Université du Mainframe 2013

4-5 avril

zEC12 : Des Mips oui... mais aussi beaucoup d'autres dispositifs de performance

Alain Maneville

Senior Certified I/T Specialist – zChampion

Jeudi 4 Avril 2013 – 15H00-15H55

Université du Mainframe 2013

4-5 avril

Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*BladeCenter®, DB2®, e business(logo)®, DataPower®, ESCON, eServer, FICON, IBM®, IBM (logo)®, MVS, OS/390®, POWER6®, POWER6+, POWER7®, Power Architecture®, PowerVM®, S/390®, System p®, System p5, System x®, System z®, System z9®, System z10®, WebSphere®, X-Architecture®, zEnterprise, z9®, z10, z/Architecture®, z/OS®, z/VM®, z/VSE®, zSeries®

The following are trademarks or registered trademarks of other companies.

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.
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.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.
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.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. 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 about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

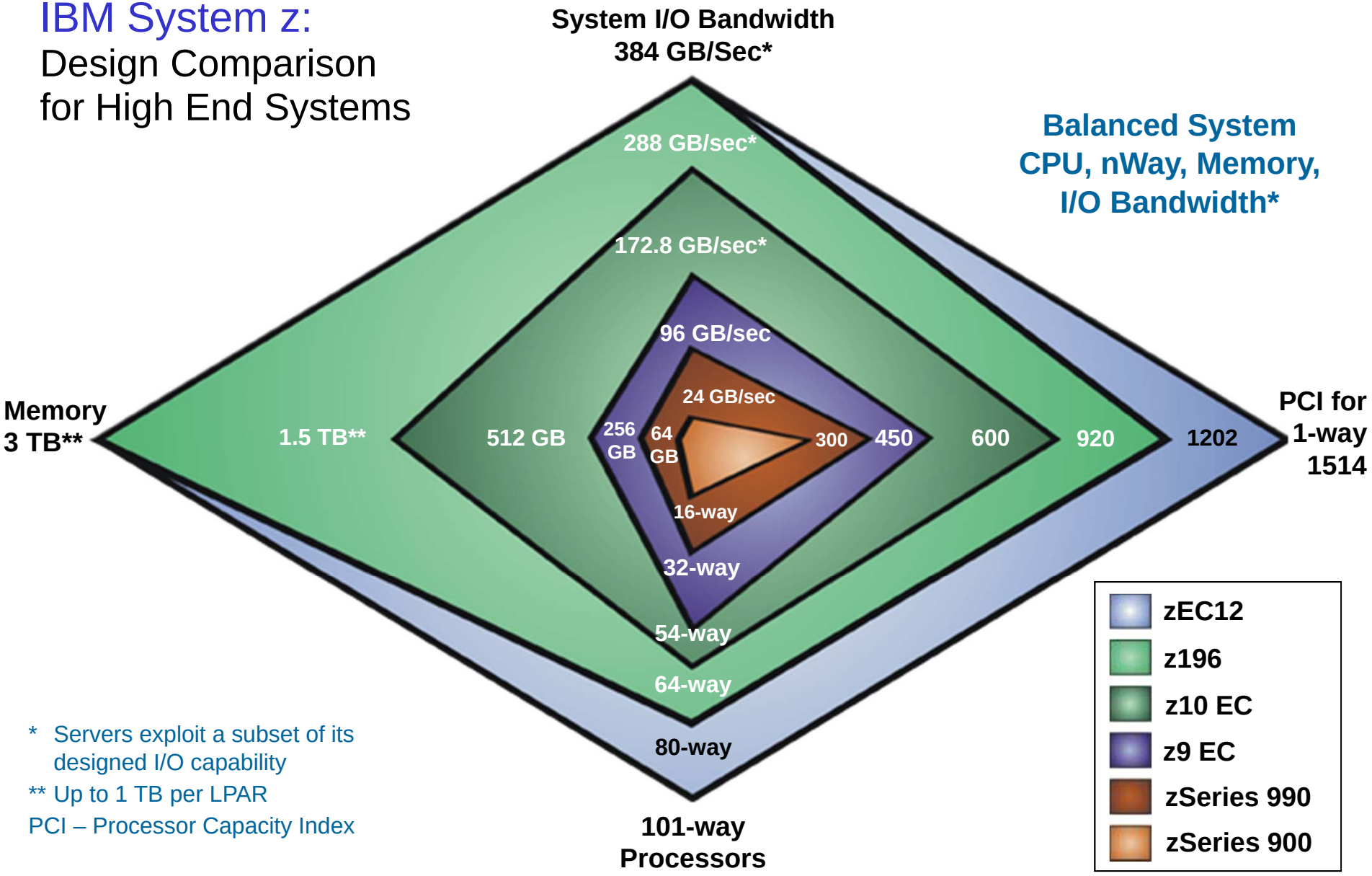
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

AGENDA

- Généralités sur la machine zEC12
- Structure du processeur
  - Extensions d'Architecture sur zEC12 - Transactional Execution
  - Extensions d'Architecture sur zEC12 – Out Of Order
  - 2 GB Pages
  - Warning Track Interruption Facility
- z196 and zEC12 System Compression et Cryptography Accelerator
- Les compilateurs PL1 et C/C++ sur zEC12
- Nouvelles instructions
- Annexe - Détail du Processing Unit

zEC12
Généralité sur la machine zEC12

IBM System z:
Design Comparison for High End Systems



* Servers exploit a subset of its designed I/O capability

** Up to 1 TB per LPAR

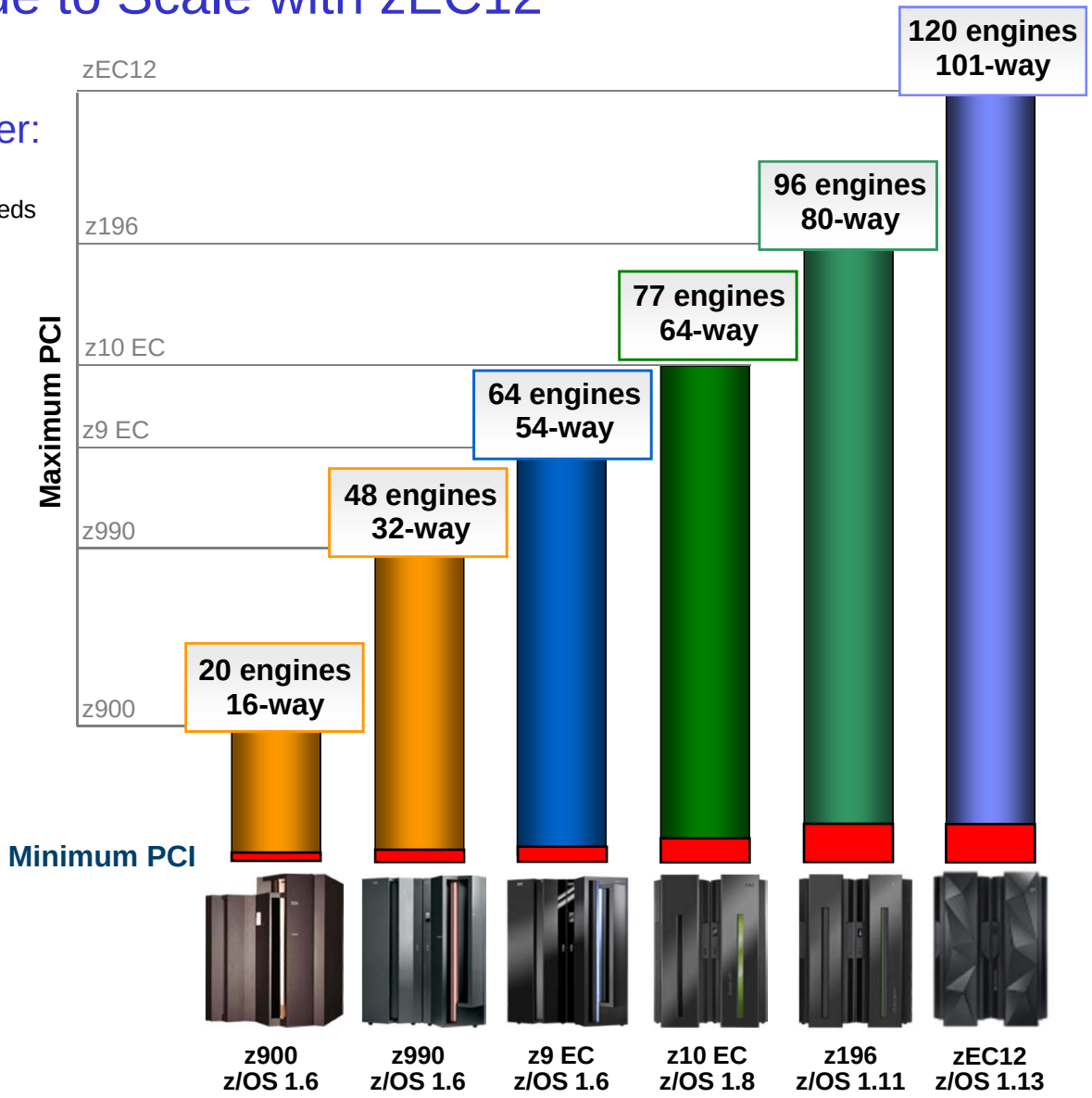
PCI – Processor Capacity Index

System z Servers Continue to Scale with zEC12

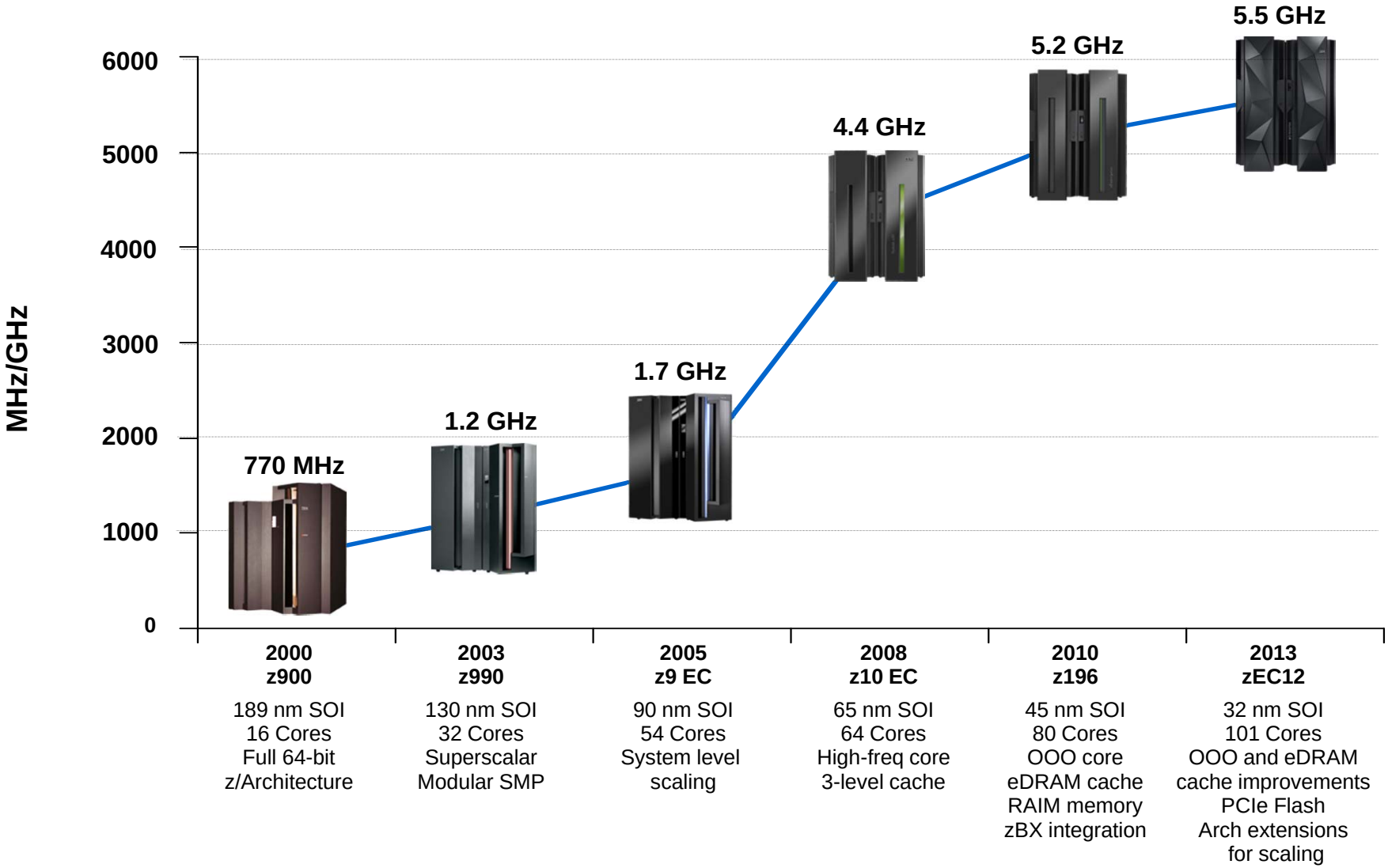
Each new range continues to deliver:

