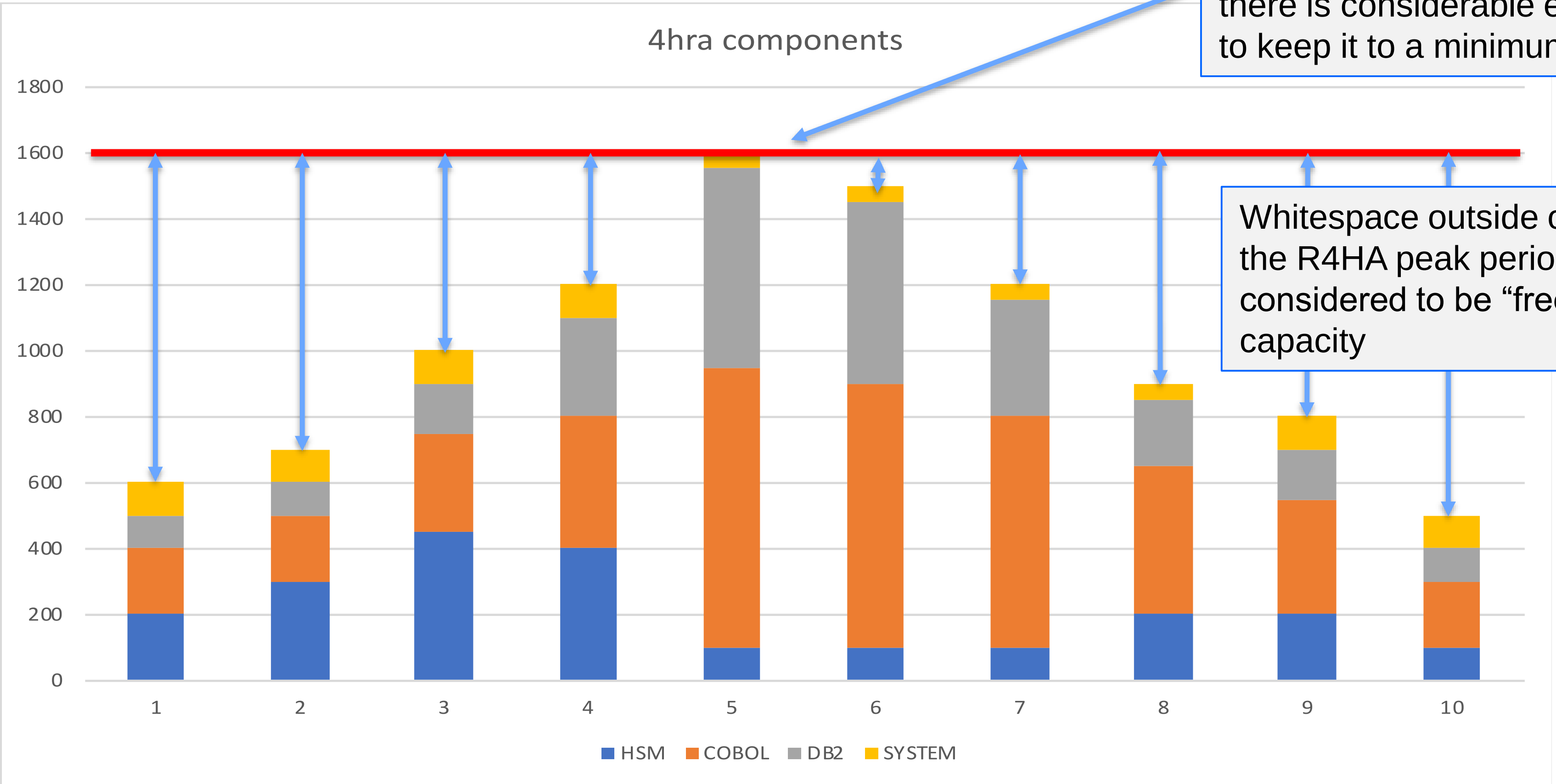
- Average is built from 48 * 5min buckets
- The average increases if the current consumption is bigger than the consumption 4 hours ago

Consequences

- From the good intent, to decouple Hardware size from software usage the sub-capacity pricing model developed towards a "savings" model.
  - The answer to "how can I optimize my R4HA" was very rewarding
  - Every MSU more on the machine, running during the R4HA has an disproportional high cost – whereas workloads outside the R4HA are not of interest (that much).
- Many clients use different products, to manage the R4HA – involving extra cost, manpower, effort, time etc.
- IBM introduced different capping technologies – Soft Capping, Group Capping, Absolute Capping, Hard Capping, Resource Capping.
  - Most of them to manage the R4HA in a multi LPAR, multi workload, single machine environment.
- Not to talk about technically "debatable" IT-Architectures, which are not good for clients – and not for IBM.

The resulting workload “culture” in R4HA world

Only the highest R4HA period gets attention, and there is considerable effort to keep it to a minimum



Whitespace outside of the R4HA peak period is considered to be “free” capacity

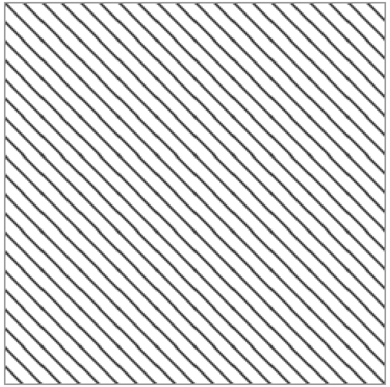
Enterprise Solutions – Options

Enterprise Capacity

Predictable & consistent monthly charges across Z stack, model requires new workload

Value:

- **Fixed capacity, includes committed growth**
- **Simple, predictable, consistent monthly charges**
- **Ultimate flexibility of workloads across z/OS, production & dev/test**
- **Single full-cap environment, reduced rates for DevTest & Growth**
- **Ability to grow further at known price points**

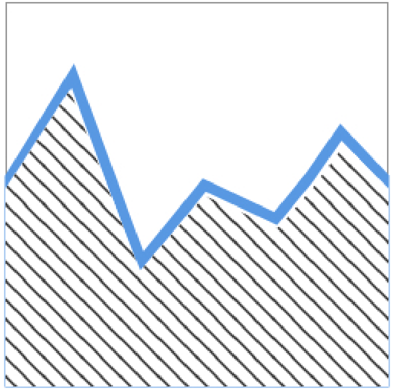


Enterprise Consumption

Predictable monthly charges with cloud-like flexibility over the contract term, model requires committed growth

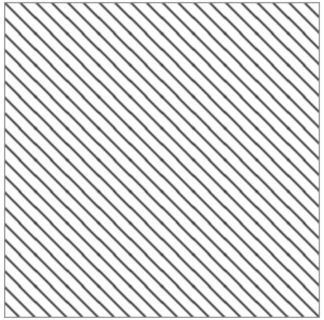
Value:

- **Baseline MSUs committed with variable discounted price for growth**
- **Pricing based on actual MSU consumption**
- **Ability to further grow at predictable and aggressive ‘per MSU’ pricing**
- **Increased/Fixed DevTest Capacity**



Capacity Model

Tailored Fit Pricing: Capacity Model

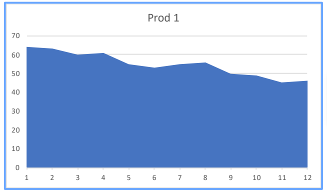


- Eliminates R4HA, subcap, sysplex rules
- Container is defined by number of engines
  - Increase of Container size is by engine granularity
  - MSUs are defined by machine model
- Blended zStack prices: single price for HW, OTC, S&S, MLC, TSS with a single, unified pricing metric (£€$/MSU)
- Can be either one "blended" price for all workloads, or it can be a price for different workload containers. In that case, engines must be dedicated to the containers.
- On/Off Capacity on Demand (CoD) capacity priced at full zStack (HW, OTC, S&S, MLC, TSS)
- Pre-negotiated zStack prices for any incremental future acquisition of capacity during the contract term
- Capacity management is done as usual

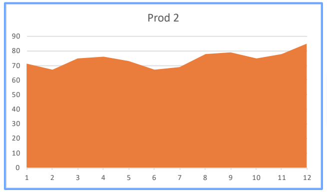
Tailored Fit Pricing: Capacity Model

Workloads

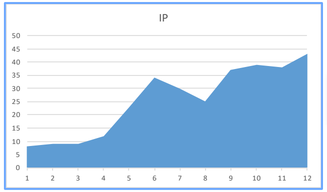
Pricing at fullcap



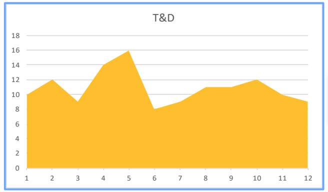
Prod 1
£€$/MSU



Prod 2
£€$/MSU



IP
£€$/MSU



T&D
£€$/MSU

- Hardware and software configurations based on the total capacity required per workload
- Different workloads
- Define other requirements (HA, DR, other)
- Decide on the number of containers
- Single container = blended zStack pricing

All workloads blended

- Multiple containers = zStack pricing per workload

Prod 1+2 | T&D | IP

Tailored Fit Pricing: Capacity Model

- An example of an MSU Grid:

Quarter	Price / MSU
Q1 2019	108,80
Q2 2019	103,93
Q3 2019	99,06
Q4 2019	94,19
Q1 2020	90,85
Q2 2020	86,49
Q3 2020	82,13
Q4 2020	77,76

- The example price above (in no actual currency) includes the whole zStack (HW, OTC, S&S, MLC, TSS), if acquired on top of the previously committed capacity during the period of the contract
- The size of (an) additional engine(s), expressed in MSUs, needs to be multiplied by the value above to arrive to the single-number price of the upgrade
- Similarly, On/Off Capacity on Demand in the Capacity Model also includes the price of the whole zStack (since SW is licensed full cap)
- Processor drawer upgrades and specialty engines have their own pricing grids
- I/O features and memory upgrades have a defined price list (no impact on HW maintenance or SW)

