Evolution of CPU and zIIP usage inside the DB2 system address spaces

Danilo Gipponi
Fabio Massimo Ottaviani
EPV Technologies

danilo.gipponi@epvtech.com
fabio.ottaviani@epvtech.com
www.epvtech.com

Disclaimer, copyright and trademarks

Disclaimer:

THE INFORMATION CONTAINED IN THIS PRESENTATION HAS NOT BEEN SUBMITTED TO ANY FORMAL REVIEW AND IS DISTRIBUTED ON AN “AS IS” BASIS WITHOUT ANY WARRANTY EITHER EXPRESS OR IMPLIED. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A USER RESPONSIBILITY AND DEPENDS ON THE USER’S ABILITY TO EVALUATE AND INTEGRATE THEM INTO THE USER’S OPERATIONAL ENVIRONMENT. WHILE EACH ITEM MAY HAVE BEEN REVIEWED FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. USERS ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK.

Copyright Notice:

Trademarks:

All the trademarks mentioned here belong to their respective companies.

Agenda

- Introduction
- System AS CPU usage in DB2 V9
- System AS CPU usage in DB2 V10 and V11
- Evolution of MSTR functions
- Conclusions

Introduction

Introduction

- DB2 is the preferred repository for mission critical data at all z/OS sites
- Performance of z/OS and non z/OS based applications accessing DB2 data is critical for every company business
- Managing even simple DB2 environments with continuously shrinking staff is more and more a challenge

Introduction

- DB2 system programmers and DBAs have normally not enough time to dedicate to performance analysis and tuning of DB2 subsystems and applications
- They also don't have time to provide data and reports needed by other performance analysts (CICS, IMS, z/OS, WebSphere, Capacity Planning)

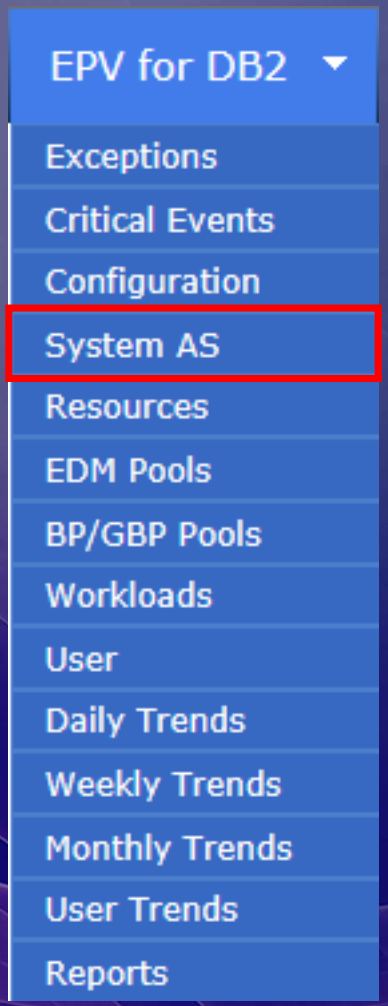
Introduction

- EPV for DB2 is an “out of the box” Performance and Capacity Management solution for z/OS environments of any size, type and complexity
- Some of the most important sites in Italy already use it
- EPV for DB2 uses standard input data normally available in any z/OS environments (mostly SMF records)

Introduction

- The main goal of EPV for DB2 is to provide a comprehensive view of DB2 environments of any level of complexity and to allow a quick and effective top down analysis of DB2 subsystems and workloads
- This goal is accomplished through a completely automated process, without any need for human intervention
- More than 500 different report types in HTML pages are provided

Introduction



- A.S. CPU USAGE
- A.S. IIP USAGE
- DIST A.S. CPU UTILIZATION
- DIST A.S. IIP UTILIZATION
- DBM1 A.S. STORAGE
- DIST A.S. STORAGE
- DBM1 A.S. VIRTUAL STORAGE
- DIST A.S. VIRTUAL STORAGE
- MSTR LOG WRITES
- MSTR LOG READS

Introduction

- All the graphs, tables reported in the following have been created starting from the EPV for DB2 views highlighted in the previous slide
- This presentation is based on a white paper published in the EPV Newsletter (January and February 2016)
- For time reasons we will not discuss the performance impact of zIIP over utilization on DB2 but we strongly suggest you to read this chapter in the paper