- New function
- Unprecedented capacity to meet consolidation needs
- Improved efficiency to further reduce energy consumption
- Continues to delivering flexible and simplified on demand capacity
- A mainframe that goes beyond the traditional paradigm

PCI - Processor Capacity Index



zEC12 Continue l'évolution CMOS commencée en 1994



zEC12 Overview

- Machine Type
  - 2827
- 5 Models
  - **H20, H43, H66, H89** and **HA1**
- Processor Units (PUs)
  - 27 (30 for HA1) PU cores per book
  - Up to 16 SAPs per system, standard
  - 2 spares designated per system
  - Dependant on the H/W model - up to 20, 43, 66,89, 101 PU cores available for characterization
    - Central Processors (CPs), Internal Coupling Facility (ICFs), Integrated Facility for Linux (IFLs), System z Application Assist Processors (zAAPs), System z Integrated Information Processor (zIIP), optional - additional System Assist Processors (SAPs)
  - Sub-capacity available for up to 20 CPs
    - 3 sub-capacity points
- Memory
  - RAIM Memory design
  - System Minimum of 32 GB
  - Up to 768 GB per book
  - Up to 3 TB for System and up to 1 TB per LPAR
    - 32 GB Fixed HSA, standard
    - 32/64/96/112/128/240/256 GB increments
  - **Flash Express**
- I/O
  - 6 GBps I/O Interconnects – carry forward only
  - Up to 48 PCIe interconnects per System @ 8 GBps each
  - Up to 4 Logical Channel Subsystems (LCSSs)
    - Up to 3 Sub-channel sets per LCSS
- STP - optional (No ETR)

zEC12
Structure du processeur

zEC12 architecture du PU core

- Dérivé du core z196
  - Amélioration du concept **Out of Order (OoO) execution**
  - Amélioration du **pipeline**, réduction des goulots d'étranglement
  - Amélioration du **Branch Prediction latency** et du débit de recherche des instructions
  - Peut décoder jusqu'à **3 instructions** par cycle et initialiser l'exécution jusqu'à **7 instructions** par cycle

- Dispositifs principaux
  - Nouveau design du 2nd-level
    - Cache structures différente pour les instructions et les opérandes (L2)
    - Plus grands caches (+33%) (1MB chacun) en amélioration la latence pour les L1 misses
  - Nouveau 2nd-level de Branch Prediction array
    - Amélioration de la capacité (24K branches) pour les grands programmes
  - Crypto / compression Co-processor par core
    - Démarrage plus rapide
    - Amélioration de la latence cache du CoP avec une réduction efficace lors des « misses »
    - Support de la conversion d'Unicode UTF8<>UTF16 (CU12/CU21 bulk improvements)
  - Support de nouveaux dispositifs architecturaux comme:
    - Transactional Execution (TX)
    - Run-time instrumentation (RI)
    - EDAT-2

zEC12 – Améliorations principales (comparées au z196)

- 50% plus de cores dans un chip CP
  - Amélioration de la fréquence de 5.7%
  - Amélioration de 25% de la capacité
- Caches plus grands et Latence plus faible
  - Total L2 / core plus grand de 33%
  - Total on-chip shared L3 est plus grand de 100%
  - Private L2 conçu pour réduire la latence des L1 miss de 45%
- 3eme Generation en High Frequency, 2eme Generation de Out of Order
  - De nombreuses améliorations du pipeline
  - # d'instructions « in flight » augmenté de 25%
- Nouveau 2nd level Branch Prediction Table
  - 3.5 fois plus de «branch »
- Multiples extensions d'architecture pour l'utilisation par le logiciel

zEC12 architecture du PU core

- Pourquoi un I-Cache et un D-Cache

Flot d'instructions

```
MVC    PARM_LEN(2),0(R3)
L      R4,PARM_LEN
LTR    R4,R4
BE     ER_PARM_LEN
C      R4,=F'13'
BNE    ER_PARM_LEN
```

I-CACHE

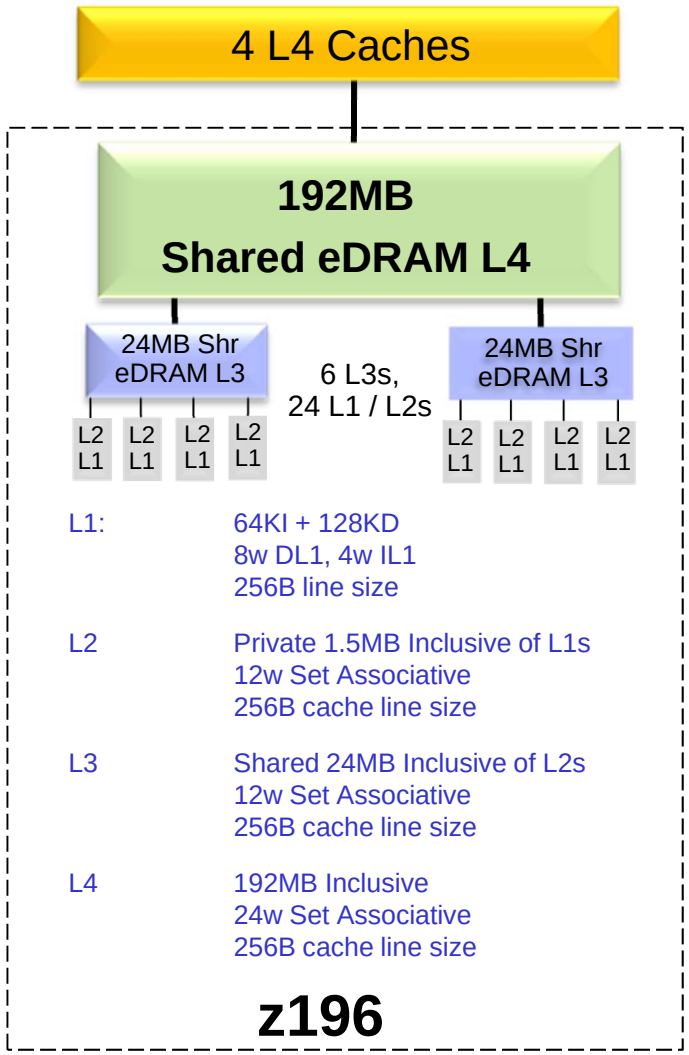
```
MVC    PARM_LEN(2),0(R3)
L      R4,PARM_LEN
LTR    R4,R4
BE     ER_PARM_LEN
C      R4,=F'13'
BNE    ER_PARM_LEN
```

D-CACHE

```
Contant[(PARM_LEN(2)]
Contenu[(0(R3)]
Contenu[PARM_LEN]
=F'13'
```

Le fait d'avoir un I-CACHE différent du D-CACHE permet de ne pas altérer le flot des instructions par le contenu des opérandes.
Sinon, il faudrait retourner en mémoire pour récupérer la suite du flot d'instructions

System z Cache Topology – z196 vs. zEC12 Comparison



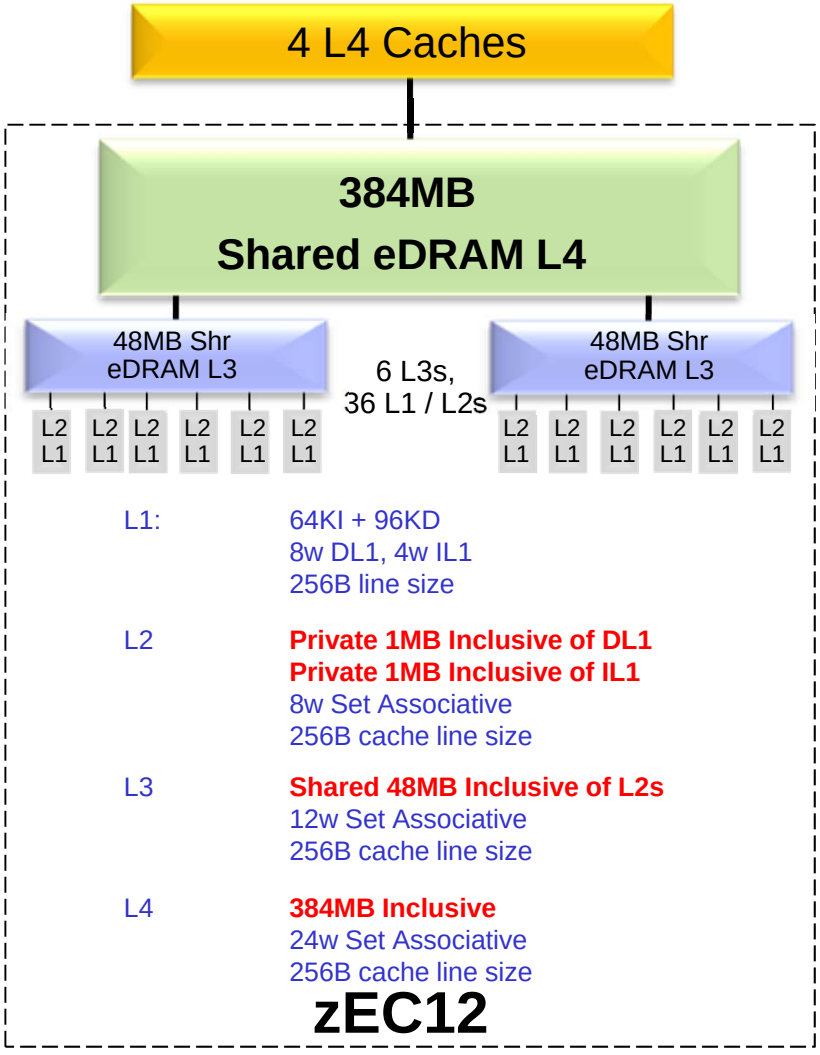
L1: 64KI + 128KD
8w DL1, 4w IL1
256B line size

L2 Private 1.5MB Inclusive of L1s
12w Set Associative
256B cache line size

L3 Shared 24MB Inclusive of L2s
12w Set Associative
256B cache line size

L4 192MB Inclusive
24w Set Associative
256B cache line size

z196



L1: 64KI + 96KD
8w DL1, 4w IL1
256B line size

L2 **Private 1MB Inclusive of DL1**
Private 1MB Inclusive of IL1
8w Set Associative
256B cache line size

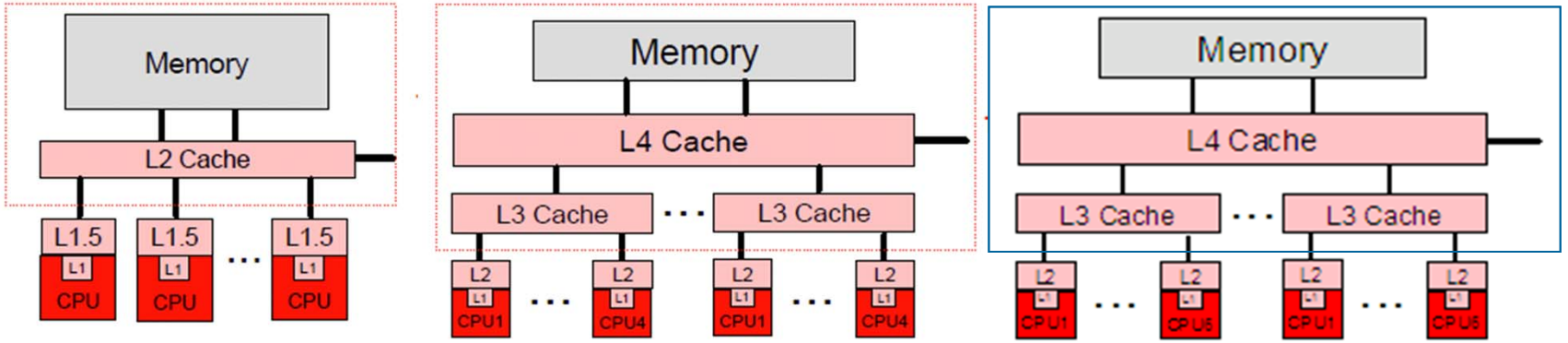
L3 **Shared 48MB Inclusive of L2s**
12w Set Associative
256B cache line size

L4 **384MB Inclusive**
24w Set Associative
256B cache line size

zEC12

La notion de Relative Nest Intensity

- La notion de Relative Nest Intensity est fondamentale pour le calibrage de la machine:
  - La zone la plus sensible à la performance de la hiérarchie mémoire est l'activité sur le « nid de la mémoire », à savoir la répartition de l'activité sur la mémoire cache partagée et la mémoire réelle.
  - Nous introduisons un nouveau terme, «Relative Nest Intensity » pour indiquer le niveau d'activité dans cette partie de la hiérarchie mémoire.
  - Plus le RNI est élevé, le plus profond dans la hiérarchie mémoire, le processeur doit aller récupérer les instructions et les données relatives à cette charge de travail.