Tailored Fit Pricing: Capacity Model

- Example of pricing an upgrade during the term of the contract, on top of previously committed (full) capacity:
- A customer's machine is currently a z14 710, they would like to increase its capacity to 712
- MSU rating of 710 machine: 1793
- MSU rating of 712 machine: 2077
- **Delta MSU: +284**

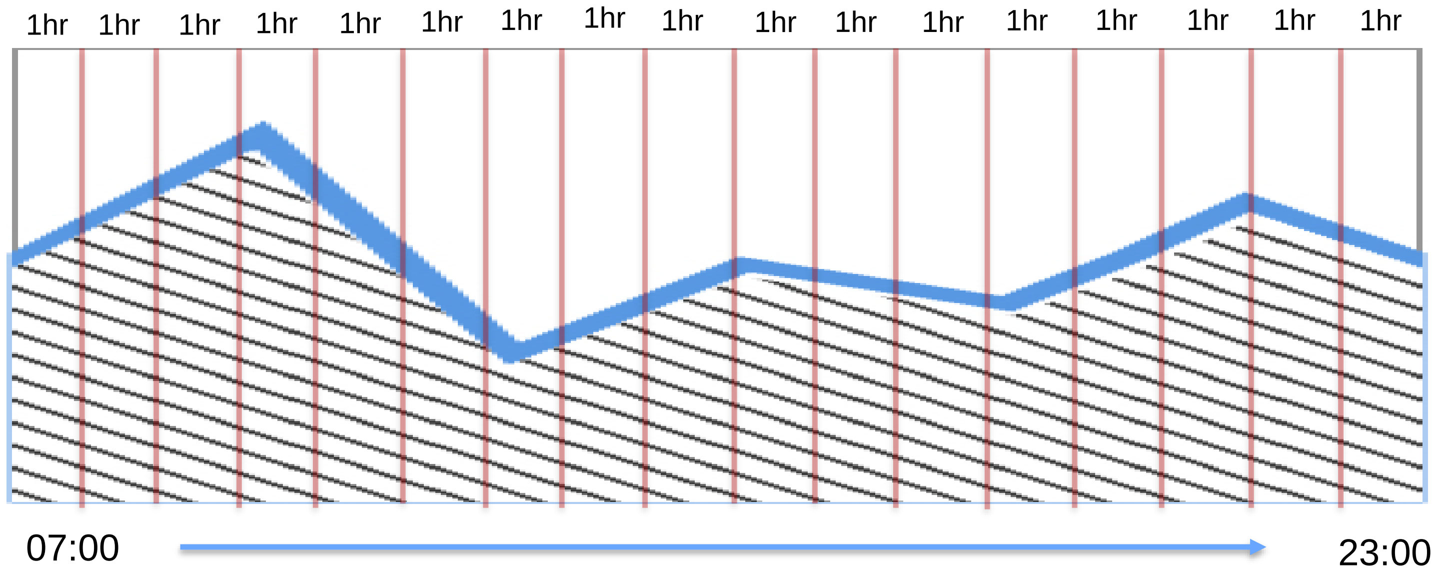
set'g	MIPS	MSU
712	17294	2077
711	16101	1939
710	14869	1793

Quarter	Price / MSU
Q1 2019	108,80
Q2 2019	103,93
Q3 2019	99,06
Q4 2019	94,19
Q1 2020	90,85
Q2 2020	86,49
Q3 2020	82,13
Q4 2020	77,76

- We are now in **Q2 2019**
- Price of the upgrade (full zStack):
  - 284 MSU x 103,93 £€$ = **29516,12 £€$**
- After this transaction, customer now has upgraded HW, acquired TSS coverage, and licensed the full SW stack to capacity setting 712 until the end of the contract term (end of Q4 2020)
- The above price is the full price in case the current machine has engines available to activate capacity setting 712; if a processor drawer upgrade is required, its price needs to be added as well

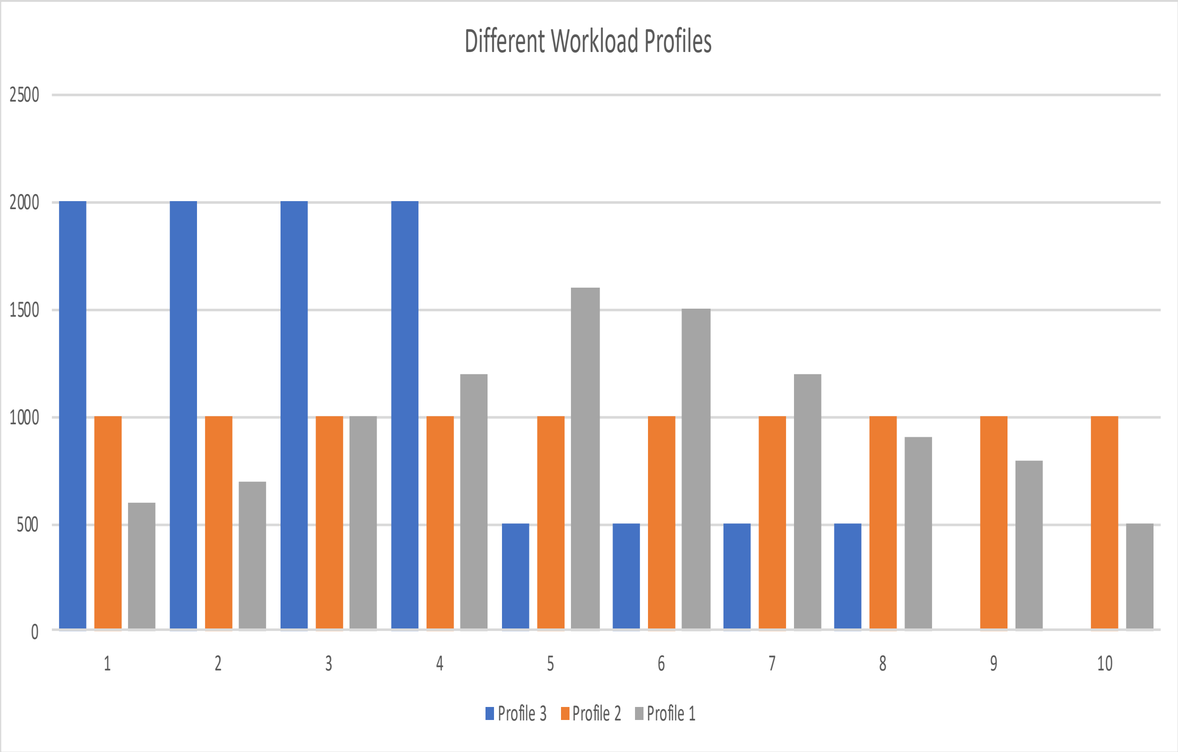
Consumption based pricing

Enterprise Consumption Measurement



- Client is collecting SMF70 records (as usual – no change)
- SMF 70 records contain the field SMF70EDT, containing the CPU consumption in the Interval
- SCRT creates an hourly view and adds up all consumption in one full hour to one number.

R4HA versus Consumption



Effect:

- Same amount of work in all 3 profiles
- Same CONSUMPTION (10.000 MSUs), meaning same price
- Different R4HA (2000 MSUs, 1600 MSUs, 1000 MSUs), mean different charges in the traditional model.

Charge Metrics for Tailored Fit Pricing

Pricing Model	Key metric for Capacity based charges	Key metric for Consumption based charges
Tailored Fit Enterprise Solutions	Hardware MSU rating. No SCRT reports.	Monthly aggregated SMF70EDT
Non co-located NewApp Solution	Monthly peak of hourly aggregated SMF70EDT	Monthly aggregated SMF70EDT
Co-located NewApp Solution (TRG)	Monthly peak of hourly aggregated SMF70_TRG_LAC	Monthly aggregate of SMF70_TRG_SUCP converted to MSU

"Monthly" indicates the billing period.

R4HA world

2000 MSUs — **L1**: CICS, Db2, z/OS

1000 MSUs — **L2**: Db2, z/OS

650 MSUs — **L3**: z/OS

500 MSUs — **Dev/Test Container**: z/OS, Db2, CICS, Cobol, Debug Tool

Pricing:

z/OS = MSU (L1 + L2 + L3) = 3650
Db2 = MSU (L1 + L2) = 3000
CICS = MSU (L1) = 2000
Dev/Test = Fixed Capacity at 500 MSUs

Consumption world

Online — **L1**: CICS, Db2, z/OS

Batch / DDF — **L23**: Db2, z/OS

500 MSUs — **Dev/Test Container**: z/OS, Db2, CICS, Cobol, Debug Tool

Pricing:

Consumed MSUs = (MSU Cons. L1 + L23)
Dev/Test = Fixed Capacity at 500 MSUs

Let’s take a look into SCRT Reports (R4HA first):

==B5========= SCRT MULTIPLEX REPORT - IBM Corp ====================

SCRT Tool Release	26.01.00				
Customer Name	What-Ever-Company				
Run Date/Time	02 Nov 2018 - 11:25				
Reporting Period	2 Oct, 2018 - 1 Nov, 2018 inclusive (31 days)				
Number of processors in Multiplex	2				
Machine identifier				M1C1	M2C1
Machine Type and Model				3906-732	3906-729
Machine Rated Capacity (MSUs)				4488	4128
Machine Model Changed				Y	Y
Exclude Data				N	N
Missing LPAR Data				Y	Y
Missing CPC Data				N	N
MLC Product Name	Number	MSU	Time		
z/OS V2 (Traditional)	5650-ZOS	4240	22 Oct 2018 - 12:00	1044	3196
z/OS V2 (zNALC)	5650-ZOS	2166	22 Oct 2018 - 12:00	2166	
DB2 11 for z/OS	5615-DB2	5266	22 Oct 2018 - 12:00	2635	2631
CICS TS for z/OS V5	5655-Y04	4364	22 Oct 2018 - 12:00	2166	2198
IBM MQ for z/OS V9	5655-MQ9	6272	22 Oct 2018 - 12:00	3140	3132
IPLA Product Name	Number	MSU	Time		
IBM Tivoli System Automation for OS/390 V3	5698-SA3	6406	22 Oct 2018 - 12:00	3210	3196
IPLA z/OS-Based	(All)	6406	22 Oct 2018 - 12:00	3210	3196

Method

- R4HA MSU per "product" (not really true – it is the LPAR the product is running in)
- Separation of z/OS traditional and z/OS zNew Application Licence Charge (NALC).
- IPLA R4HA z/OS number for pricing One Time Charge (OTC) products
- However this all leads to bill for mainframe Software at the end of each month.