System AS

CPU usage in V9

System AS – CPU usage in V9

- No zIIP offload is possible with DB2 V9
- The next picture shows the split of DBM1, MSTR and IRLM functions in TCB and SRB mode (based on DB2 Managing Performance manual)
- Underlined functions refer to data sharing environments

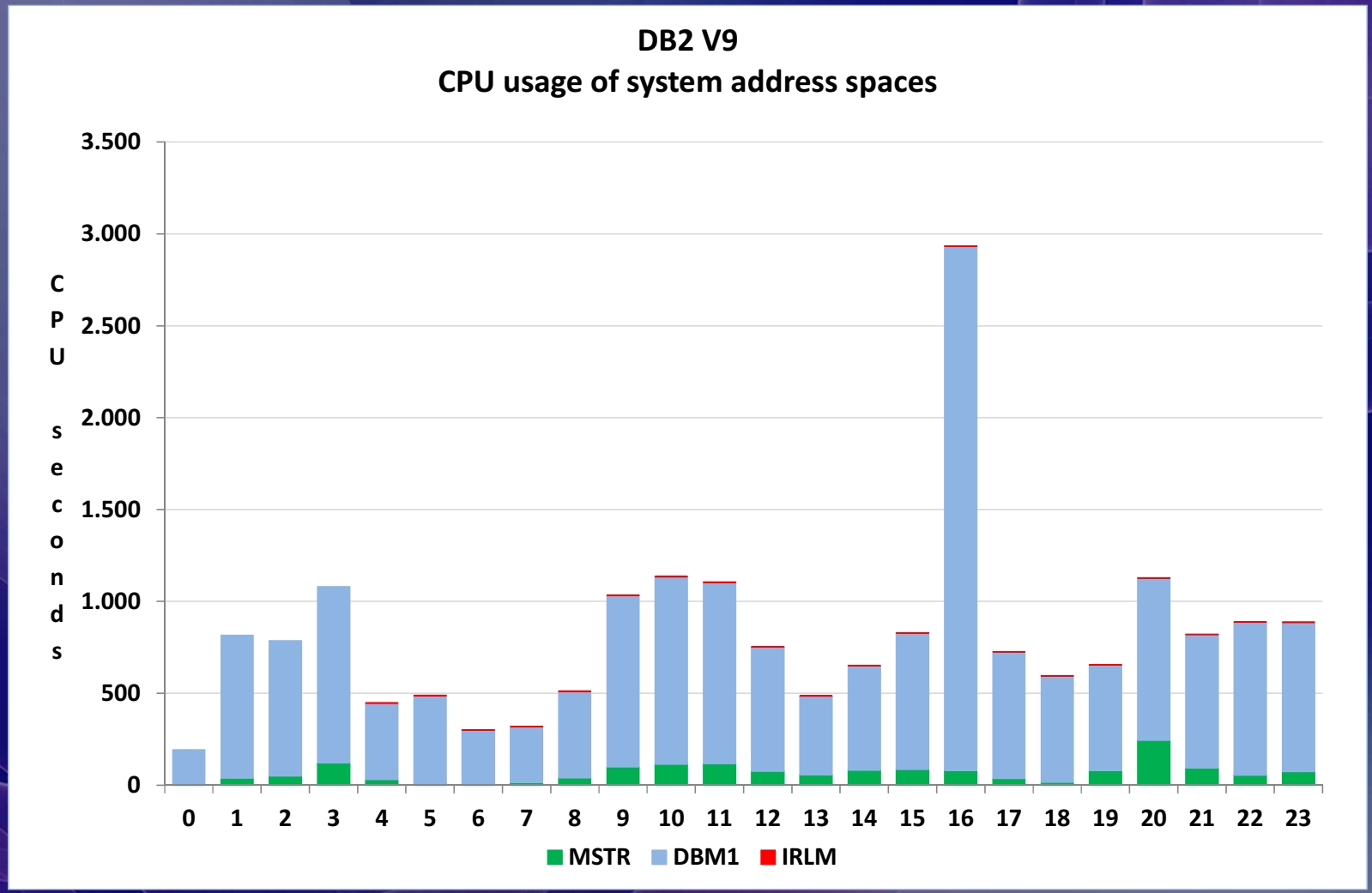
System AS – CPU usage in V9

System AS	TCB mode	SRB mode
DBM1	Opening and closing of dataset	Deferred writes
	DBM1 full system contraction	Prefetch reads
	Preformat	Parallel child tasks
	Extend	Castouts
		Asynchronous GBP writes
		P-lock negotiation
		Notify Exit
		Page set close or pseudo-close to convert to non-GBP dependent
		GBP checkpoints
MSTR	Archiving	Physical log writes
	BSDS (bootstrap data set) processing	Thread deallocation
		Update commit (including unlocking of page P-locks)
		Backouts
		Checkpoints
IRLM	Error checking	Local IRLM latch contention
	Management	IRLM and XES global contention
		Asynchronous XES contention
		P-lock negotiation
		Deadlock detection