Changement de la conception du Core comparé au z196

- Réduction des Cache Penalty
  - Nouvelle structure cache L1/L2 pour la réduction de la latence sur le L2 pour les « miss » instruction et data
    - Plus grand L2 (1M-Byte pour instruction et Data L2 cache – chacun) avec latence réduite.
  - D-TLB buffer pour améliorer le taux de hit
  - **Plusieurs HW prefetch et SW prefetch (PFD**)

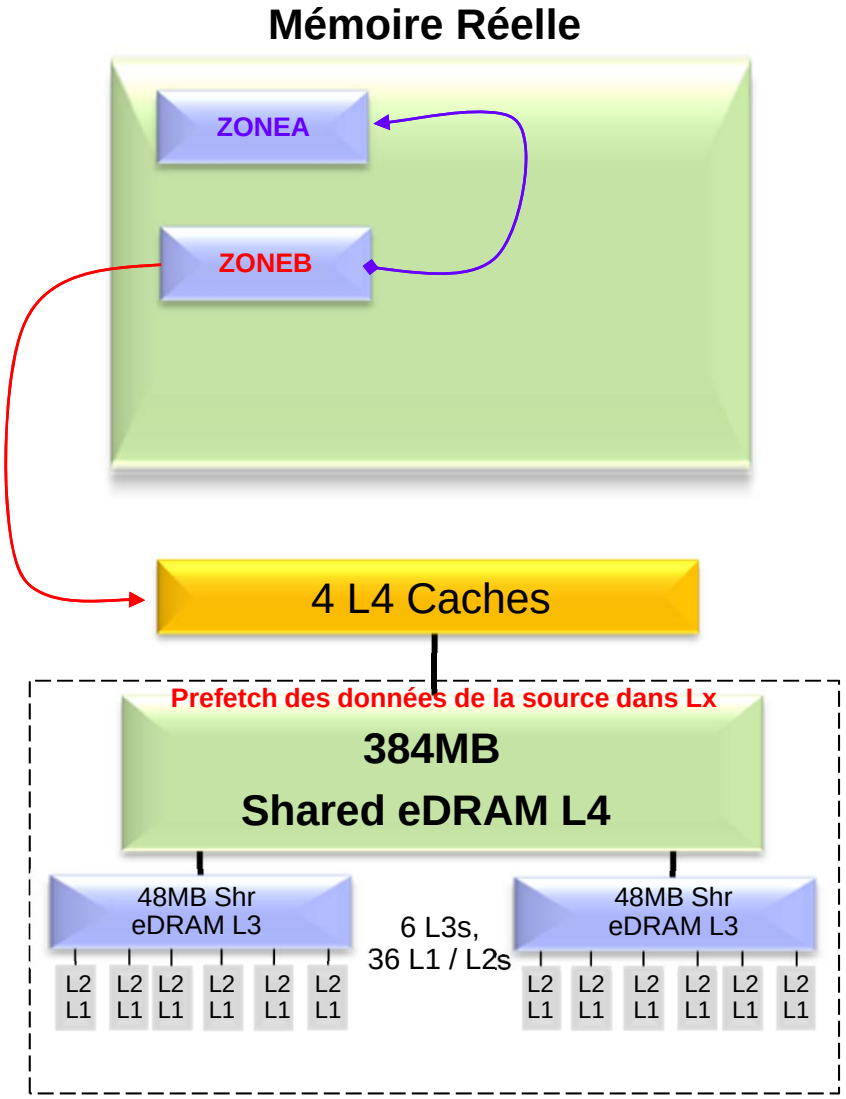
- Amélioration de la structure du Branch Prediction et Sequential Instruction Fetching
  - BTB (BTB2) secondaire fournissant **33% de Branch en plus**
  - Débit de prédiction plus rapide dans BTB1 en utilisant la table FIT (Fast re-Indexing Table)
  - Amélioration du débit séquentiel de traitement du flot des instructions

- Millicode
  - Exécution HW d'instructions précédemment exécutées en millicode (TR/TRT; STCK*)
  - Millicode simplifié pour réduire les interlocks hardware pour les instructions courantes:
    - e.g. MVCP
  - Amélioration du MVCL (voir exemple)
    - Aide millicode pour pré charger les data dans le L4 (en plus du pré chargement architecturé dans le L3)
    - Chevauchement de la prise en compte des « misses » entre:
      - L1/2<>L3
      - L3<>L4
      - L4<>Memory
    - Accélération des mouvement de données (alignés ou non) à travers les caches et la mémoire.

Changement de la conception du Core comparé au z196

- Détail MVCL

```
         LA    R4,ZONEA            R4<- ADRESSE CIBLE
         L     R5,LONGZA           R5<- LONGUEUR CIBLE
         LA    R6,ZONEB            R6<- ADRESSE SOURCE
         L     R7,LONGZB           R7<- LONGUEUR SOURCE
         ICM   R7,B'1000',BLANC    PADDING BLANC
*
         MVCL  R4,R6               ZONEB dans ZONEA
*
ZONEA    DS    2000C               CIBLE
ZONEB    DS    2000C               SOURCE
LONGZA   DC    A(2000)             LG CIBLE
LONGZB   DC    A(2000)             LG SOURCE
BLANC    DC    C' '                BLANC
```



Extensions d'Architecture sur zEC12

- Transactional Execution (a/k/a Transactional Memory) – voir exemple
  - Séquences définies par le SW traitées comme une seule "transaction" par le HW
  - Permet au logiciel d'être plus efficace
    - Application Hautement parallèles
    - Génération de code adaptée
    - Moins de « verrouillage » logiciel pour l'exécution du code
  - Prévu pour Java; a plus long terme pour DB2, z/OS, autres.
- Runtime instrumentation
  - Informations en Real-time sur les caractéristiques de programmes dynamiques
  - Permet l'optimisation de la recompilation dans JVM/JIT
- 2 GB page frames- voir exemple
  - Amélioration de l'efficacité pour les DB2 buffer pools, Java heap, et autres larges structures
- Directives du SW pour améliorer les performances du HW
  - **L'intention** d'utilisation des Data améliore la gestion des caches
  - **Branch pre-load** améliore l'efficacité du branch prediction
  - **Block prefetch** amène les données proches du processeur (en avance), réduisant la latence d'accès.
- Decimal format conversions
  - Permet une plus grande exploitation du Decimal Floating Point facility en COBOL

Transactional Execution (ou Transactional Memory)

Extensions d'Architecture sur zEC12 - Transactional Execution

- Définition
  - Innovation (concept vieux de 20 ans) qui permet à une séquence d'instructions de mettre à jour des données en mémoire de manière exclusive alors qu'on est dans un environnement multi-Thread.
  - Elle assure que, quand un ***Thread*** particulier essaye de mettre à jour les données en mémoire et que d'autres ***Threads indépendants*** mettent à jour simultanément de manière asynchrone ces données en mémoire, le ***Thread*** particulier est informé de cette condition.
  - Transactional Execution ne veut pas dire que la mémoire elle même est « transactionnelle » , mais qu'il existe une nouvelle façon par le Hardware d'accéder à cette mémoire, qui permet à cette mémoire d'être utilisée de cette façon pour les mises à jour.
  - La **TRANSACTION** est le bloc de code qui s'exécute sans « sérialisation ».
- Mise en œuvre
  - Permet au SW d'indiquer au HW le début et la fin d'un groupe d'instructions qui doit être exécuté de manière unitaire ou « ATOMIC ».
    - Une « Atomic Transaction » veut dire TOUT ou RIEN du point de vue de l'exécution.
  - Unité de travail exécutée entièrement ou réinitialisée aux conditions initiales.
    - Sauvegarde de la situation avant exécution dans une zone « Transactionnal Memory »
    - Rappel de cette sauvegarde si problème lors de l'exécution
  - Permet l'exécution sans « sérialisation » d'un « bloc » de code
- Intérêt
  - Sans « sérialisation » d'autres peuvent utiliser ce code ou **lire** le résultat (Multi Threading)
- Exemple
  - Gestion par z/OS des: Chaines, pointers, queues, I/O devices/paths

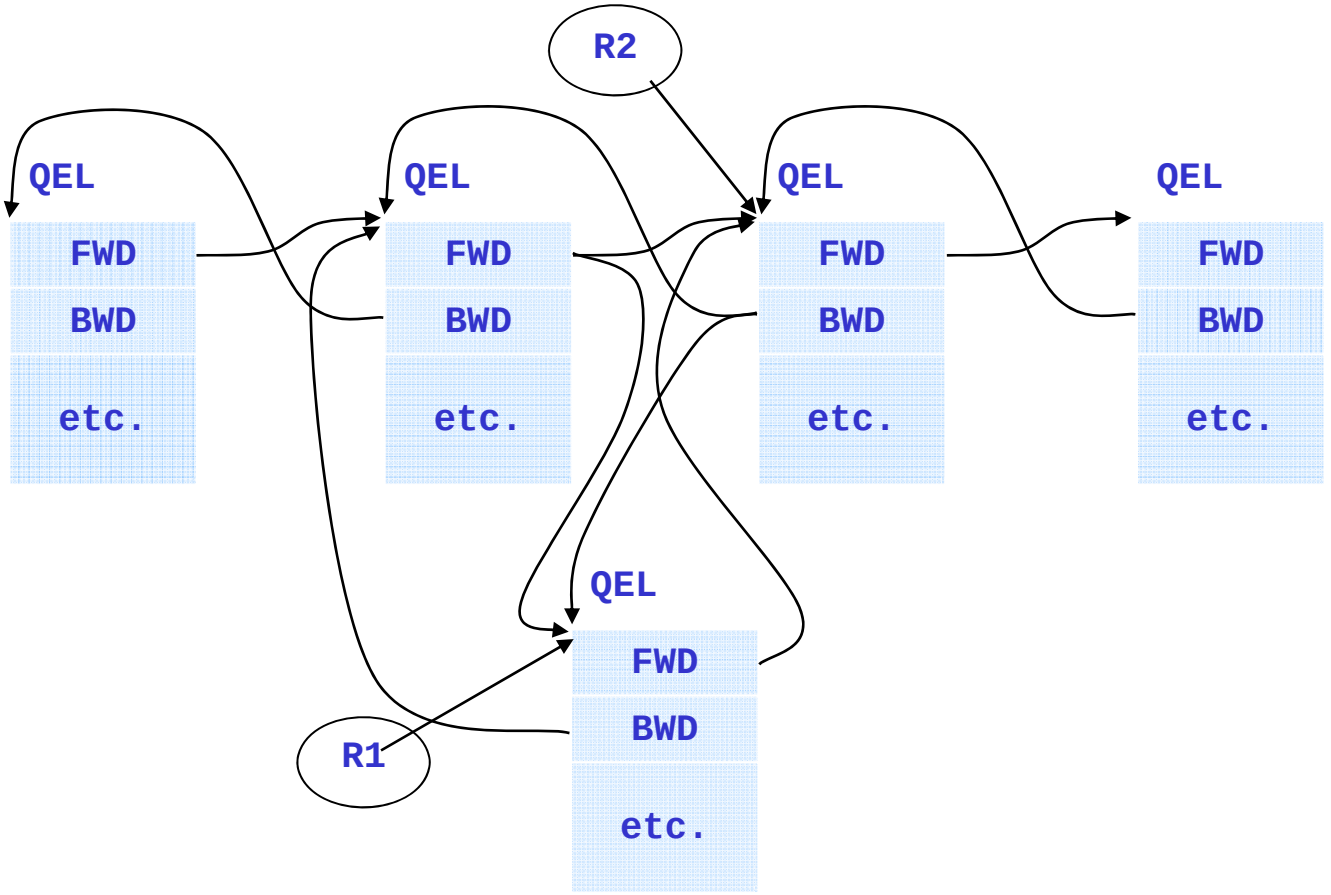
Extensions d'Architecture sur zEC12 - Transactional Execution

- A été mise en œuvre par du SW jusqu'à présent
  - Techniques incluant des locks, 2-phase commit/roll-back
  - La mise en œuvre SW est « pessimiste » - assume qu'aucun autre Thread n'exécutera ces instructions en se protégeant par des « Locks ».