Let’s take a look into SCRT Reports (consumption):

==N7===

DETAIL LPAR USAGE DATA SECTION

	Total MSU Consumed	Peak Hour Consumption	Date/Time	OS
I46	26347	92	08 Oct 2018 - 05:00	z/OS
I49	180724	1135	20 Oct 2018 - 01:00	z/OS
I4D	140301	815	07 Oct 2018 - 09:00	z/OS
I4E	4931	14	03 Oct 2018 - 02:00	z/OS
I4PE001	22772	670	31 Oct 2018 - 00:00	z/OS
I4PK001	967	18	28 Oct 2018 - 00:00	z/OS
I4PN001	4256	61	30 Oct 2018 - 08:00	z/OS
I4PP001(zNALC)	71207	1944	29 Oct 2018 - 10:00	z/OS
I4PP003	24454	921	30 Oct 2018 - 02:00	z/OS
I4PP005	15641	468	31 Oct 2018 - 02:00	z/OS
I4PV001	11846	41	13 Oct 2018 - 03:00	z/OS
I4TK003	3880	46	05 Oct 2018 - 16:00	z/OS
I4TT001	6504	22	20 Oct 2018 - 00:00	z/OS
ZNALI41(zNALC)	476583	2269	23 Oct 2018 - 11:00	z/OS
ZNALI43	154918	717	19 Oct 2018 - 00:00	z/OS
CPC	1145331	3995	22 Oct 2018 - 11:00	

Method

- Consumed MSUs per CEC are added up.
- The N7 Section is for planning. After signing a contract, a Container would be defined and LPARs would be assigned to a container.
- No "Consumed MSUs" per product. Meaning – it doesn’t matter where your SW is running.
- Leads to - no "pricing architectures" necessary.
- R4HA will not be reported in SCRT once client converts to Tailored Fit Pricing

Defining containers in SCRT - Example

3906-710
Serial: 02-ABCDE

Production Consumption Solution LPARs (Solution ID = A):
SYS1, SYSA

DevTest Solution LPARs (Solution ID = B):
SYS2, SYSB

3906-7A0
Serial: 02-2BCDF

Defining the Production Consumption Solution:

SPECIAL DD *
CONTAINER CPC=3906-ABCDE,IMAGE_ID=SYS1,ID=A
CONTAINER CPC=3906-2BCDF,IMAGE_ID=SYSA,ID=A
UPDATE CONTAINER, ID=A, SET_NAME="My Prod Solution"

Tailored Fit Enterprise Report - Key Sections

Section B5 is the report header.

It provides a summary view of each container across the entire environment.

Relevant metrics are shown for each solution container (i.e. solution ID).

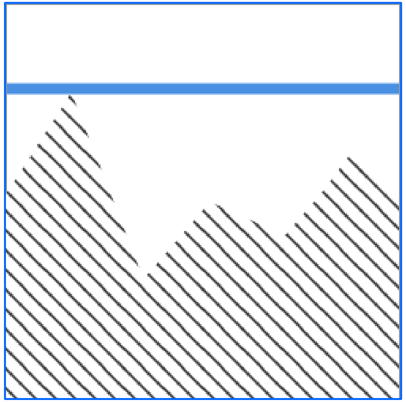
These include

- For DevTest solutions, a container peak size in MSU.
- The total monthly MSU consumption per container.

==B5========= SCRT ENTERPRISE TAILORED FIT REPORT - IBM Corp ===================						
SCRT Tool Release	27.1.0					
Name of Person Submitting Report:	John Customer					
E-Mail Address of Report Submitter:	customer@abc.com					
Phone Number of Report Submitter:	800-555-0123					
Customer Name	ABC Corp					
Run Date/Time	23 Apr 2019 - 13:53					
Reporting Period	2 Aug, 2017 - 1 Sep, 2017 inclusive (31 days)					
Number of processors in Multiplex	2					
Machine identifier					M1C1	M2C1
Customer number					200000000	200000000
Machine Serial Number					84-23456	84-A9876
Machine Type and Model					2964-714	2964-716
Machine Rated Capacity (MSUs)					2129	2358
Machine Model Changed					N	N
Exclude Data					N	N
Missing LPAR Data					N	Y
Missing CPC Data					Y	N
Container Identifier	Container Name	MSU	Time			
CPS2	DevTest	210	08 Aug 2017 - 05:00 UTC			210
Container Identifier	Container Name	TOTAL MSU Consumption				
CPS1	Production	1717505			813554	903951
CPS2	DevTest	56957				56957

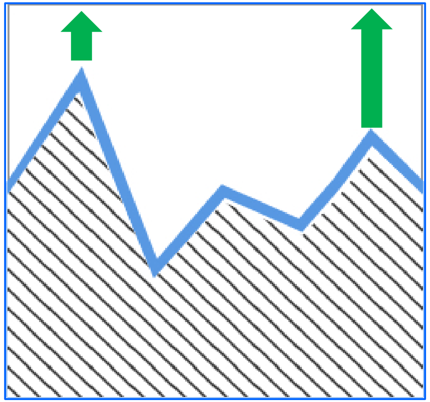
Per machine contribution to container metrics are displayed here:

IPLA software in a Consumption Solution



E.G. CICS VUE
1000 MSUs

MSU entitlement x 5000 MSU Hours
= MSU Hours in a year.



1000 MSU
entitlement x 5000
MSU Hours
=
5 million CICS VUE
MSU hours per
year

IPLA In a Capacity Model. Limitations:
- Capping requirement remains at license MSU entitlement.
- Prevents full adoption of Consumption value.
- Contradictory to the Consumption Model.

IPLA in a Consumption model. Value:
- All capping can be removed.
- IPLA products can be used in environments larger than their entitlement.
- Full value of Consumption is realized.
- Annual true-up of MSUs consumed.

Applicable to all IPLA Capacity-Based Software. IPLA licensing at full capacity of Consumption Container also available.

Things to think about

Capacity planning in a Consumption World

- Many clients used a capping algorithm to limit their CPU consumption to a certain MSU value. After this headroom for peaks was added, often something like 30% above the capping line.
- In short: The size of the machine was determined by the R4HA capping value.
- In a consumption-based world, we need to look differently into sizing machines. The driving question becomes “what helps the business”?
  - Is running more things in parallel (whenever during the day) beneficial for the business?
  - How big needs the machine to be, to reduce the batch window and may allow for a more elaborated ETL process after the batch?
  - OOCoD is the “old way” of adding capacity. But you either need to be very clever to anticipate peaks, or you need to accept delays until the capacity was added. “Always on” capacity is the better alternative to cover spiky workloads.

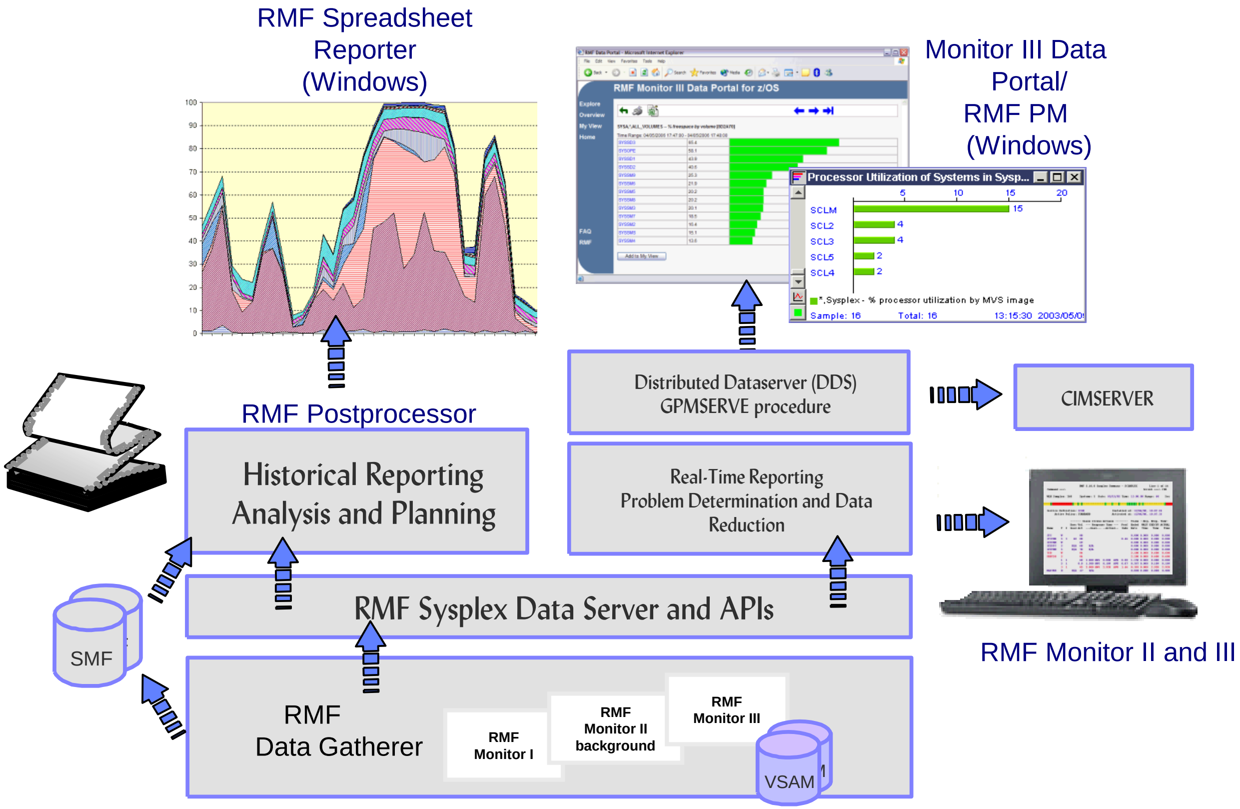
Capping in a "Consumption based" installation?