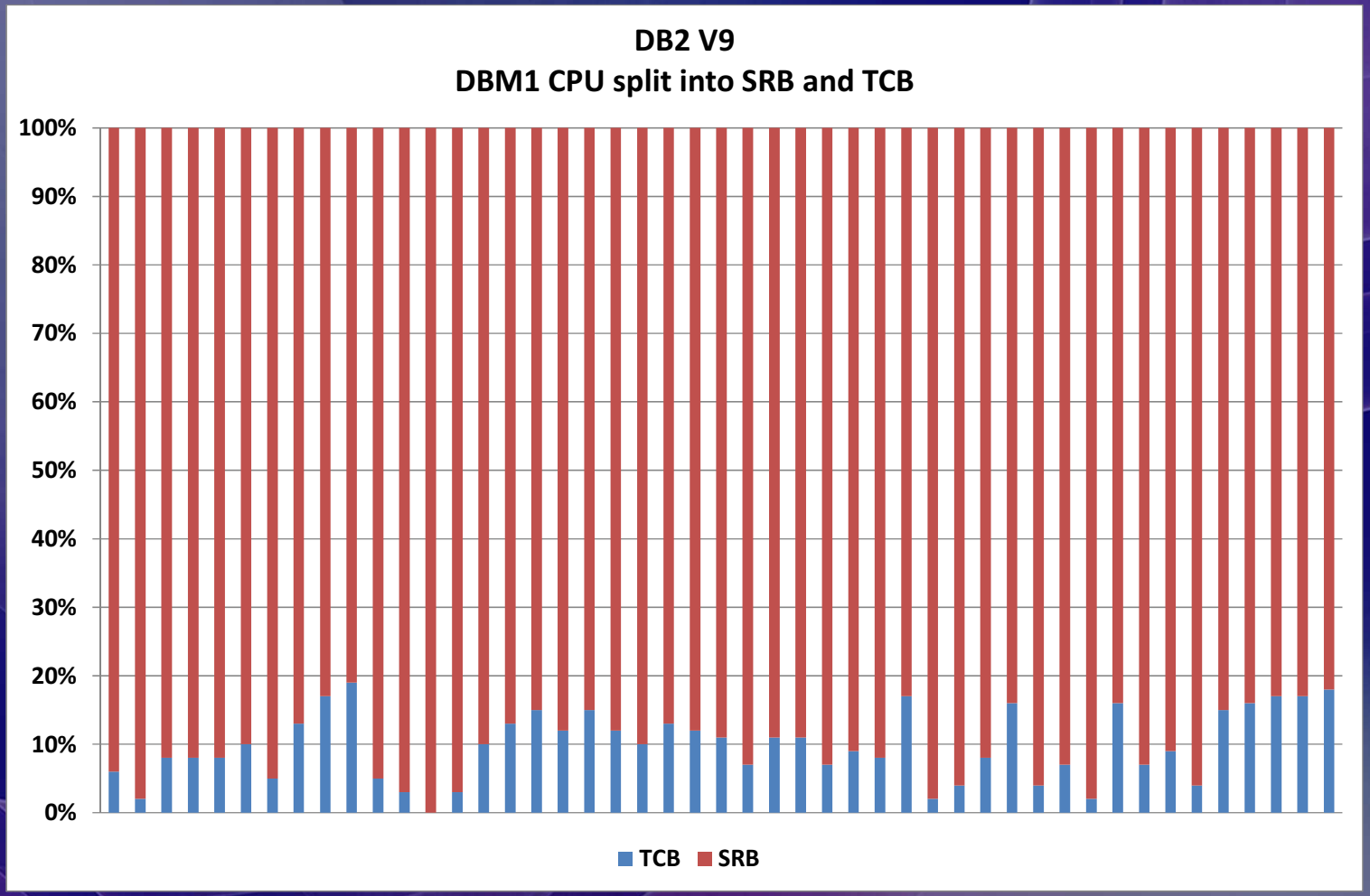
System AS – CPU usage in V9

- DBM1 is by far the major CPU consumer
- Normally most of DBM1 functions run in SRB mode