- Une mise en œuvre correcte avec plus de parallélisme sur zEC12 car:
  - Les images z/OS doivent prendre en compte plus de:
    - Multiple cores
    - Multiple HW threads
    - Multiples SW threads
  - La mise en œuvre HW est optimiste et spéculative
    - RC si un autre THREAD exécute les instructions (chance …)

Extensions d'Architecture sur zEC12 - Transactional Execution

- Exemple – insertion d'un élément dans une liste à double lien (FWD, BWD)



Extensions d'Architecture sur zEC12 - Transactional Execution

- Résolution SW

```
* R1  - address of the new queue element to be inserted.
* R2  - address of the insertion point; new element is inserted
        before the element pointed to by R2.

NEW      USING   QEL,R1
CURR     USING   QEL,R2
         SETLOCK OBTAIN, …        Serialize access to queue (macro).


         LG      R3,CURR.BWD      Point to previous element.
PREV     USING   QEL,R3           Make it addressable.
         STG     R1,PREV.FWD      Update prev. forward ptr.
         STG     R1,CURR.BWD      Update curr. backward ptr.
         STG     R2,NEW.FWD       Update new forward ptr.
         STG     R3,NEW.BWD       Update new backward ptr.
         SETLOCK RELEASE, …
```

Note, SETLOCK est une *macro* Instruction

Extensions d'Architecture sur zEC12 - Transactional Execution

- Résolution avec Transactional Memory

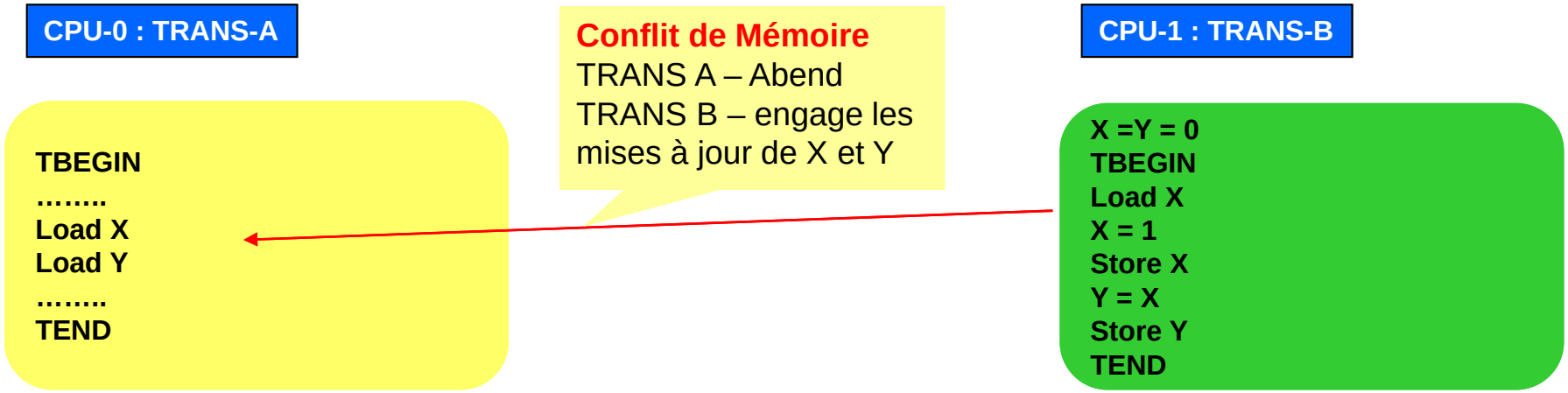
```
* R1  - address of the new queue element to be inserted.
* R2  - address of the insertion point; new element is inserted
        before the element pointed to by R2.

NEW      USING   QEL,R1
CURR     USING   QEL,R2
         LHI     R15,10          Load retry count.
LOOP     TBEGIN  TDB,X'C000'     Begin transaction (save GRs 0-3)
         JNZ     ABORTED         Nonzero CC means aborted.
         LG      R3,CURR.BWD     Point to previous element.
PREV     USING   QEL,R3          Make it addressable.
         STG     R1,PREV.FWD     Update prev. forward ptr.
         STG     R1,CURR.BWD     Update curr. backward ptr.
         STG     R2,NEW.FWD      Update new forward ptr.
         STG     R3,NEW.BWD      Update new backward ptr.
         TEND    TDB             End transaction.
         ...
ABORTED  JO      NO_RETRY        CC3: Nonretryable abort.
         JCT     R15,LOOP        Retry transaction a few times.
         J       NO_RETRY        No joy after 10x; do it the hard way.
```

Extensions d'Architecture sur zEC12 - Transactional Execution

- Nouvelles Instructions (AR10551 - replacing AR10440)
  - TBEGIN/TEND – indique le début et la fin de la transaction
    - Un conflit de mémoire est détecté par le HW si une autre CPU met à jour la zone mémoire
      - Voir exemple page suivante
    - Les Transactions peuvent être imbriquées.
    - Le dernier TEND entraine l'engagement des mises à jour.
  - TBEGINC – indicates the beginning of a constrained transaction
    - A constrained transaction should fit into a certain set of requirements
    - The hardware will use various algorithms under-the-cover to handle conflicts and to help its chance of successfulness
  - ETND – retrieves current transaction nesting depth
  - TABORT – deliberately causes a transaction to abort (from the outermost transaction)
  - NTSTG – Performs a store that will be committed regardless of whether the transaction aborts
    - mainly for SW debug
  - PPA – Perform processor assist
    - special algorithm implemented in millicode to improve chance of an unconstrained transaction to be successful after it is aborted

Extensions d'Architecture sur zEC12 - Transactional Execution

- Exemple de Conflit Mémoire potentiel
  - Dans cet exemple, un Conflit Mémoire entre deux transactions s'exécutant sur deux ***Cores*** entraine que la Transaction A sur le CPU 0 défaille, et par conséquent, les valeurs chargées X et Y sont rejetées. La transaction B sur le CPU 1 a réussi donc les valeurs mises à jour de X et Y sont engagés et mis à la disposition d'autres ***Cores*** après le TEND.



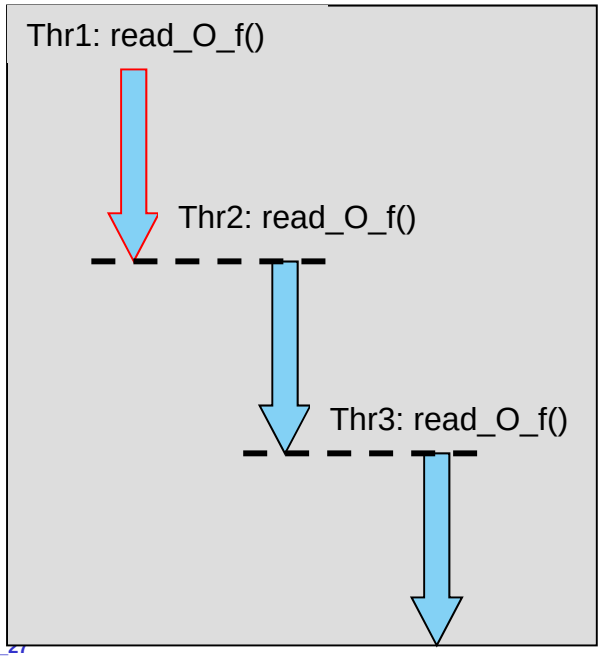
CPU-0 ne pourra voir que X=Y=0, ou X=Y=1
Mais jamais X=1 et Y=0

Extensions d'Architecture sur zEC12 - Transactional Execution

- Lock Elision

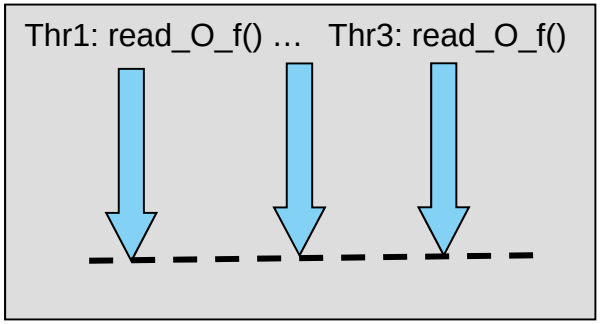
Threads doivent sérialiser même en lecture

```
read_O_f() {
  Wait_for_lock();
  read O.f;
  Release_lock();
}
```



Lock elision permet aux lecteurs une exécution //

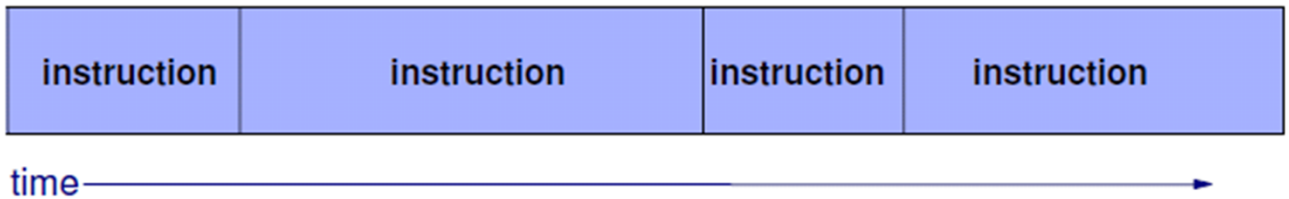
```
read_O_f()
  TRANSACTION_BEGIN
  read O.lock();
  BRNE use_lock
  read O.f
  TRANSACTION_END
```



Out Of Order Operation (OoO)

Extensions d'Architecture sur zEC12 – Out Of Order

- Généralités sur l'exécution des Instructions - base
  - Les Instructions sont exécutées dans l'ordre de leur séquence de codage.
  - Chaque instruction se termine avant que la prochaine commence.
  - Les Instructions ont un temps d'exécution variable.
  - Les Instructions ont un accès direct et immédiat à la mémoire centrale.



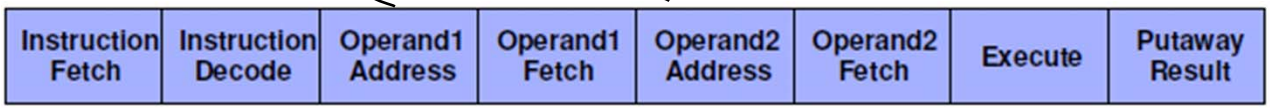
Mais ce n'est qu'une illusion

Extensions d'Architecture sur zEC12 – Out Of Order

- Généralités sur l'exécution des Instructions – la vue « PIPELINE »
  - L'exécution d'une instructions individuelle est vraiment une séquence d'activités dépendantes (en fonction du type d'instruction).

- Exemple : A R1,D2(X2,B2) R1=GPR1 – D2=déplacement – X2=GPR Index – B2=GPR base

- Exemple : CLC D1(L,B1),D2(B2) Operand1 = D1(B1) – Operand2=D2(B2)

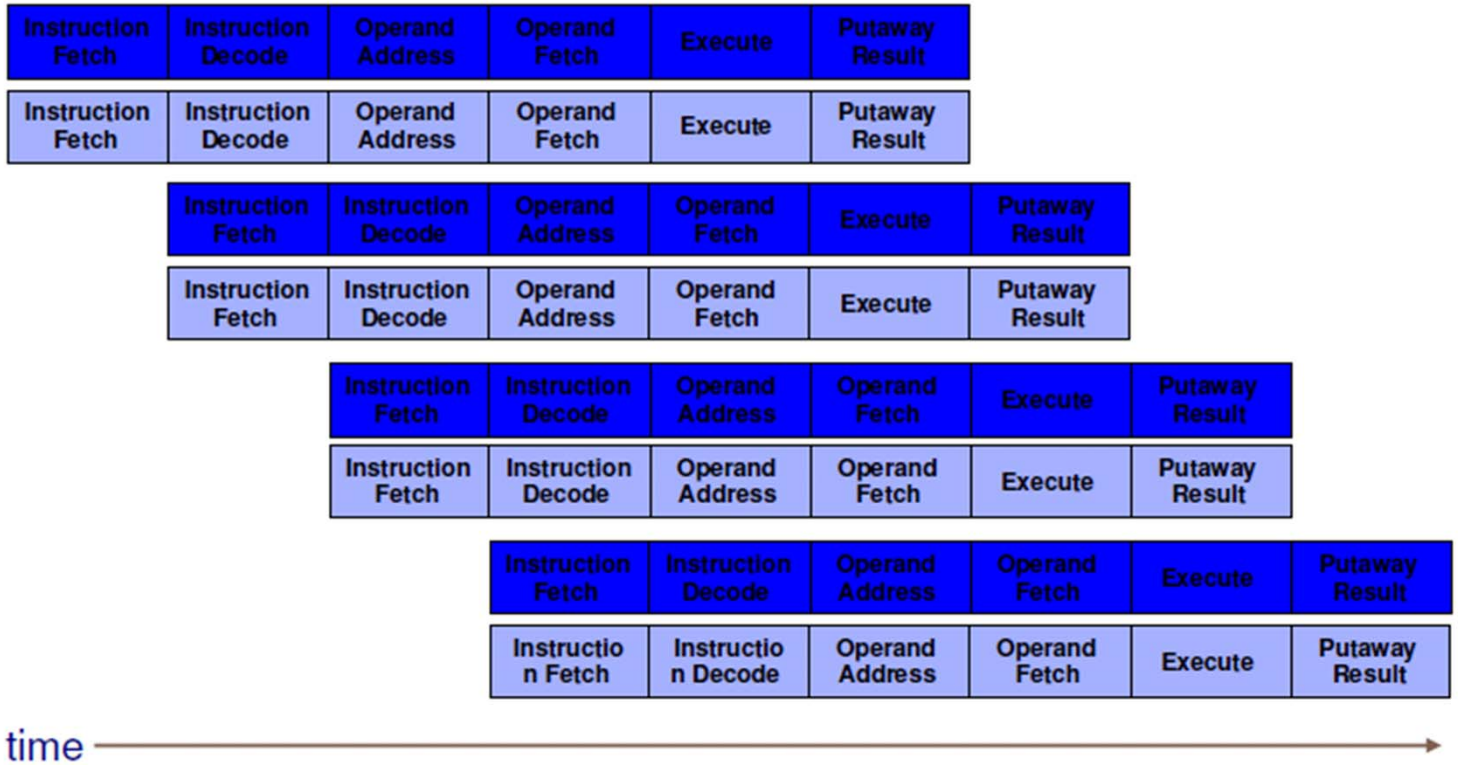


Extensions d'Architecture sur zEC12 – Out Of Order

- Généralités sur l'exécution des Instructions – la vue « PIPELINE »
  - Chaque étape de l'exécution d'une instruction est mis en œuvre par des composants distincts de sorte que **leur exécution peut se chevaucher**.