- The R4HA based pricing was driven by "peak usage in the month" (IPLA and MLC)
- Everything running in one specific 4h window of a month was pricing relevant – outside the window it was at least not directly pricing relevant
- The various capping algorithms helped clients to limit the MSU *peak* usage – with all negative consequences described earlier
- In a "Consumption based" installation, capping is irrelevant – as the pricing is not derived from the peak, but rather from every MSU consumed. Thus, controlling peaks alone (as capping does) is not helpful
- As there can‘t be a "control via capping" in a "Consumption based" installation – other methodologies need to be applied

Actions to manage consumption

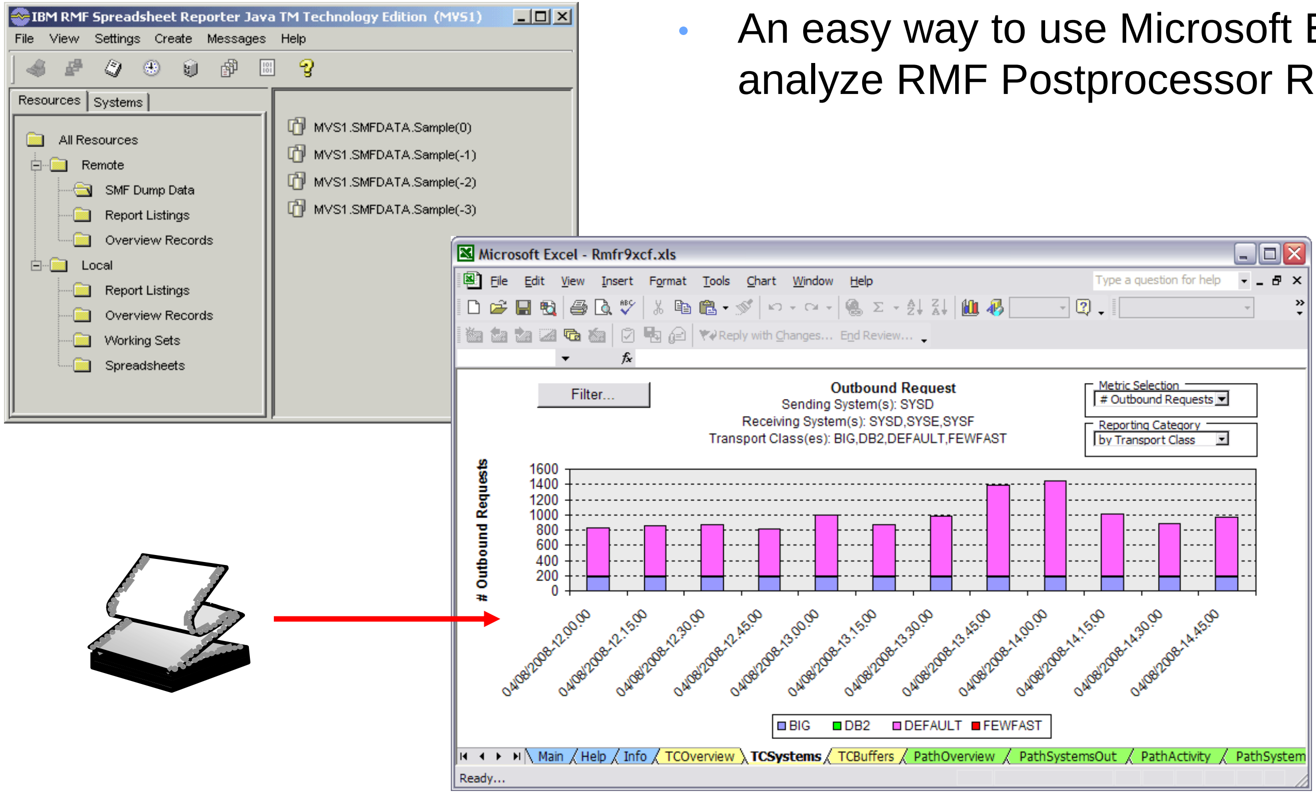
- Comparing past days with actual days can help to understand if the consumption is within expectation or not
- For example, a bank usually has a high peak at the first one or two days of the month. Therefore the 1. and 2. of May (e.g.) needs to be compared with the 1. and 2. of April / March / February to understand if consumption is in line with expectations. Comparing it with a "normal" workday would lead to wrong results
- One possible tool to do this is IBM Resource Management Facility (RMF) Spreadsheet reporter (comes with RMF), or customer-built tools and functions based on RMF data
- Another approach would be taking advantage of the DETAILED INTERVAL DATA in SCRT reports (for the time being, this is an undocumented function)
- The IBM Z Decision Support (IZDS) product provides the most functionality for this task, however, it requires Db2 for z/OS to store its data

RMF Product Overview



RMF Spreadsheet Reporter

- An easy way to use Microsoft Excel to analyze RMF Postprocessor Report data

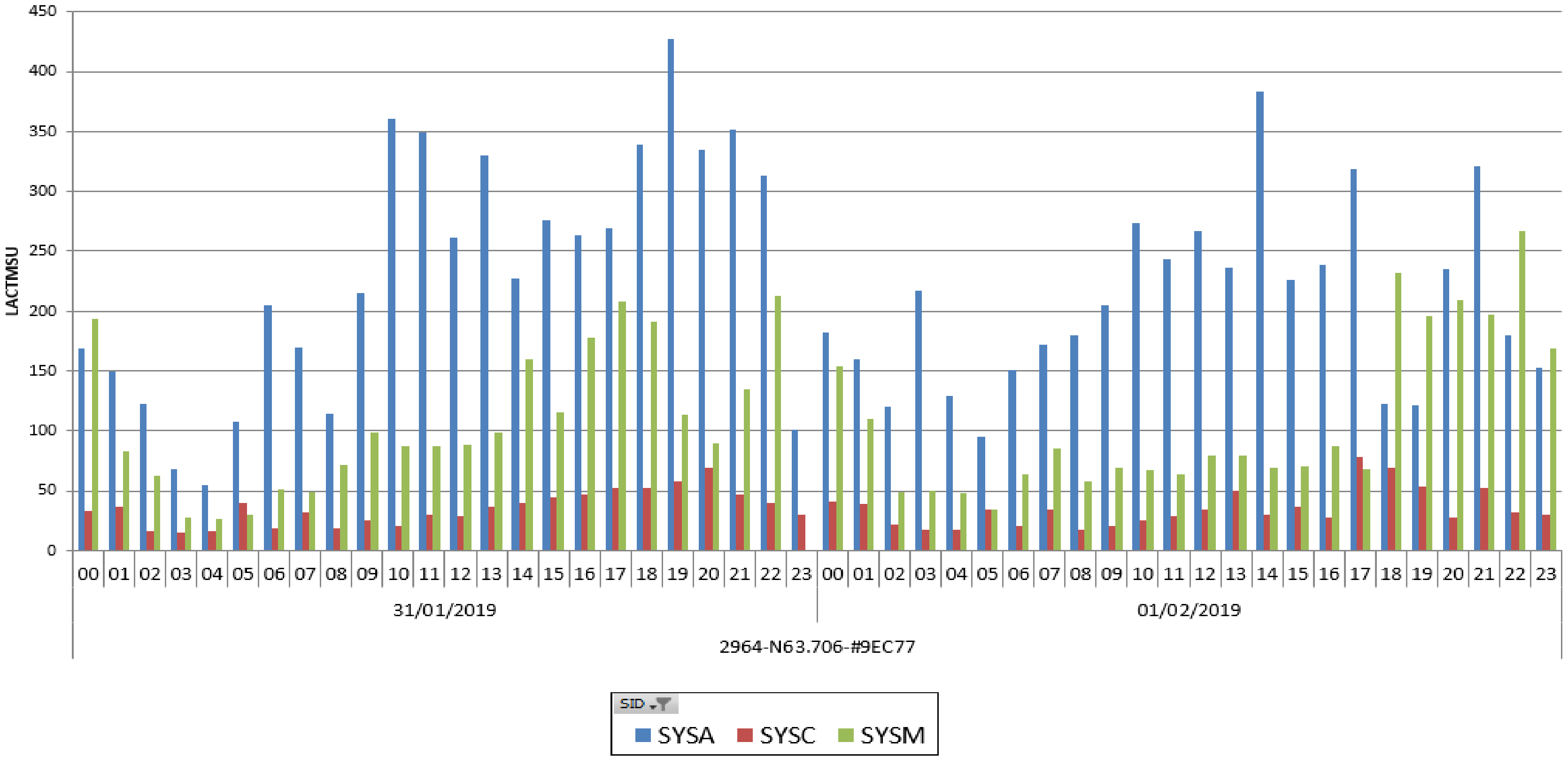


RMF Spreadsheet Reporter

- You can use RMF Overview Control statements to extract specific metrics from SMF records; this output can be converted into Excel Spreadsheet XLS format. At this point, you can either process the data using your own methods, or use the supplied generic RMF Overview Report Spreadsheet to get charts out of the data.
- The list of supported overview control statements can be found in the RMF Users Guide, Chapter 15. Long-term reporting with the Postprocessor, Overview and exception conditions