System AS – CPU usage in V9



System AS – CPU usage in V9



System AS

CPU usage in V10 & V11

System AS – CPU usage in V10 & V11

- Most of DBM1 functions offloaded to zIIP in DB2 V10 (light blue) and DB2 V11 (light green)
- With DB2 V11 MSTR logging functions are offloaded to zIIP (light green)

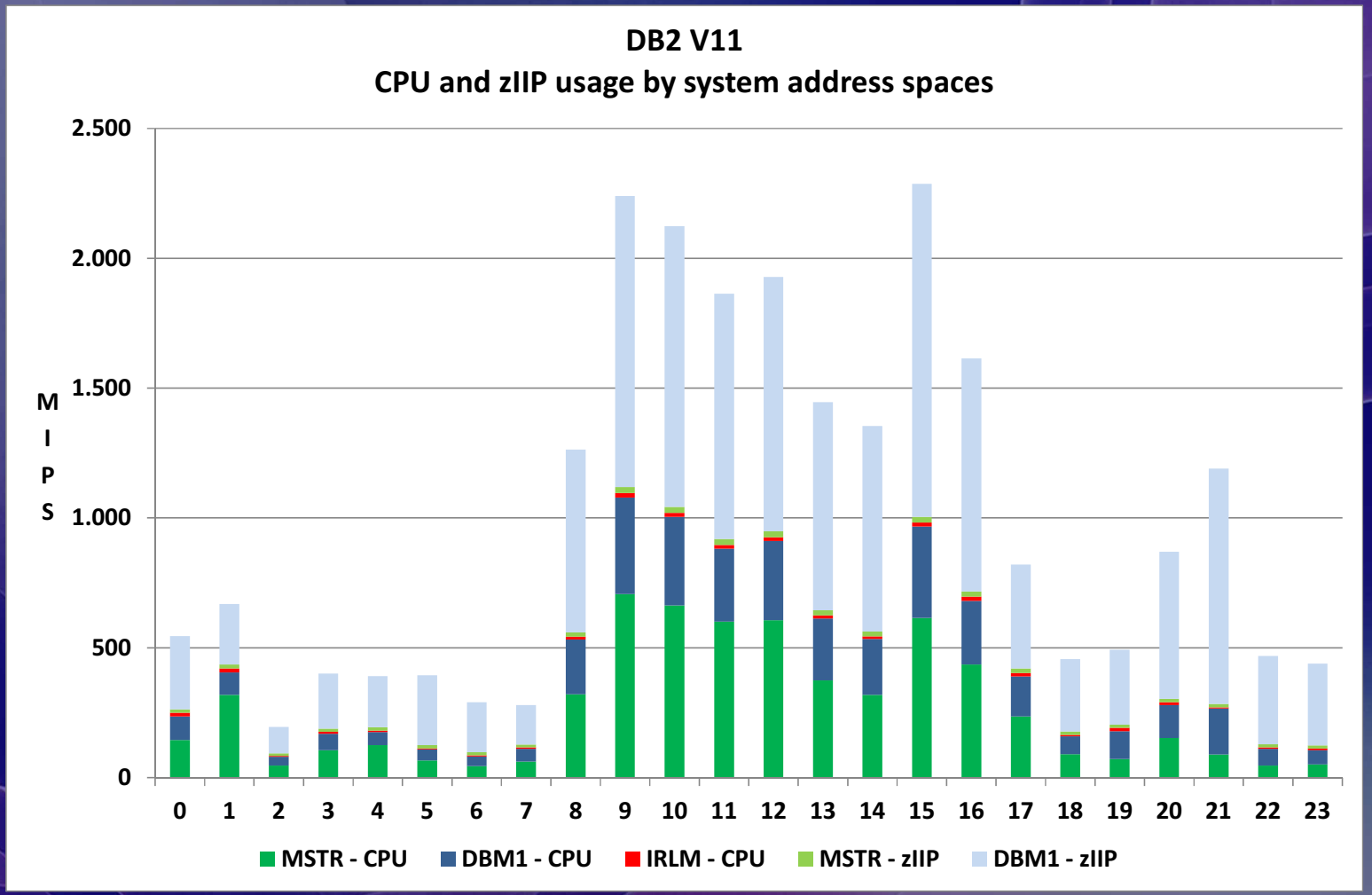
System AS – CPU Usage in V10 & V11

System AS	TCB mode	SRB mode
DBM1	Opening and closing of dataset	Deferred writes
	DBM1 full system contraction	Prefetch reads
	Preformat	Parallel child tasks
	Extend	Castouts
		Asynchronous GBP writes
		P-lock negotiation
		Notify Exit
		Page set close or pseudo-close to convert to non-GBP dependent
		GBP checkpoints
MSTR	Archiving	Physical log writes
	BSDS processing	Thread deallocation
		Update commit (including unlocking of page P-locks)
		Backouts
		Checkpoints
IRLM	Error checking	Local IRLM latch contention
	Management	IRLM and XES global contention
		Asynchronous XES contention
		P-lock negotiation
		Deadlock detection