Extensions d'Architecture sur zEC12 – Out Of Order

- Généralités sur l'exécution des Instructions – la vue « PIPELINE » / Superscalar
  - Une machine SUPER SCALAR exécute plusieurs instructions simultanément car il a plusieurs unités pour chaque étape du PIPELINE. Mais l'ordre apparent des instructions est maintenu.



Extensions d'Architecture sur zEC12 – Out Of Order

- Out Of Order
  - Exécuter les instructions avant leur ordre normal d'exécution quand leurs interdépendances ont été résolues.
  - File d'attente de 40 instructions
  - Jusqu'à 72 instructions en cours

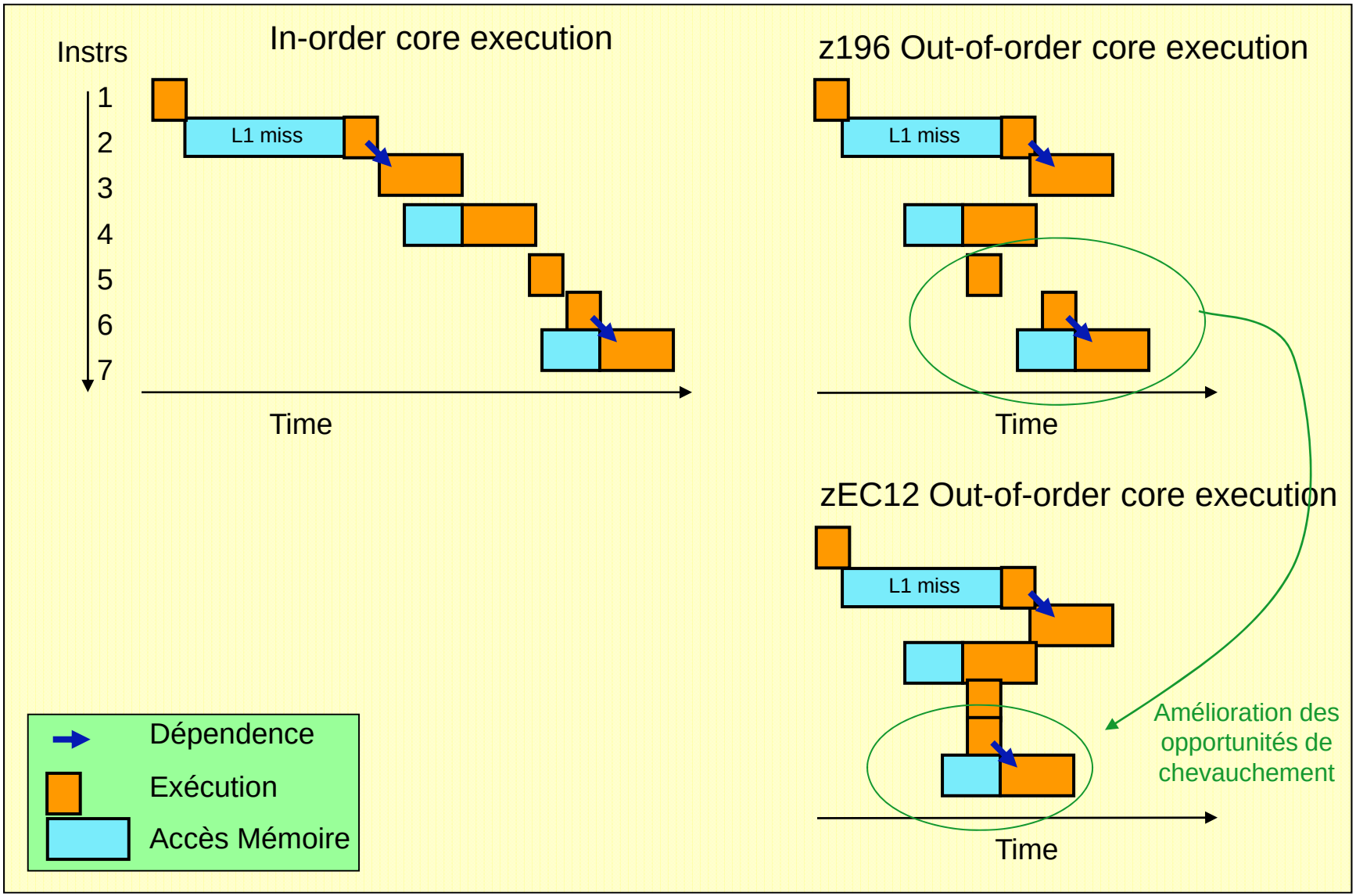
Extensions d'Architecture sur zEC12 – Out Of Order - Bénéfices

- Réorganiser l'exécution des instructions
  - Instructions bloquées dans le Pipeline car en attente du résultat d'une précédente instruction ou la ressource qu'elle nécessite est engagée.
  - Dans un in-order core, instruction bloquée, bloque toutes les futures instructions dans le flot du code
  - **Dans un out-of-order core**, les futures instructions sont autorisées à s'exécuter avant l'instruction bloquée

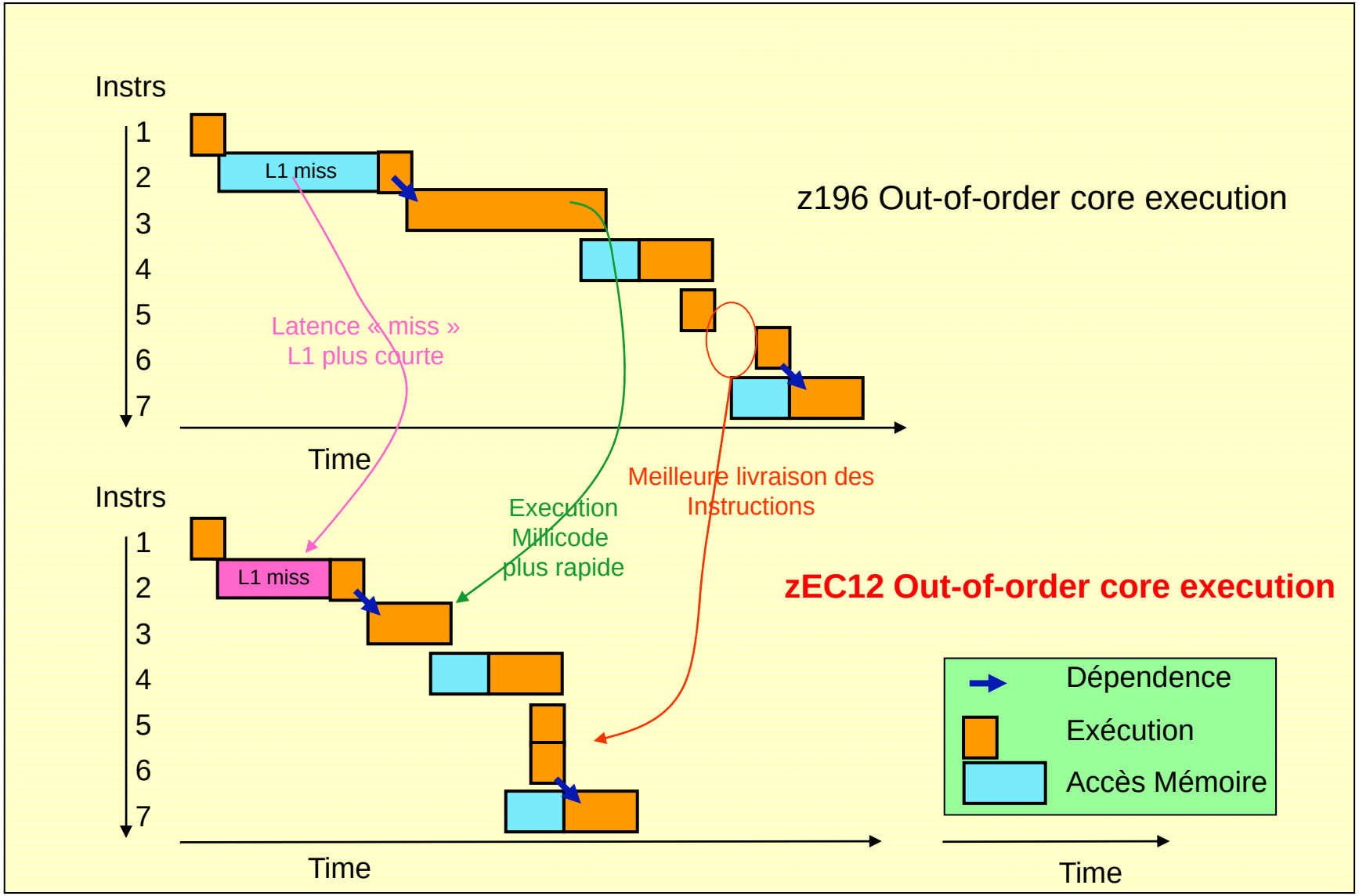
- Réorganiser l'accès à la mémoire
  - Instructions qui peuvent être bloquées par l'accès à la mémoire car elles attendent un résultat nécessaire à calculer cet accès mémoire.
  - Dans un in-order core, les instructions futures sont bloquées
  - **Dans un out-of-order core**, les futures instructions qui peuvent calculer leur adresse mémoire sont autorisées à s'exécuter.

- Masquer la latence d'accès à la mémoire
  - De nombreuse instructions accèdent aux données via la mémoire
  - Les accès mémoire peuvent avoir un L1 miss et nécessiter 10 à 500 cycles additionnels pour retrouver les données en mémoire
  - Dans un in-order core, les futures instructions du flot du code sont bloquées.
  - **Dans un out-of-order core**, les futures instructions qui ne sont pas dépendantes de cet accès mémoire sont autorisées à s'exécuter.

Extensions d'Architecture sur zEC12 – OoO– z196 % zEC12



Extensions d'Architecture sur zEC12 – OoO – Exécution améliorée



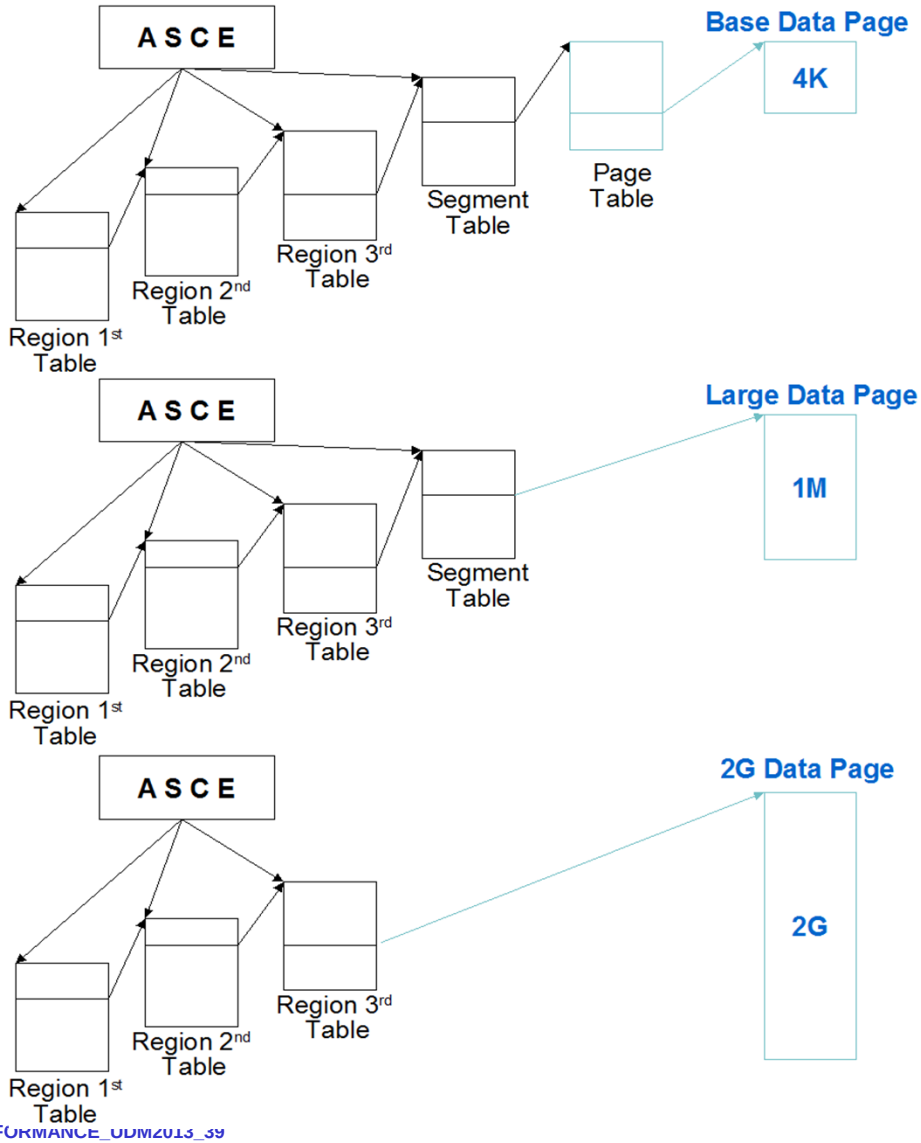
2 GB Pages

2 GB Pages

- Amélioration de la couverture du TLB sans augmentation proportionnelle de sa taille en utilisant les pages de 2 GB:
  - Une page de 2 GB est une page mémoire qui :
    - Est 2048 fois plus grande qu'une Large Page (1MB) et
    - 524288 fois plus grande qu'une page de base (4K)
  - Une page de 2 GB permet à une seul entrée de TLB de remplir beaucoup plus d'Address Translations qu'avec les Large pages ou les pages de base
  - Plus :
    - Meilleure performance en diminuant le nombre de TLB misses pour une application.
    - Moins de temps de conversion de Virtual Addresses en Physical Addresses
    - Moindre utilisation de mémoire réelle pour les structures DAT
- Dispositif EDAT-2
  - 2G translations seront stockées dans les TLB2
  - Utilisation de la Region-Third-Table-Entry (RTTE)