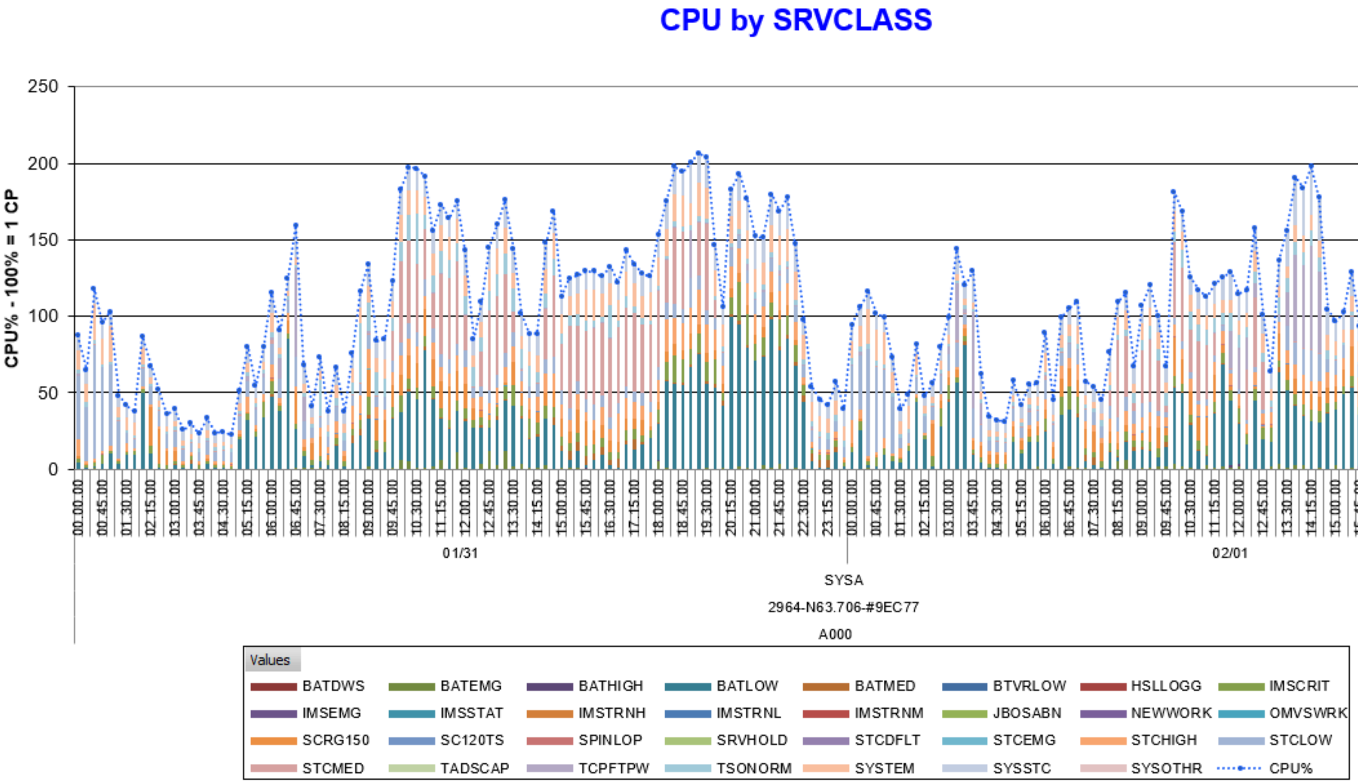
https://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3sc342664?OpenDocument

Monitoring CPU consumption with SMF 7x: ACTUAL MSU



Average actual MSU consumption by hour, over 2 days, for 3 partitions in the same machine

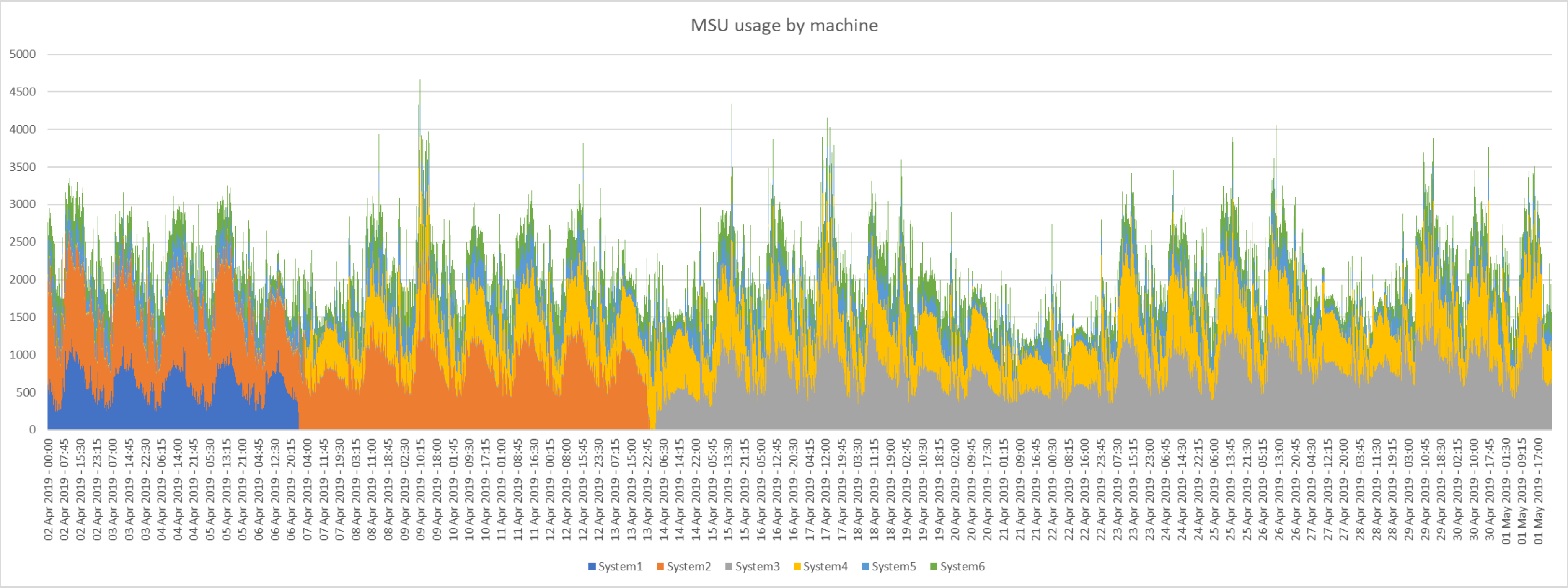
Monitoring CPU consumption with SMF 7x: CPU Consumption by service class



For one of the previous partition, we can see the breakout of CPU consumption by WLM service class, along the 2 days

SCRT-based (using DETAILED INTERVAL DATA / V9 section)

- + : SCRT reports are being generated and sent to IBM every month (very little extra work)
- + : SMF interval-level granularity time-wise (default is 15 min.)
- – : machine-level granularity only (not LPAR or lower)



What is CPSTRACE?

- An optional report generated by SCRT
- CPS = Container Pricing Solution

CPSTRACE DD statement

You can use the optional CPSTRACE DD statement to specify a data set or file that is to contain detailed, hour-by-hour tracking for Container Pricing reporting. The trace output is organized as comma-separated fields to allow the file to be read by a spreadsheet application.

The CPSTRACE output contains a section for each container. Each container section begins with a header record, followed by an hour-by-hour view of the TRGs and dedicated LPARs that contributed to the container's rolling 4-hour average utilization for each hour.

The CPSTRACE output is divided into two sub-reports labeled CPSTRACE 1 and CPSTRACE 2. Each sub-report contains a section for each container.

- The CPSTRACE 1 sub-report output contains a section for each container. Each container section begins with a header record followed by an hour-by-hour view of the TRGs and dedicated LPARs that contributed to the container's rolling 4-hour average utilization for each hour.

- The CPSTRACE 2 sub-report output contains a section for each container. Each container section begins with a header record followed by **an hour-by-hour view of the TRGs and dedicated LPARs that contributed to the container's MSU consumption for each hour**.