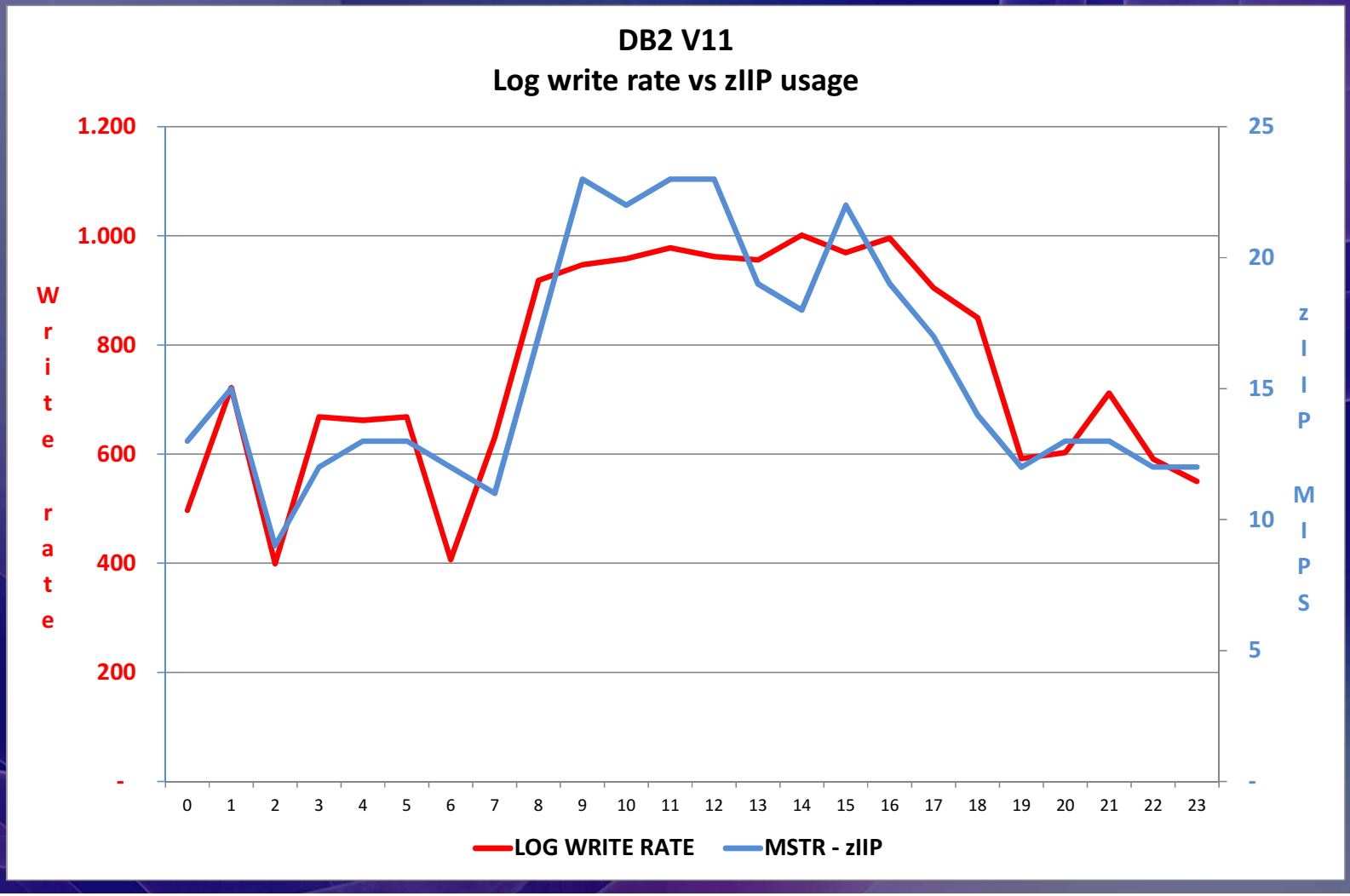
System AS – CPU usage in V10 & V11

- MSTR CPU usage higher in DB2 V10 and V11
- MSTR offload to zIIP in DB2 V11 generally not enough to balance that increase
- About 20 MIPS offloaded to zIIP per 1.000 write/sec
- See next slides

System AS – CPU Usage in V10 & V11



System AS – CPU Usage in V10 & V11



Evolution of MSTR functions

Evolution of MSTR functions

- Many functions have been extended and added to the MSTR AS in the recent DB2 versions
- The first new important function, introduced with DB2 V9, is the System Monitor task; this task continuously checks the health of the system at one-minute intervals trying to address the following major issues:
  - ✓CPU stalls in DB2 resulting in latch contention
  - ✓critical usage of DBM1 virtual storage below the 2GB bar (IFCID 225 produced by class 1 statistics trace)