Le mécanisme du D A T

- Le processus – Différence entre PAGES de 4K, PAGES de 1MB et PAGES de 2GB



Run-time Instrumentation

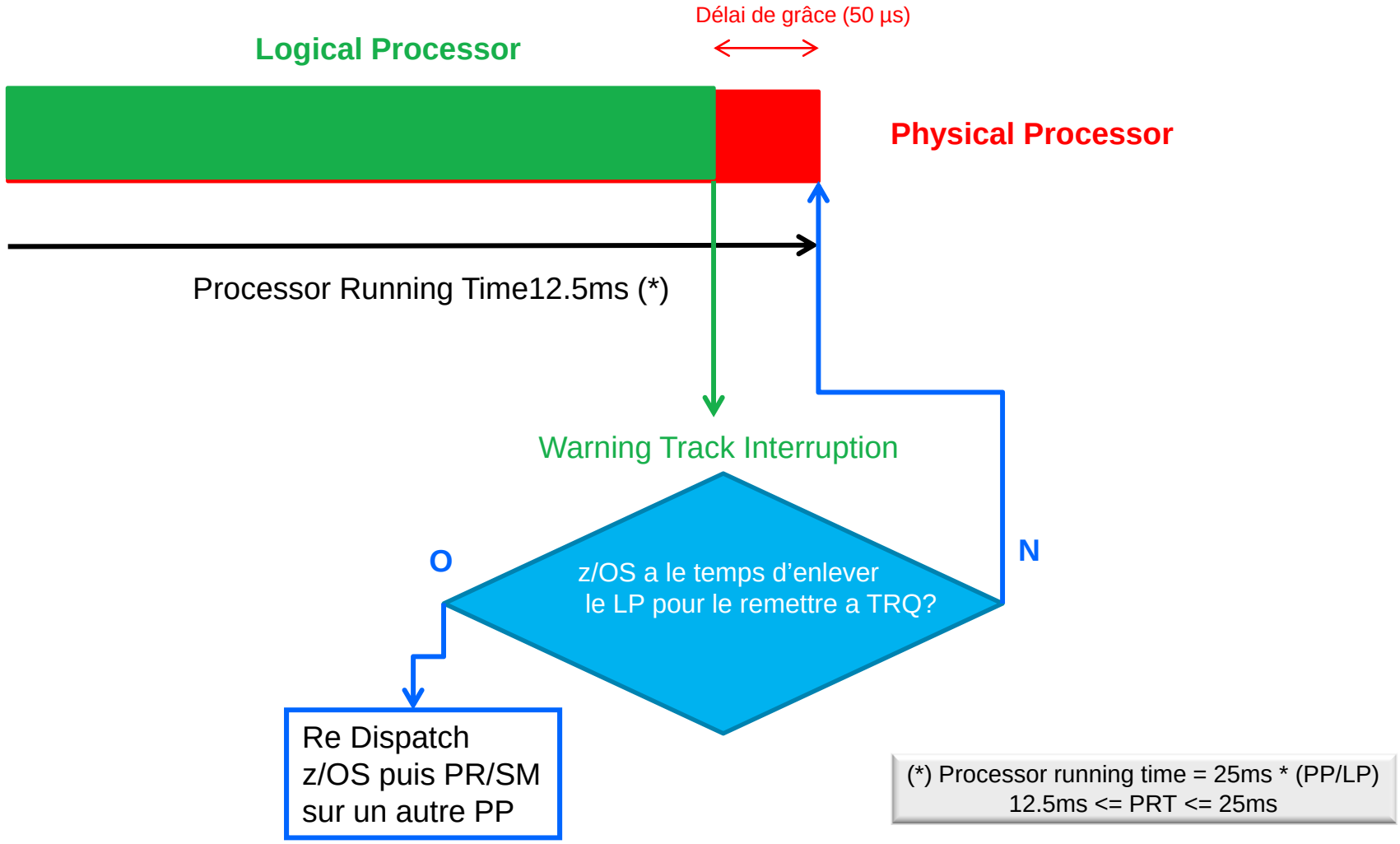
Extensions d'Architecture sur zEC12 – Run Time Instrumentation

- Run-time Instrumentation
  - Nouveau dispostif HW pour gérer les « runtimes »
    - Adaptés à Java Runtime Environment (JRE)
    - Dynamique et auto-tuning pour la recompilation OnLine
  - Pas le même dispositif que CPU Measurement Facility (CPUMF)
    - Peuvent s'exécuter en concurrence
  - Permettre l'optimisation dynamique sur la génération de code en cours d'exécution
    - Moins d'overhead que le « software-only profiling » actuel.
    - Donne des informations sur le HW ainsi que sur les caractéristiques du programme
    - Améliore les prises de décision de JRE en donnant un feedback immédiat
- Points clé
  - Un buffer de collecte capturant une trace run-time jusqu'à un point d'échantillonnage en fournissant:
    - "comment sommes nous arrivés la ", ex: Branch History
    - Valeur de profiling (des GPR) dans le contexte d'un chemin tracé
  - Meta-data collectées pour les informations de « ce qui s'est passé » avec le point d'échantillonnage
    - Cache miss
    - Branch prediction/resolution
  - 3 modes d'échantillonnage – par:
    - cycle count, instruction count, ou indications explicites

Warning Track Interruption Facility
APARS OA37186 et OA37803

Warning Track Interruption Facility

- Généralité sur le dispositif



Warning Track Interruption Facility

- Généralité (Principle Of Operation)
  - PR/SM reconnaît si un CP logique doit être "enlevé" d'un CP physique et émet une interruption d'avertissement (WTI, alias EXT 1007) et fixe un délai de grâce à z / OS pour retourner le CP logique à PR/SM
    - Si le "délai de grâce" expire avant que z/OS retourne le CP logique à PR/SM, PR/SM "undispatche" le CP logique et le redispatchera plus tard.
  - Cette « interruption » fournit les moyens par lesquels une interruption d'avertissement (Warning-Track) peut être présentée à un processeur dans une configuration partagée (une partition logique).
  - Le programme de contrôle peut utiliser l'interruption comme signal pour faire que l'unité de travail en cours de « dispatch » puisse être exécutée sur un CPU différent de la configuration.
  - Cette interruption est envoyée pour informer un programme qui arrive à la fin de son intervalle d'exécution (Time Slice).
    - Le but est que le programme (SCP) « dispatche » maintenant cette unité de travail sur un autre CPU de la configuration.
    - Par ailleurs, l'intention du dispositif est de présenter une interruption a une CPU en polarisation verticale (MEDIUM ou Low-Unparked).

Warning Track Interruption Facility

- Généralité (vue z/OS et WLM)
  - "Warning Track est une nouvelle possibilité qui permet à PR/SM, de signaler par une interruption extérieure au système d'exploitation que la tranche de temps (PR/SM), prendra fin sous peu.
  - Quand z/OS reçoit un WTI:
    - Sauvegarde l'état du travail en cours (rend le travail disponible à être redispatché sur un autre CP)
    - Envoie un DIAG 49C pour retourner le CP à PR/SM
      - Il devient de la responsabilité de PR/SM de redispatcher le CP logique quand c'est possible et de reprendre l'exécution à l'instruction après DIAG 49C
    - z/OS assure le suivi des statistiques suivantes:
      - Combien de DIAG 49Cs ont réussi / pas réussi à retourner le CP à PR/SM avant la fin de la période de grâce
      - Combien de temps PR/SM a "undispatché" le CP logique pour les DIAG 49Cs réussi
  - Suite à cette interruption le Dispatcher peut "enlever" le travail du processeur logique en cours (à condition que la DU soit préemptible).
  - Cette possibilité est utilisée sur les processeurs VM et VL (Unparked) en particulier pour éviter à une Dispatchable Unit (DU) d'être "verrouillée" sur un CP logique qui risque de ne pas recevoir un autre Time Slice de PR/SM pendant un certain temps.
  - Ainsi, la valeur de Warning track, c'est qu'il aborde l'inconvénient du Dispatching sur un VL(Unparked) et permet donc d'utiliser la capacité "Discretionary" de façon plus fiable (par exemple, sans impact sur les temps de réponse).

Warning Track Interruption Facility

- Utilisation par HiperDispatch
  - Vous avez une unité de travail en cours d'exécution sur un LP (VM ou VL Unparked) et que ce LP arrive à la fin de son TIME SLICE OU elle est exécuté sur un PP qui appartient à un LP (VH) et un LP veut maintenant utiliser "son" PP:
    - Le moteur physique est enlevé de ce LP.
    - Du point de vue de z/OS, le travail est toujours en cours.
    - Ce travail ne peut pas se terminer tant que le LP n'est pas re-dispatché.
    - Selon l'utilisation de la machine et la priorité relative de ce LP (qui est probablement faible s'il utilise la totalité de son TIME SLICE), il pourrait prendre un certain temps avant qu'il ne soit dispatché à nouveau.
    - Tout autre travail dans le système qui est en attente que ce travail soit terminé doit maintenant aussi « attendre ».
  - Pour éviter ce genre de situation, le zEC12 fourni cette interruption – Warning Track Interrupt
    - Indique à un LP qu'on va lui prendre le PP sur lequel il est en cours d'exécution
  - PR/SM
    - Envoie cette interruption à un VM (ou un VL Unparked) quand:
      - Le LP arrive à la fin de son TIME SLICE (généralement 12.5 ms)
      - Le LP s'exécute sur un PP normalement attribué à un VH et doit rendre ce PP
  - z/OS
    - Remet ce travail en file d'attente et relâche le PP.
    - Plutôt que d'avoir à attendre que le LP soit à nouveau dispatché, le travail interrompu peut continuer à s'exécuter et se terminer sur un autre LP.

Warning Track Interruption Facility

- Mise en œuvre
  - z/OS V1R12 et V1R13+
  - HIPERDISPATCH=YES
  - Support z/OS avec APAR OA37186
  - Support RMF avec APAR OA37803
  - Support SMF (SMF 70)
    - SMF70WTS
      - Number of times PR/SM issued a warning-track interruption to a logical processor and z/OS **was able** to return the logical processor within the grace period.
    - SMF70WTU
      - Number of times z/OS **was NOT able** to return the logical processor within the grace period.
    - SMF70WTI
      - Total amount of time (in milliseconds) that a logical processor was not dispatched on a physical CP.