Example CPSTRACE output

```
==CPSTRACE=========================================================
Container Pricing Detailed Data

==CPSTRACE 1=======================================================
Four Hour Rolling Average

CPS1 DevTest Solution    Z111111-N31BB29-8FC80FDC07-NSDTZZZZ-91CC-465B-98CA-0565FF-E42B4F
CPS1 Date Time           Processor    Partition - LPAR1  LPAR2 CPC   Container
CPS1                     (tttt-sssss) TRG -       (lpar) TRG1  Total Total
CPS1 02 Aug 2018 - 00:00 3906-12345                 1450    49 1449       1449
CPS1 02 Aug 2018 - 01:00 3906-12345                 2283   251 2534       2534
CPS1 02 Aug 2018 - 02:00 3906-12345                 2638   275 2913       2913
CPS1 02 Aug 2018 - 03:00 3906-12345                 2795   531 3326       3326
CPS1 02 Aug 2018 - 04:00 3906-12345                 1976   841 2817       2817
...

==CPSTRACE 2=======================================================
MSU Consumption

CPS1 DevTest Solution    Z111111-N31BB29-8FC80FDC07-NSDTZZZZ-91CC-465B-98CA-0565FF-E42B4F
CPS1 Date Time           Processor    Partition - LPAR1  LPAR2 CPC   Container
CPS1                     (tttt-sssss) TRG -       (lpar) TRG1  Total Total
CPS1 02 Aug 2018 - 00:00 3906-12345                5803    197 6000       6000
CPS1 02 Aug 2018 - 01:00 3906-12345                3330    810 4140       4140
CPS1 02 Aug 2018 - 02:00 3906-12345                1423     93 1516       1516
CPS1 02 Aug 2018 - 03:00 3906-12345                 627   1024 1651       1651
CPS1 02 Aug 2018 - 04:00 3906-12345                2529   1437 3966       3966
...
```

CPSTRACE analyzer

CPSTRACE analyzer	Version: 1.2 (June 27, 2019)	Time periods to compare (1 to 12):			3		Container selection for analysis:		Container 1 (CPS 1)	
							Average daily MSU consumption	Highest observed daily MSU consumption	Highest week average daily MSU consumption	Highest whole week avg daily MSU consumption
	Reporting period	Containers	Machines	Names of containers	Machine S/Ns	Selected				
Import Base (reference) CPSTRACE report file	Apr 2019	2	2	Production1 DevTest1	3906-00001 3906-00002	CPS1	158 422	186 835	163 498	163 215
Import Analyzed CPSTRACE report file for this row	May 2019	2	2	Production1 DevTest1	3906-00001 3906-00002	CPS1	159 065	193 493	166 423	166 423
Import Analyzed CPSTRACE report file for this row	Jun 2019	2	2	Production1 DevTest1	3906-00001 3906-00002	CPS1	161 802	204 466	167 421	167 421
					Allowed:		166 343	196 177	171 673	171 376
Allow	5	% difference								

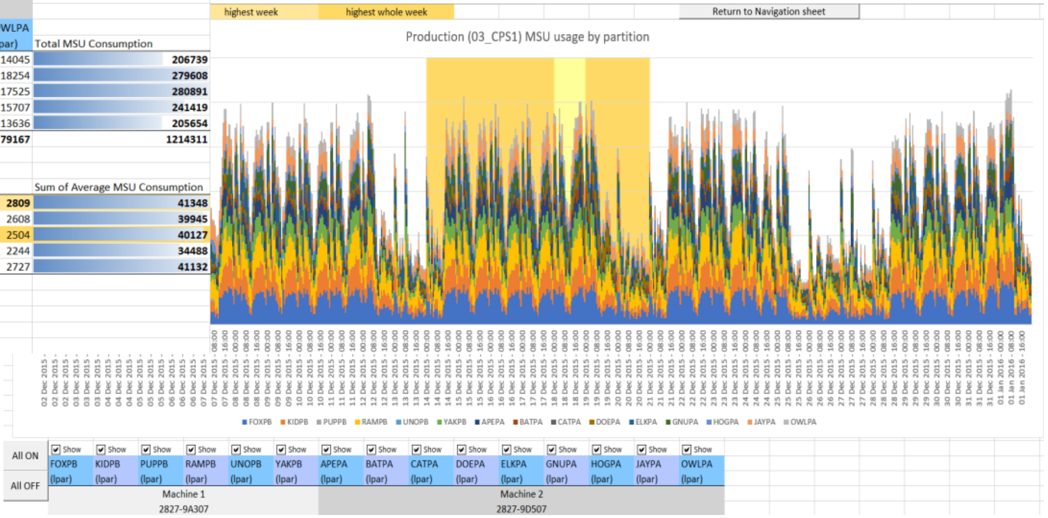
- at or below reference
- above reference, but within allowed difference
- above allowed difference, up to 2x of allowed % difference
- above 2x of % difference

Container name:
Production

Totals by week	# of days	Week number	FOXPB (lpar)	KIDPB (lpar)	PUPPB (lpar)	RAMPB (lpar)	UNOPB (lpar)	YAKPB (lpar)	APEPA (lpar)	BATPA (lpar)	CATPA (lpar)	DOEPA (lpar)	ELKPA (lpar)	GNUPA (lpar)	HOGPA (lpar)	JAYPA (lpar)	OWLPA (lpar)	Total MSU Consumption
			Machine 1 2827-9A307						Machine 2 2827-9D507									
02 Dec 2015 - 06 Dec 2015	5	0	31732	26348	754	28793	1000	20415	12392	6201	6750	6422	12664	12939	7566	18718	14045	206739
07 Dec 2015 - 13 Dec 2015	7	1	45341	35129	1152	39717	1406	26286	18492	8875	5866	7376	16533	17914	11375	25892	18254	279608
14 Dec 2015 - 20 Dec 2015	7	2	44824	33343	1081	39087	1399	25821	18770	9903	5222	12776	17053	17073	10483	26531	17525	280891
21 Dec 2015 - 27 Dec 2015	7	3	38132	28873	1067	36091	1292	22906	15795	7986	4549	8405	15339	14711	8143	22423	15707	241419
28 Dec 2015 - 01 Jan 2016	5	4	33993	25528	766	29302	1018	18795	13863	6890	4409	7225	13454	13178	7097	16500	13636	205654
Totals			**194022**	**149221**	**4820**	**172990**	**6115**	**114223**	**79312**	**39855**	**26796**	**42204**	**75043**	**75815**	**44664**	**110064**	**79167**	**1214311**

Daily averages by week	# of days	Week number	FOXPB (lpar)	KIDPB (lpar)	PUPPB (lpar)	RAMPB (lpar)	UNOPB (lpar)	YAKPB (lpar)	APEPA (lpar)	BATPA (lpar)	CATPA (lpar)	DOEPA (lpar)	ELKPA (lpar)	GNUPA (lpar)	HOGPA (lpar)	JAYPA (lpar)	OWLPA (lpar)	Sum of Average MSU Consumption
02 Dec 2015 - 06 Dec 2015	5	0	6346	**5270**	151	5759	200	**4083**	2478	1240	**1350**	1284	2533	2588	1513	3744	**2809**	**41348**
07 Dec 2015 - 13 Dec 2015	7	1	6477	5018	**165**	5674	201	3755	2642	1268	838	1054	2362	2559	**1625**	3699	2608	**39945**
14 Dec 2015 - 20 Dec 2015	7	2	6403	4763	154	5584	200	3689	2681	**1415**	746	**1825**	2436	2439	1498	**3790**	2504	**40127**
21 Dec 2015 - 27 Dec 2015	7	3	5447	4125	152	5156	185	3272	2256	1141	650	1201	2191	2102	1163	3203	2244	**34488**
28 Dec 2015 - 01 Jan 2016	5	4	**6799**	5106	153	**5860**	**204**	3759	**2773**	1378	882	1445	**2691**	**2636**	1419	3300	2727	**41132**

highest week
highest whole week
by daily average



IBM Z Decision Support (IZDS)

formerly IBM Tivoli Decision Support for z/OS (TDSz)

Key features:

- Automated, near real-time IT Operational data collection
- Continuous curation of the data collected, storing the results in Db2 for z/OS database for querying and further analysis
- Customized reports to communicate valuable system performance, capacity management, resource availability and cost allocation information
- Possibility to add new data sources to the collection and data consolidation process as needed
- Enhanced reporting capabilities in green screen as well as with Tivoli Common Reporting (TCR) and out of the box TCR reports
- Ability to stream Z curated data to Analytic platforms like Splunk and ELK Stack through the integration with IBM Common Data Provider for z Systems

IZDS sample output

CP Container MSU Monthly

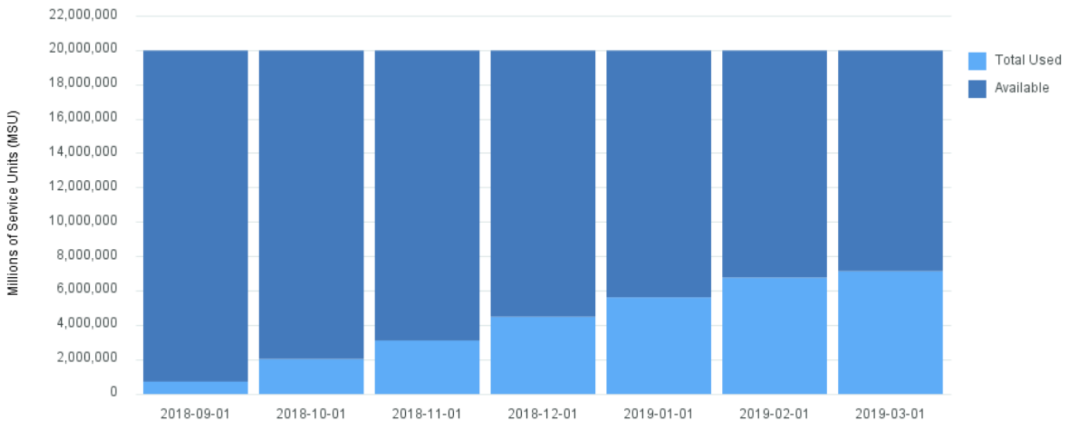
CP CONTAINER MSU MONTHLY - CPENT02

Container Name

* ENTERPR1

Submit

CONTAINER MSU BY MONTH - START DATE: 01 September 2018



Date	Used	Total Used %	Total Used	Available	Entitlement
2018-09-01	716,774	4%	716,774	19,283,226	20,000,000
2018-10-01	1,317,306	10%	2,034,079	17,965,921	20,000,000
2018-11-01	1,079,881	16%	3,113,961	16,886,039	20,000,000
2018-12-01	1,381,970	22%	4,495,931	15,504,069	20,000,000
2019-01-01	1,128,416	28%	5,624,347	14,375,653	20,000,000
2019-02-01	1,151,899	34%	6,776,246	13,223,754	20,000,000
2019-03-01	373,589	36%	7,149,835	12,850,165	20,000,000
Overall - Summary	**7,149,835**	**36%**	**7,149,835**	**12,850,165**	**20,000,000**