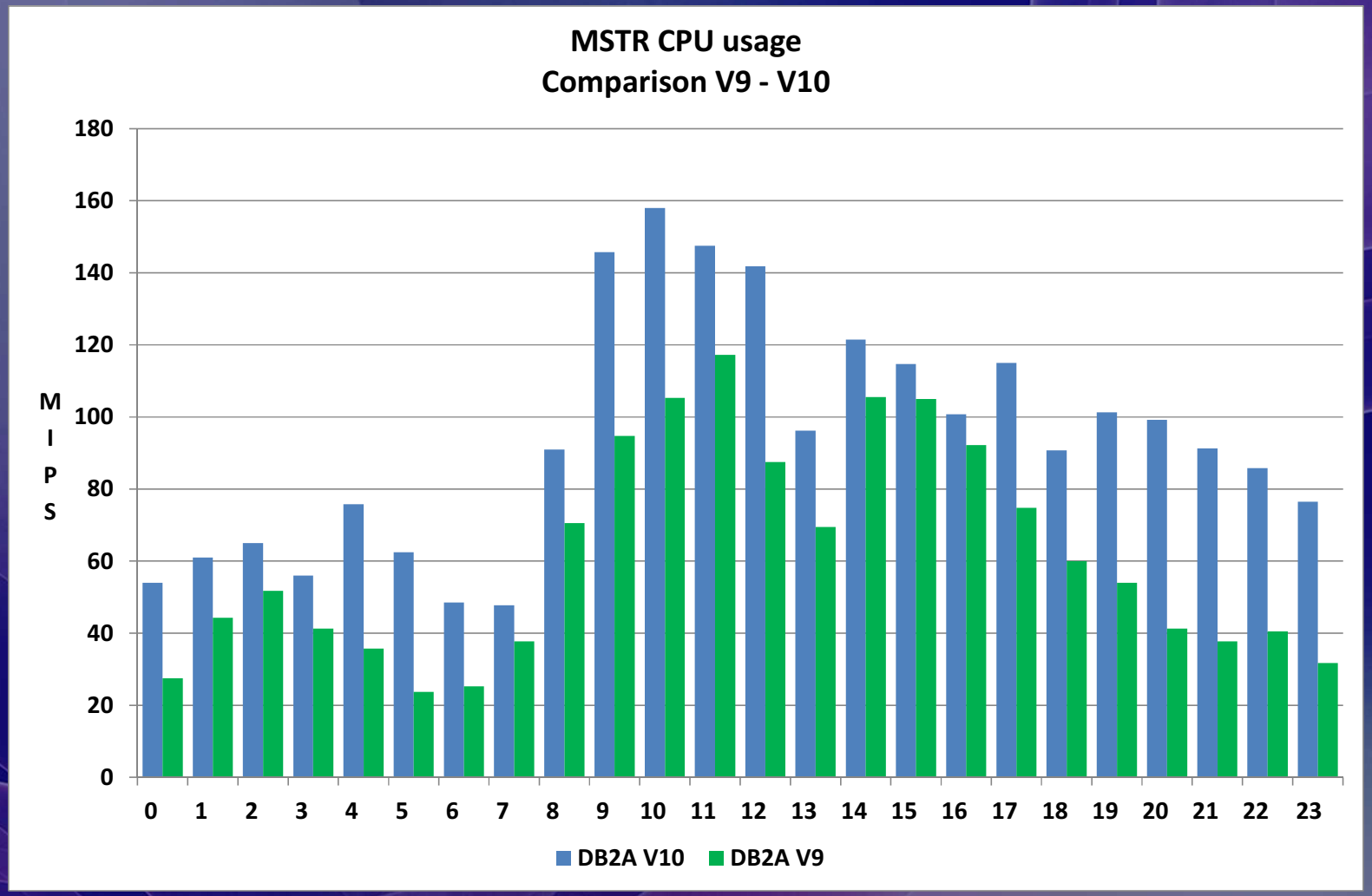
Evolution of MSTR functions

- With DB2 V10 and V11, virtual storage constraints have been practically eliminated by exploiting 64-bit virtual storage and allowing DB2 to use much more real storage to improve application performance
- So IBM introduced other new functions in MSTR to help customers monitor the real and auxiliary storage used by DB2 subsystems (many IFCID 225 enhancements)

Evolution of MSTR functions

- With DB2 V10 the amount of virtual storage that needs to be monitored is an order of magnitude higher than before
- An increase of MSTR CPU usage can be considered normal
- However by tracking MSTR CPU usage when migrating to DB2 V10 some customers discovered an excessive increase
- Most relevant corrections: DB2 APAR (PM49816) and z/OS APAR (OA37821)

Evolution of MSTR functions



Evolution of MSTR functions

- However the most important MSTR new function in DB2 V10 and V11 is the interaction with the z/OS RSM (Real Storage Manager) to control and manage the DB2 real storage
- Two new system configuration parameters are provided:
  - REALSTORAGE_MAX; this parameter sets the maximum of real and auxiliary storage (in GB) a DB2 subsystem can use; DB2 will terminate if this threshold is reached; new message DSNS003I will be written by MSTR when DB2 approaches the specified threshold