Warning Track Interruption Facility

- Mise en œuvre
  - RMF
    - Utiliser les champs OVERVIEW (pas de report direct avec RMF III ni RMF PP)
    - WTRKCP / WTRKAAP / WTRKIIP
      - Percentage of times PR/SM issued a warning-track-interruption to a GCP/AAP/IIP and z/OS was able to return it to PR/SM within the grace period.
      - Source:
        - » SMF70WTS et SMF70WTU
      - Calcul:
        - » WTS / (WTS + WTU)
    - WTRKTCP / WTRKTAAP / WTRKTIIP
      - Time in milliseconds that a GCP/AAP/IIP was not dispatched on a physical GCP/AAP/IIP by PR/SM due to warning- track processing..
      - Source
        - » SMF70WTI

Warning Track Interruption Facility

- Mise en œuvre
  - RMF
    - Exemple:
      - OVERVIEW(REPORT)
      - OVW(WTRKCPP(**WTRKCP**))
      - OVW(WTRKTCPT(**WTRKTCP**))

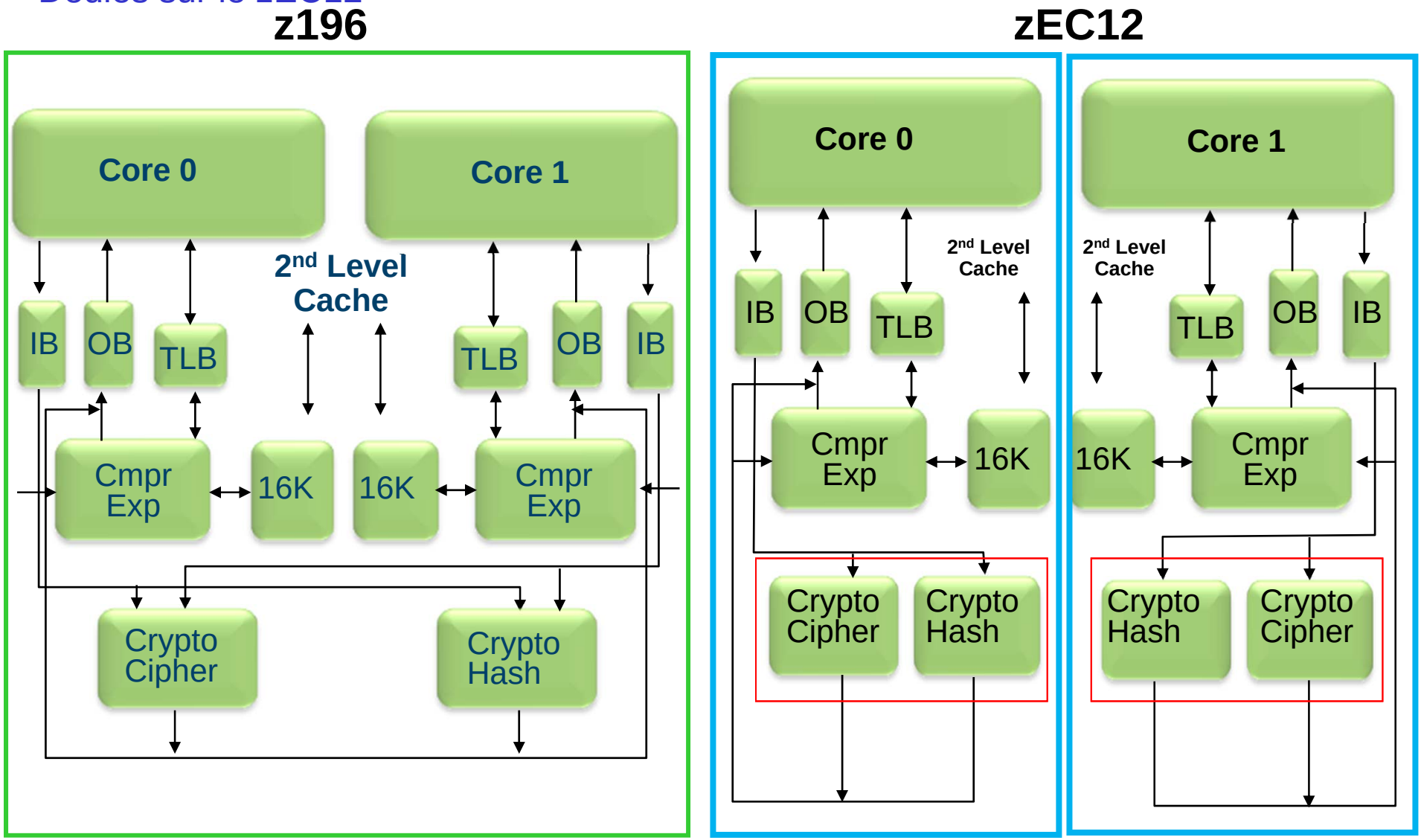
```
                                R M F   O V E R V I E W   R E P O R T

            z/OS  V1R13             SYSTEM ID SC04            START 02/01/2013-00.59.35  INTERVAL 00.10.00
                                    RPT VERSION V1R13 RMF     END   02/01/2013-03.39.36  CYCLE 1.000 SECONDS

 NUMBER OF INTERVALS 15           TOTAL LENGTH OF INTERVALS 02.30.00
-DATE    TIME     INT       WTRKCPP  WTRKTCPT
 MM/DD HH.MM.SS HH.MM.SS
 02/01 01.09.35 00.10.00       67.4        25
 02/01 01.19.35 00.10.00       50.5        16
 02/01 01.29.35 00.09.59       61.5        30
 02/01 01.39.35 00.10.00       56.8         9
 02/01 01.49.35 00.09.59       59.6        13
 02/01 01.59.35 00.10.00       56.7        36
 02/01 02.09.35 00.09.59       65.3        19
 02/01 02.19.35 00.09.59       56.1        25
 02/01 02.29.35 00.10.00       57.6        22
 02/01 02.39.35 00.09.59       62.7        13
 02/01 02.49.35 00.09.59       59.5         8
 02/01 02.59.35 00.10.00       55.6        12
 02/01 03.09.35 00.10.00       61.3        11
 02/01 03.19.35 00.09.59       63.5        46
 02/01 03.29.35 00.10.00       58.3        23
```

Le LP n'a pas été dispatché par PR/SM pendant 23ms .

z/OS a réussit à redonner le LP à PR/SM avant la fin du TS 58.3% des cas.

zEC12
System Compression et Cryptography Accelerator

z196 and zEC12 System Compression et Cryptography Accelerator Dédiés sur le zEC12



zEC12
Annexe – Les compilateurs PL1 et C/C++
Performance sur les Middlewares

Annexe – Les compilateurs PL1 et C/C++ / Middlewares

- Mettre à jour le compilateur au niveau PL/I for z/OS V4R3
  - **Option ARCHITECTURE(10)**
  - Utilisation des nouvelles instructions comme:
    - Decimal-Floating-Point Zoned-Conversion Facility
  - Optimisation du code généré.
    - Performant dans les programme utilisant PICTURE et DFP

  - Exemple (*compilé une fois avec ARCH(9) puis avec ARCH(10))* *1/3*

```
ARCH(9)
*process float(dfp);
 pic2dfp: proc( ein, aus ) options(nodescriptor);
   dcl ein(0:100_000) pic'(9)9' connected;
   dcl aus(0:hbound(ein)) float dec(16) connected;
   dcl jx  fixed bin(31);
   do jx = lbound(ein) to hbound(ein);
     aus(jx) = ein(jx);
   end;
 end;
```

Annexe – Les compilateurs PL1 et C/C++ / Middlewares

- Mettre à jour le compilateur au niveau PL/I for z/OS V4R3
  - Exemple - **ARCH(9)** 2/3 – Généré assembler (17 instructions)

```
0060    F248    D0F0    F000    PACK    #pd580_1(5,r13,240),_shadow4(9,r15,0)
0066    C050    0000    0035    LARL    r5,F'53'
006C    D204    D0F8    D0F0    MVC     #pd581_1(5,r13,248),#pd580_1(r13,240)
0072    41F0    F009            LA      r15,#AMNESIA(,r15,9)
0076    D100    D0FC    500C    MVN     #pd581_1(1,r13,252),+CONSTANT_AREA(r5,12)
007C    D204    D0E0    D0F8    MVC     _temp2(5,r13,224),#pd581_1(r13,248)
0082    F874    D100    2000    ZAP     #pd586_1(8,r13,256),_shadow3(5,r2,0)
0088    D207    D0E8    D100    MVC     _temp1(8,r13,232),#pd586_1(r13,256)
008E    5800    4000            L       r0,_shadow2(,r4,0)
0092    5850    4004            L       r5,_shadow2(,r4,4)
0096    EB00    0020    000D    SLLG    0,r0,32
009C    1605                    OR      r0,r5
009E    B3F3    0000            CDSTR   f0,r0
00A2    EB00    0020    000C    SRLG    r0,r0,32
00A8    B914    0011            LGFR    r1,r1
00AC    B3F6    0001            IEDTR   f0,f0,r1
00B0    6000    E000            STD     f0,_shadow1(,r14,0)
```

Annexe – Les compilateurs PL1 et C/C++ / Middlewares

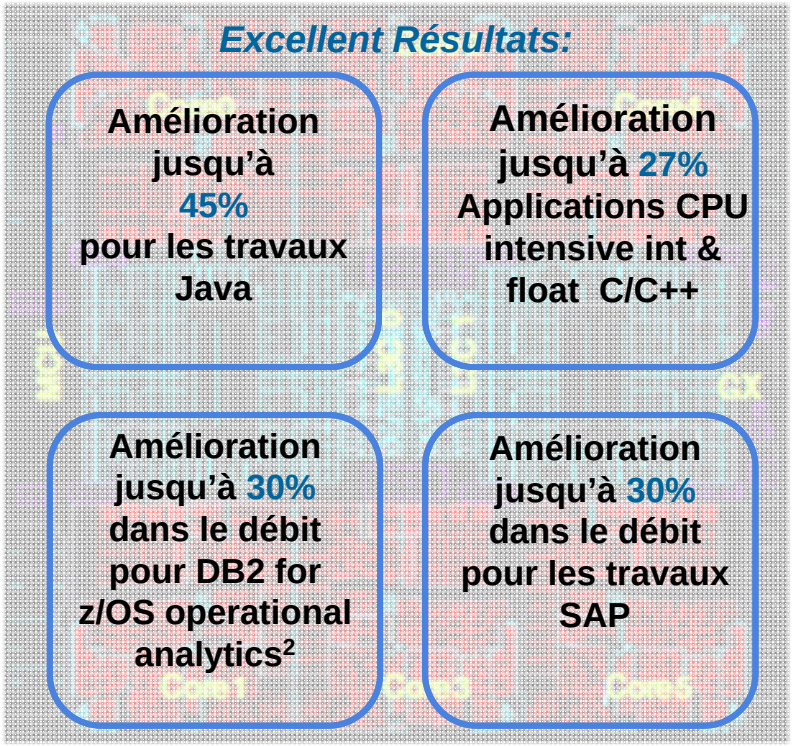
- Mettre à jour le compilateur au niveau PL/I for z/OS V4R3
  - Exemple - **ARCH(10)** 3/3 – Généré assembler (8 instructions) et s'exécute plus de 4 fois plus vite!

```
0060          EB2F        0003        00DF        SLLK        r2,r15,3
0066          B9FA        202F                    ALRK        r2,r15,r2
006A          A7FA        0001                    AHI         r15,H'1'
006E          B9FA        2023                    ALRK        r2,r3,r2
0072          ED08        2000        00AA        CDZT        f0,#AddressShadow(9,r2,0),b'0000'
0078          B914        0000                    LGFR        r0,r0
007C          B3F6        0000                    IEDTR       f0,f0,r0
0080          6001        E000                    STD         f0,_shadow1(r1,r14,0)
```

Annexe – Les compilateurs PL1 et C/C++ / Middlewares

- Synthèse: Chip Processeur optimisé pour les performances des logiciels
  - Exploité par Java, PL/I, compilateurs, DB2, et plus.
- Leadership dans la conception:
  - OoO
  - Caches
- Nouvelles fonctions matérielles
  - **Transactional Execution Facility** pour du parallélisme et l'évolutivité.
  - **Runtime Instrumentation Facility** pour réduire l'overhead Java.
  - **2 GB page frames** pour améliorer les performances des DB2 buffer et Java heaps.
  - **Jusqu'à 30% d'amélioration** du débit IMS grâce à un CPU, cache plus rapides et au compilateur

.



zEC12
Annexe – Nouvelles instructions

Annexe – 23 Nouvelles instructions

Name	Mnemonic	Opcode
COMPARE AND REPLACE DAT TABLE ENTRY	CRDTE	B98F
EXTRACT TRANSACTION NESTING DEPTH	ETND	B2EC
NONTRANSACTIONAL STORE	NTSTG	E325
TRANSACTION ABORT	TABORT	B2FC
TRANSACTION BEGIN (Nonconstrained)	TBEGIN	E560
TRANSACTION BEGIN (Constrained)	TBEGINC	E561
TRANSACTION END	TEND	B2F8
CONVERT FROM ZONED	CDZT	EDAA
CONVERT FROM ZONED	CXZT	EDAB
CONVERT TO ZONED	CZDT	EDA8
CONVERT TO ZONED	CZXT	EDA9
BRANCH PREDICTION PRELOAD	BPP	C7
BRANCH PREDICTION PRELOAD RELATIVE LONG	BPRP	C5
NEXT INSTRUCTION ACCESS INTENT	NIAI	B2FA
LOAD AND TRAP	LAT	E39F
LOAD AND TRAP	LGAT	E385
LOAD HIGH AND TRAP	LFHAT	E3C8
LOAD LOGICAL AND TRAP	LLGFAT	E39D
LOAD LOGICAL THIRTY ONE BITS AND TRAP	LLGTAT	E39C
COMPARE LOGICAL AND TRAP	CLT	EB23
COMPARE LOGICAL AND TRAP	CLGT	EB2B
ROTATE THEN INSERT SELECTED BITS	RISBGN	EC59
PERFORM PROCESSOR ASSIST	PPA	B2E8