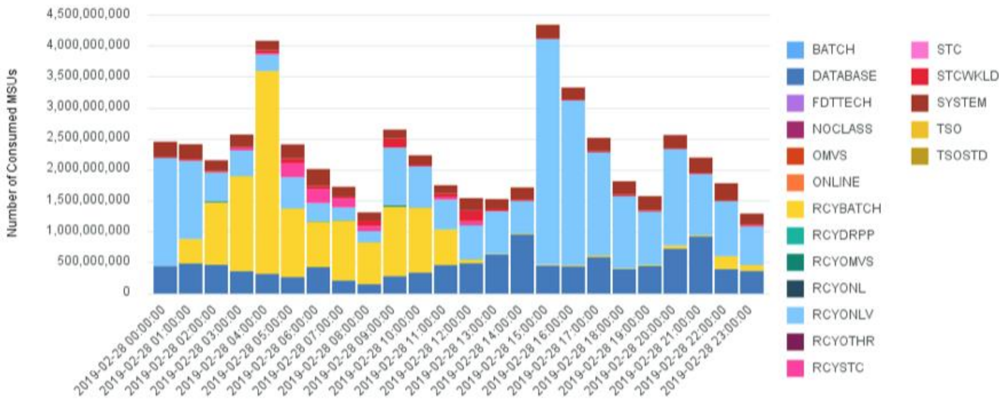
CP Workload MSU by LPAR

CP WORKLOAD MSU BY LPAR - CPENT07

Report Level: HOURLY | MVS System ID: RRCY | Date From: Feb 28, 2019 | Date To: Feb 28, 2019 | Hour From: 00 | Hour To: 23 | Submit

Only applicable at Timestamp Level

WORKLOAD MSU LPAR BY HOUR



Date/Time	BATCH	DATABASE	FDTTECH	NOCLASS	OMVS	ONLINE	RCYBATCH	RCYDRPP	RCYOMVS	RCYONL	RCYONLV	RCYOTHR	RCYSTC	STC	STCWKLD	SYSTEM	TSO	TSOSTD
2019-02-28 00:00:00	0	444,414,131	0	495,232	0	0	1,205,927	0	901,508	0	1,749,590,302	22,548	72,195	0	16,547,212	243,953,103	0	16,139
2019-02-28 01:00:00	0	490,788,527	0	927,045	0	0	391,247,041	0	1,021,490	0	1,257,965,114	30,823	7,931,015	0	24,324,983	242,739,934	0	5,764
2019-02-28 02:00:00	0	468,783,686	0	688,445	0	0	1,009,583,339	0	13,616,911	0	461,547,286	45,120	9,864,156	0	19,016,847	171,629,847	0	4,455
2019-02-28 03:00:00	0	366,598,601	0	1,027,875	0	0	1,530,483,629	0	2,338,718	0	413,213,554	35,100	39,228,160	0	16,506,554	201,902,247	0	24,077
2019-02-28 04:00:00	0	322,590,910	0	761,881	0	0	3,272,167,136	0	682,130	0	265,096,362	30,566	26,708,852	0	47,095,701	149,904,982	0	15,575
2019-02-28 05:00:00	0	271,788,944	0	646,158	0	0	1,101,421,356	0	1,153,809	0	506,999,422	34,244	228,631,020	0	73,842,049	228,678,258	0	44,107
2019-02-28 06:00:00	0	429,339,925	0	912,349	0	0	732,363,147	0	4,123,885	0	299,047,437	32,611	227,069,909	0	46,871,761	274,522,391	0	49,050
2019-02-28 07:00:00	0	215,634,761	0	791,000	0	0	962,004,296	0	4,427,485	0	211,755,265	28,095	146,679,225	0	21,603,539	164,966,205	0	10,537
2019-02-28 08:00:00	0	153,792,831	0	651,282	0	0	673,615,623	0	5,094,907	0	177,234,419	62,844	79,772,138	0	93,077,619	131,002,308	0	970,585
2019-02-28 09:00:00	0	286,970,118	0	725,879	0	0	1,117,516,055	0	20,067,515	0	935,878,331	42,376	882,809	0	146,980,618	143,412,356	0	27,392
2019-02-28 10:00:00	0	341,265,981	0	593,033	0	0	1,045,487,725	0	1,021,198	0	669,937,667	38,830	725,086	0	33,069,580	144,357,632	0	6,534

CP Container MSU Monthly Forecast

CP CONTAINER MSU MONTHLY FORECAST - CPENT04

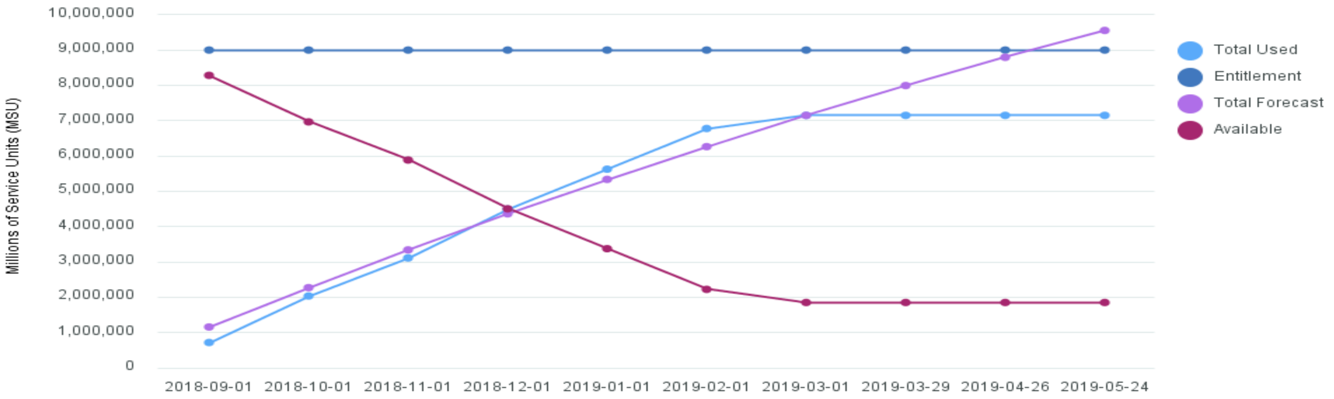
Container Name: * ENTERPR1

Forecast Algorithm: * LINEAR

Forecast Timestamp: * May 10, 2019 5:52:00 AM

Submit

CONTAINER MSU FORECAST BY MONTH - START DATE: 2018-09-01



Date	Used	Total Used %	Total Used	Forecast	Total Forecast %	Total Forecast	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
2019-05-24		79%	7,149,835	758,814	106%	9,553,150	1,850,165	9,000,000
Overall	**7,149,835**	**79%**	**7,149,835**	**9,553,150**	**106%**	**9,553,150**	**1,850,165**	**9,000,000**

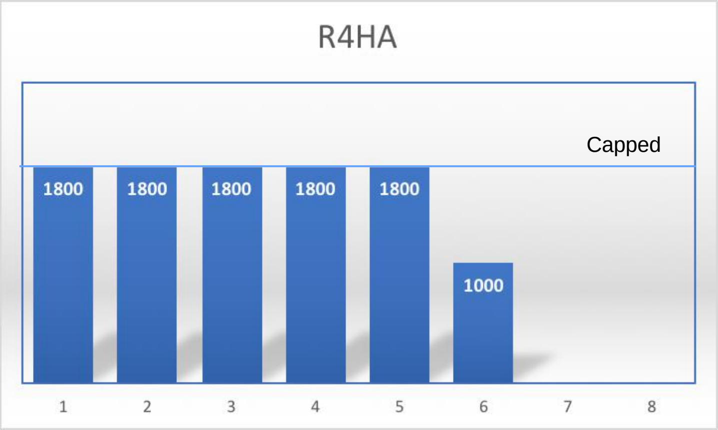
Conclusions on managing MSU consumption

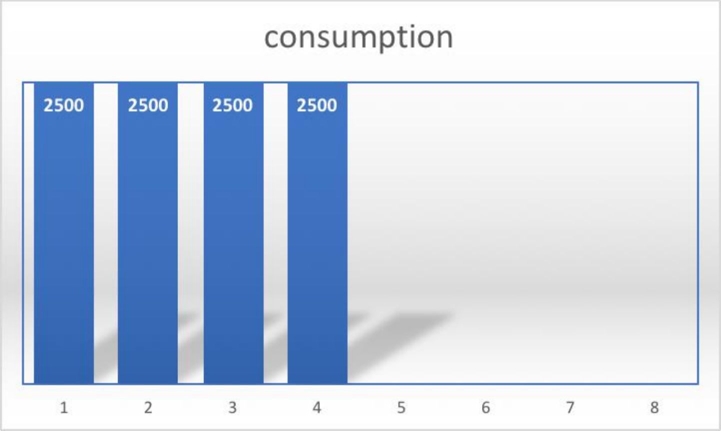
- Capping is no longer an option to manage cost in a consumption environment
- Understanding the workloads running on Z and their „normal“ consumption is key to deciding if everything is running „just fine“ or not
- For gaining this insight, clients need to compare past consumption per day, STC, JOB with actual consumption values
- It is important to compare „like for like“ days. For example - comparing the 1st working day in a month with a vacation day is for a bank not a useful comparison. The 1st working day of last month needs to be compared with the 1st working day of the actual month
- IBM offers tools like RMF (rather simple analysis), SCRT and IZDS (very detailed analysis) to help with monitoring and managing the MSU consumption

Optimizing for technical excellence

Maximize the hardware, minimize the batch window:

- Let’s assume the nightly batch requires a total of 10,000 MSUs to complete
- Let’s assume the machine is rated at 2,500 MSUs, but capped at 1,800 for the R4HA
- **By removing unnecessary soft caps, batch windows can be dramatically reduced**