  - REALSTORAGE_MANAGEMENT; this parameter will tell DB2 how to manage thread storage pages that are backed in real storage but not used anymore

Evolution of MSTR functions

- DB2 thread storage is now allocated in a memory object in 64-bit shared storage
- To give back real storage frames backing virtual pages inside a memory object the IARV64 DISCARDDATA service has to be used together with the optional KEEPREAL parameter

Evolution of MSTR functions

- When DB2 uses DISCARDDATA with KEEPREAL(YES), the storage is only "virtually freed"; RSM flags the page as freed or unused, but the storage is still in real storage with the data
- Real memory statistics not always accurate
- When DB2 uses KEEPREAL(NO), RSM frees and reclaims the page immediately

Evolution of MSTR functions

- Depending on parameters settings and on the system condition customers experimented high CPU usage in MSTR or a high amount of memory apparently free but not usable by z/OS
- IBM provided some corrections trying to find the best compromise between CPU and memory usage

Evolution of MSTR functions

- This is the situation after APAR PM99575 :
  - ✓if REALSTORAGE_MAX boundary is approaching or z/OS has notified DB2 (through ENF 55 signal) that there is a critical auxiliary shortage; MSTR issues DISCARDDATA requests with KEEPREAL=NO
  - ✓if REALSTORAGE_MANAGEMENT is set to OFF, MSTR will not issue DISCARDDATA requests; more real storage is used;