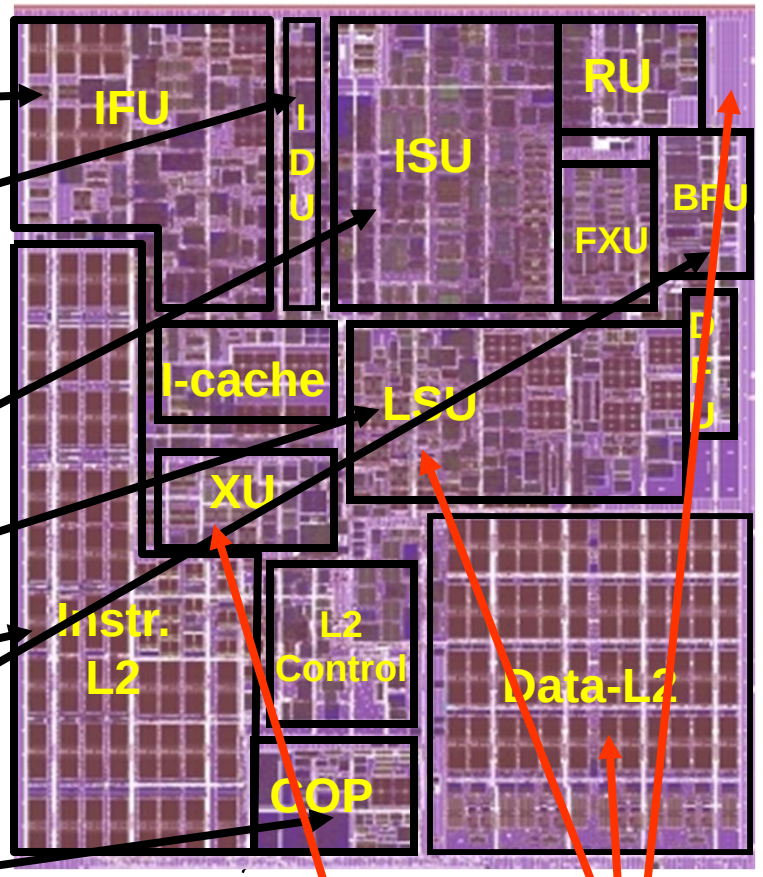
Explanation:

1. Also includes changed behavior for IDTE and IPTE
2. Also includes changed behavior for BCR, PFD, PFDRL, and STCMH

zEC12
Annexe - Détail du Processing Unit

zEC12 PU Details

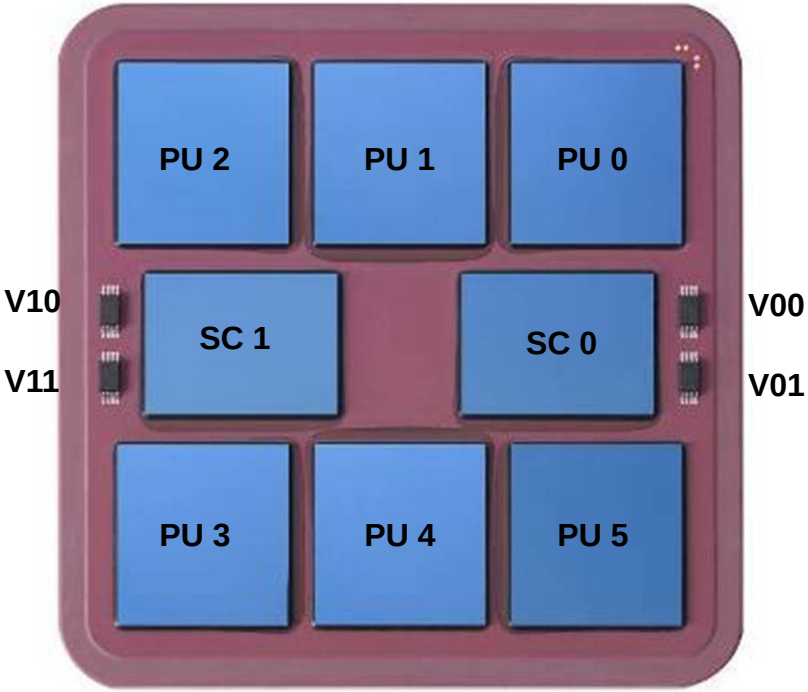
- Improved Instruction Fetching Unit
  - faster branch prediction
  - 2nd-level branch prediction
  - improved sequential instruction stream delivery
- Improved Out-of-order efficiency
  - better group formation, including regrouping
  - “uncracked” common instructions
  - on-the-fly Culprit/Victim detection for store-load hazards
- Increased Execution/Completion Throughput
  - bigger GCT, speculative completion
  - virtual branch unit
- Innovative Local Data-Cache design
  - store banking in Data-L1
  - unique Data-L2 cache (1M-byte) design
- Dedicated Instruction-L2 cache (1M-byte)
- Optimized Floating-Point Performance
  - Increased physical pool for FPRs
  - fixed-point divide in DFU
- Dedicated Co-Processor
  - Designed for improved start-up latency
  - UTF8<->UTF16 conversion support



- Main Architectural Extension Support
  - Transactional Memory support
  - Run-time instrumentation support
  - EDAT-2 support

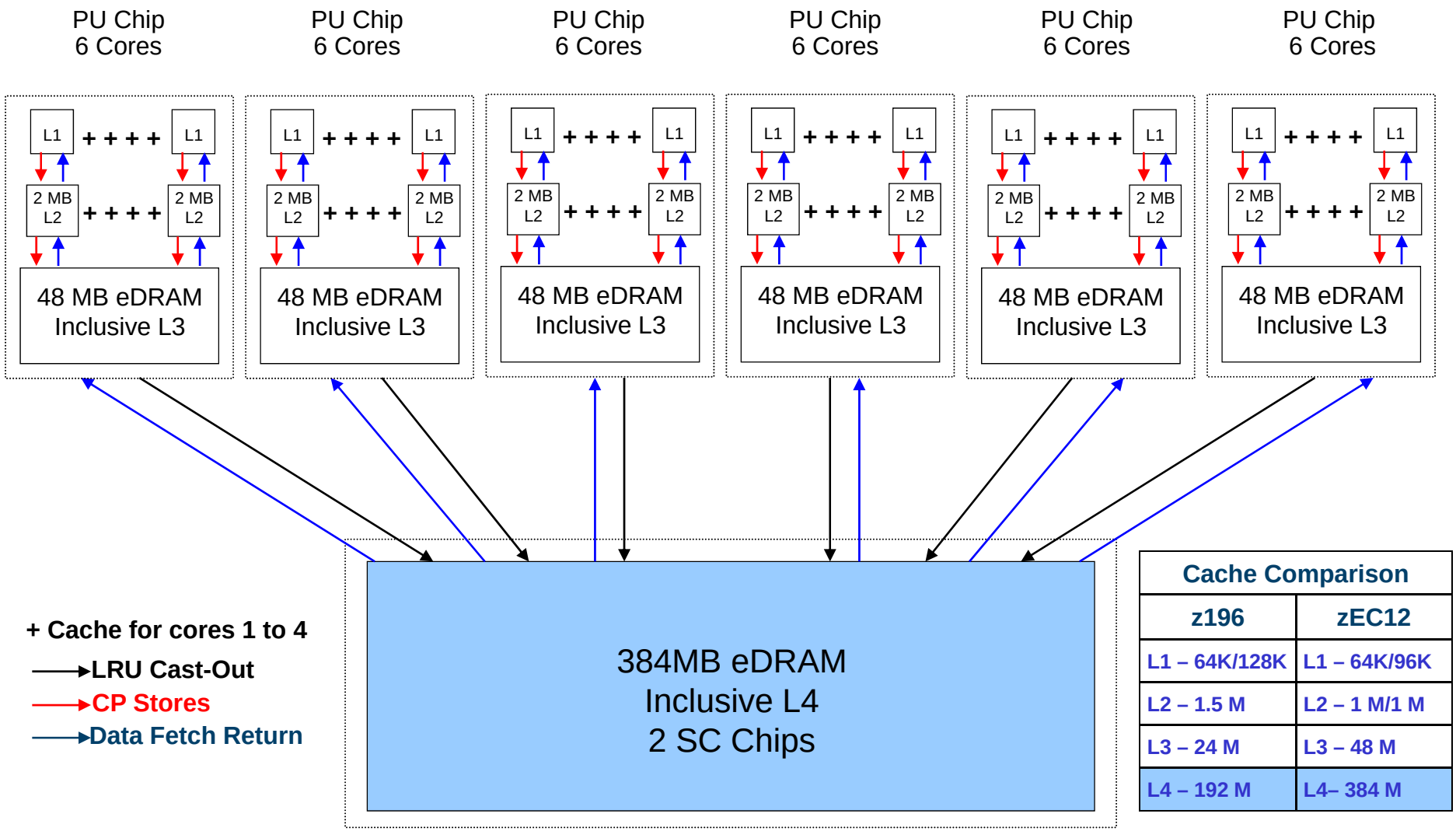
zEC12 Multi-Chip Module (MCM) Packaging

- 96mm x 96mm MCM
  - 102 Glass Ceramic layers
  - 8 chip sites
- 7356 LGA connections
  - 27 and 30 way MCMs
  - Maximum power used by MCM is 1800W



- CMOS 13s chip Technology
  - PU, SC, S chips, 32nm
  - 6 PU chips/MCM – Each up to 6 active cores
    - 23.7 mm x 25.2 mm
    - 2.75 billion transistors/PU chip
    - L1 cache/PU core
      - 64 KB I-cache
      - 96 KB D-cache
    - L2 cache/PU core
      - 1 MB I-cache
      - 1 MB D-cache
    - L3 cache shared by 6 PUs per chip
      - 48 MB
    - 5.5 GHz
  - 2 Storage Control (SC) chip
    - 26.72 mm x 19.67 mm
    - 3.3 billion transistors/SC chip
    - L4 Cache 192 MB per SC chip (384 MB/Book)
    - L4 access to/from other MCMs
  - 4 SEEPROM (S) chips – 1024k each
    - 2 x active and 2 x redundant
    - Product data for MCM, chips and other engineering information
  - Clock Functions – distributed across PU and SC chips
    - Master Time-of-Day (TOD) function is on the SC

zEC12 Book Level Cache Hierarchy



Cache Comparison	
z196	zEC12
L1 – 64K/128K	L1 – 64K/96K
L2 – 1.5 M	L2 – 1 M/1 M
L3 – 24 M	L3 – 48 M
L4 – 192 M	L4– 384 M

zEC12 Functional Comparison to z196

Processor / Memory	▪ Uniprocessor Performance	▪ Up to 25% performance improvement over z196 uniprocessor [1]
	▪ System Capacity	▪ Up to 50% system capacity performance improvement over z196 80-way [1]
	▪ Processor Design	▪ New 5.5 GHz processor chip versus 5.2 GHz
	▪ Cache	▪ zEC12 has 33% more L2 cache, instruction and data (total 2 MB versus total 1.5 MB on z196), 100% more L3 cache (total 48 MB versus 24 MB on z196), 100% more L4 cache (384 MB versus 196 on z196)
	▪ Models	▪ Five models with up to 4 books (z196 had five models)
	▪ Processing cores	▪ Up to 101 cores to configure, up to 80 on z196
	▪ Granular Capacity	▪ Up to 161 capacity settings versus 125 on the z196
	▪ Memory	▪ Up to 3 TB RAIM memory (same as z196)
	▪ Fixed HSA	▪ Up to 32 GB fixed HSA versus 16 GB fixed on z196
Virtualization and Alternative Processors	▪ Virtualization	▪ zEnterprise Unified Resource Manager provides virtualization management for blades installed in the zBX Mod 003. ▪ zEnterprise Unified Resource Manager has “resource workload awareness” where hybrid resources can be managed and optimized across the zEnterprise.
	▪ zEnterprise BladeCenter Extension (zBX)	▪ zEnterprise System is a truly integrated hardware platform that is able to span and intelligently manage resources across mainframe and distributed technologies – including select POWER7 and IBM System x blades ▪ Supported optimizer is IBM WebSphere DataPower XI50 in the zBX Mod 003. ▪ zBX Model 003 (versus zBX Model 002 which attaches to z196)
Connectivity	▪ HiperSockets™	▪ Both zEC12 and z196 support of 32 HiperSockets
	▪ FICON	▪ New OSA-Express4S 1000 BASE-T included in PCIe I/O infrastructure. FICON Express8S and OSA-Express4S adapters available on zEC12, z196, z114
	▪ I/O subsystem	▪ zEC12 has industry standard 8 GBps InfiniBand supports high speed connectivity and high bandwidth
	▪ Internal I/O Bandwidth	▪ Coupling with HCA-3 InfiniBand Coupling Links
	▪ Coupling	▪ Crypto Express4S enhanced with new FIPS 140-2 Level 4 cert and PKCS#11 support
	▪ Cryptography	▪ Elliptic Curve Cryptography (ECC)
RAS	▪ RAS Focus	▪ New IBM zAware offers high speed analytics facilitates the ability to consume large quantities of message logs for smarter monitoring
	▪ Availability	▪ zEC12 offers advanced memory enhancements (RAIM) and advanced power and thermal optimization and management that can help to control heat / improve RAS ▪ New PCIe Flash Express on zEC12 to handle paging workload spikes and improve availability – not available on z196
Environmentals	▪ Energy	▪ Power Save modes for processor
	▪ Cooling	▪ New improved integrated cooling system ▪ Optional Non Raised Floor and overhead cabling options for both I/O and (New!) Power ▪ Optional water cooling and DC power

zEC12 : Des Mips oui... mais aussi beaucoup d'autres dispositifs de performance

Alain Maneville

Senior Certified I/T Specialist – zChampion

FIN DU DOCUMENT

Université du Mainframe 2013

4-5 avril