Batch Optimization

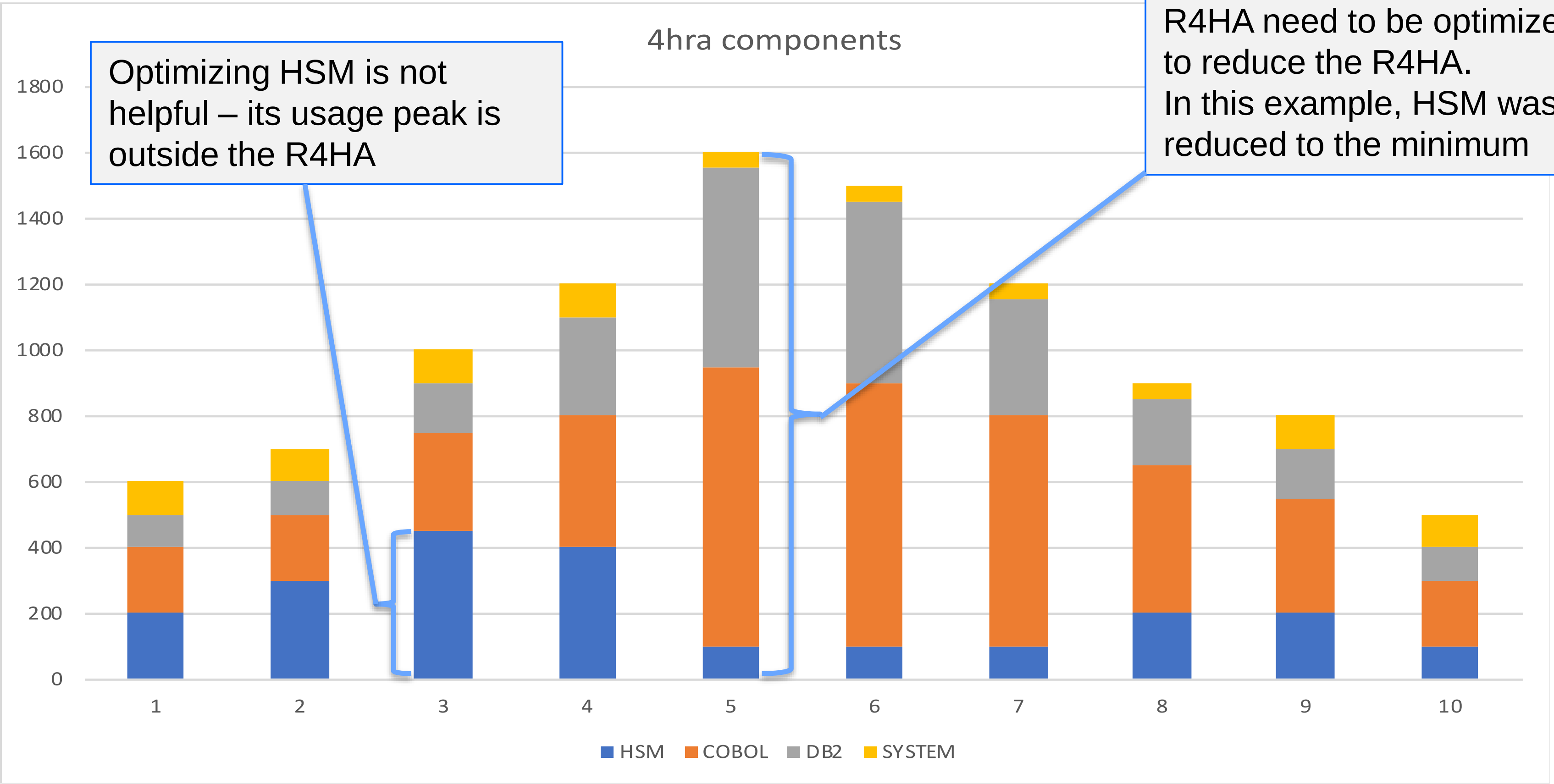
- As we are measuring "consumed MSUs" now - the time in the month when these MSUs get consumed doesn’t matter any more.
- Which means **there is no value anymore, in delaying certain, less important workload until the more important workload finishes** - except (of course) hardware is the limiting factor.
- This enables clients to run their batch workload uncapped thus gaining time between "batch end" and "online start". This gain can be extremely useful in case some abnormal batch job behavior - which needs to be recovered - occurs during the nightly batch.
- Many clients are not capping their online workload (because of the severe consequences concerning response times) - but they do cap their batch as it has the tendency to use all available capacity in the machine when uncapped. This is no longer a valid pattern.
- **Care should be taken when increasing parallelism:**
  - The memory footprint needed to run more work in parallel may increase (check paging rates)
  - The DASD IO rate may increase. This can be perfectly fine, but please consider that "Thin Provisioning", "PPRC", "Flashcopy", "Cascaded Flashcopy" ,… may add additional workload to your DASD boxes once you increase the Host IO rate.
  - CF IO rates may increase when running more workloads in parallel. Which means you need to check your coupling infrastructure for bandwidth and increased workload.

Optimizing the MSU consumption

- The R4HA charging methodology focused on the peak R4HA value per product per month. Which means, software running in these 4 hours contributed to the R4HA value and consequently, software not running in these 4 hours did not contribute.
- When using the consumption methodology, **every MSU consumed in a month contributes to the overall consumed MSU per month,** therefore, every single piece of work should be considered for optimization
- Software optimization is no longer only rewarding when the software is running in the R4HA. It is always rewarding – regardless when the software is running
- Best example is DFHSM/TCT (Transparent Cloud Tiering). DFHSM/TCT enables direct communication between DASD and TAPE systems for moving data thus reducing the amount of CPU used on z/OS. However – clients usually don’t run DFHSM migration during the R4HA
- Other programs with high potential for optimization are COBOL 6.2 and Db2 V12

DF/HSM TCT, Cobol 6.2, Db2 V12, etc.

Programs contributing to the R4HA need to be optimized to reduce the R4HA.
In this example, HSM was reduced to the minimum



Optimizing the MSU consumption – continued

Taking advantage of additional hardware resources may easily pay for itself.

- More L1 and L2 Cache available - reducing the RNI (newer machines, **such as the z15**)
- Less “food fight” going on in the machine, therefore reduced overhead (RoT: 4% less MSUs for 10% less in CPU busy)

Examples:

- **Integrated zEDC Accelerator (z15)** / zEDC Express (previous generations) for hardware-based compression
  - the throughput for each On-Chip Compression unit is 12GB/s, which equates to 48GB/s per drawer or 240GB/s for a fully populated 5 drawer z15
  - On-Chip Compression provides a up to 5% improvement in compression ratios for BSAM/VSAM datatsets over zEDC, while maintaining full compatibility
  - The IBM z Systems Batch Network Analyzer (zBNA) Tool’s zEDC analysis directly shows potential I/O and CPU savings
- **z15**: addition of new instructions for sort acceleration, which enable the improvement of sorting algorithms, reducing CPU utilization

Optimizing the MSU consumption – continued

Examples – continued:

- **IBM Automatic Binary Optimizer for z/OS (ABO)**
  - Optimize COBOL modules originally compiled with:
    - Enterprise COBOL for z/OS V4 and V3
    - COBOL for OS/390® & VM V2
    - COBOL for MVS™ & VM V1.2
    - COBOL/370 V1.1
    - VS COBOL II V1.4.0 and V1.3.x (LE enabled modules only)
- **Shared Memory Communication (SMC)**
  - By switching from HiperSockets to SMC (Direct) for z/OS-based applications, the utilization of the general processors for the z/OS workload can decrease significantly
- **CP Assist for Cryptographic Functions (CPACF) + Crypto Express6S**
  - Hardware accelerated encryption on every microprocessor core (CPACF) + on PCIe Hardware Security Module (HSM)
  - Performance improvements of up to 6x for selective encryption modes, with lower MSU consumption

Using IDAA / ETL in a Consumption world

In the past some clients were “reluctant” to install an IDAA or vIDAA with the argument “my complex queries are not allowed to run during the R4HA peak”. Which makes prefect sense in a “peak measurement world”.

With Tailored Fit Pricing it doesn’t matter any more “when” your jobs run - they are always counting, even on Sunday morning.

The usual ETL process companies have implemented in these days oftentimes run “sometimes outside the R4HA peak”. As this is a complex, resource consuming process. With tailored fit pricing you need to optimize the process itself instead of running it on some other day in the week (remember: there is no such thing as ”peak workload” any more).

Maybe it makes even more sense to optimize the ETL process with the help of IDAA and run the whole analytics on Z instead of anywhere else (especially once you put the cost of the other platform, time, security etc. into the equation).

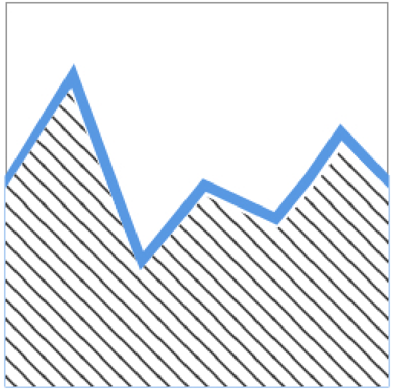
Removing MSUs in case of an error

- Excluding MSU consumption in exceptional circumstances

- New flavor of the EXCLUDE command:
  - EXCLUDE MSU_CONSUMPTION,CPC=tttt-sssss,HOUR=yyyy/mm/dd/hh,CPUTIME=seconds,ID=solutionid

- SCRT will convert the CPUTIME specified to MSU using the machine capacity at the hour specified and reduce the reported consumption by that amount in B5, CPSx headers

- Reasonable justification required in section D5
  - Use to adjust for a runaway job, IBM defect, etc.



z/End