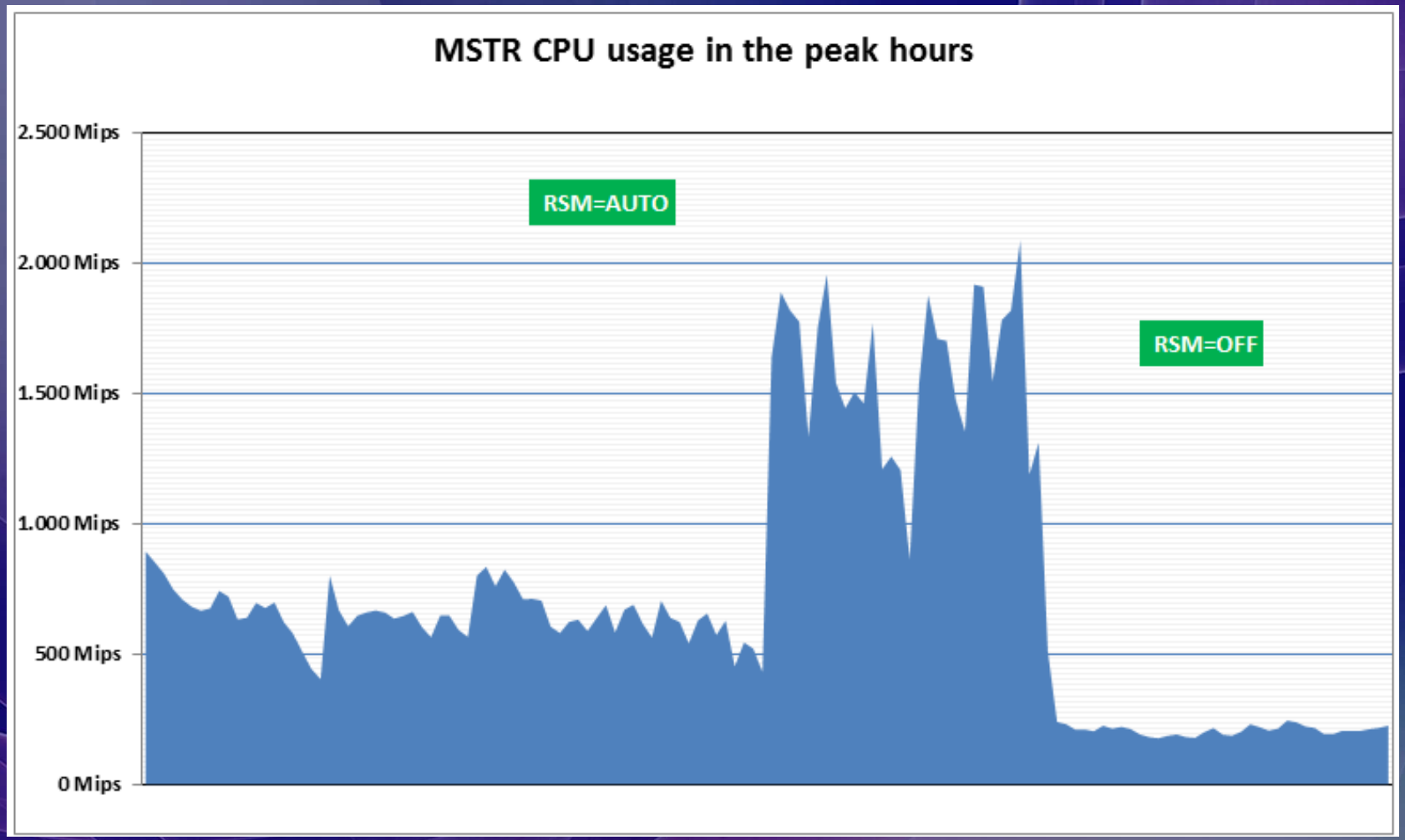
Evolution of MSTR functions

- ✓if REALSTORAGE_MANAGEMENT is set to AUTO (default) with no paging in the system, MSTR issues DISCARDDATA requests with KEEPREAL=YES to free storage at thread deallocation or after 120 commits; more CPU is used and charged to MSTR
- ✓if REALSTORAGE_MANAGEMENT is set to AUTO (default) with paging or REALSTORAGE_MANAGEMENT is set to ON, MSTR issues DISCARDDATA requests with KEEPREAL=YES to free storage at thread deallocation or after 30 commits; stack storage is also discarded; even more CPU is used and charged to MSTR

Evolution of MSTR functions

- Unfortunately even after all these fixes some of our customers complain about excessive real storage usage when REALSTORAGE_MANAGEMENT is set to OFF
- Running with the AUTO default should be the best option but one of our customers, running DB2 V11, had recently a very bad surprise

Evolution of MSTR functions



Conclusions

Conclusions

- DB2 evolution in the last years allowed more and more DBM1 and now MSTR logging activities to become zIIP eligible with positive effect on hardware and software costs
- On the other hand the exploitation of 64 bit virtual storage and the availability of much bigger quantities of real storage increased the functions to be performed by the MSTR AS
- Appropriate reporting has to be available to control and solve possible CPU and memory issues

Questions ?

fabio.ottaviani@epvtech.com
www.epvtech.com
www.tps-data.eu