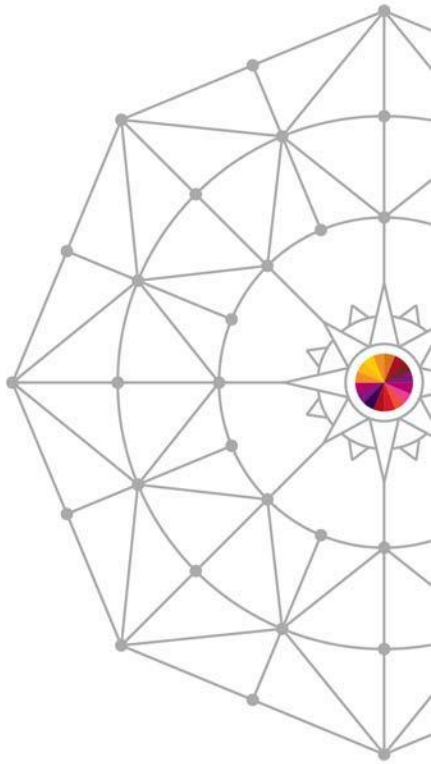
Session 15602:
The Skinny on Coupling Thin Interrupts

Frank Kyne

Watson & Walker

frank@watsonwalker.com

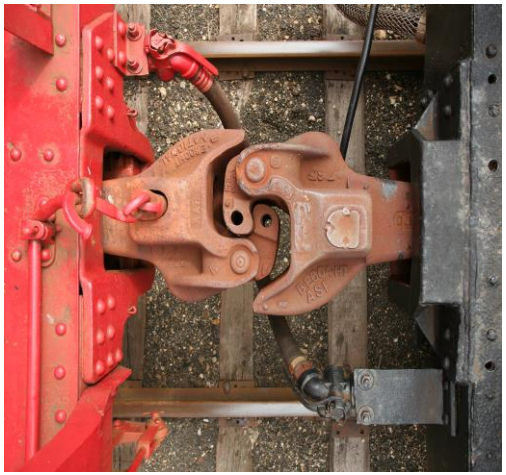


Insert Custom Session QR if Desired.

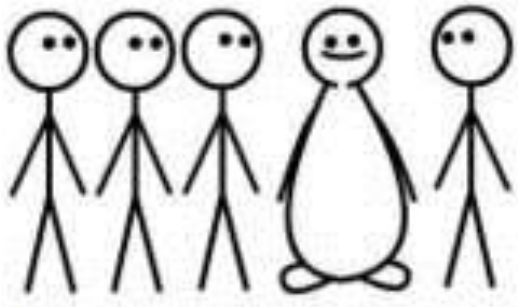
#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Why we are here



+



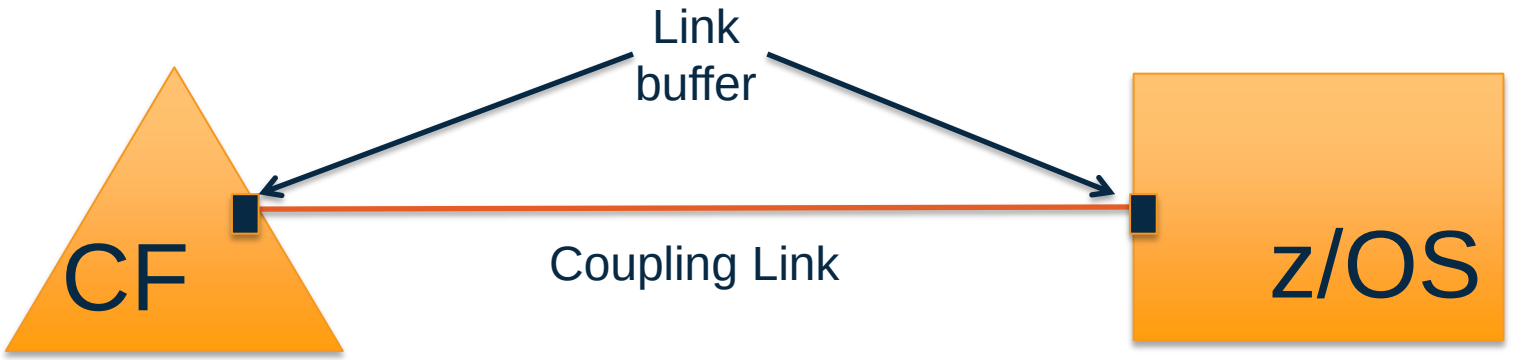
+

Welcome

- Hi, thanks for coming
- Who I am
- PLEASE ask questions as I go along
- Note to DFSORT – I'm sorry for "putting the mockers" on you….

Coupling Thin Interrupts

- What exactly IS a Coupling Thin Interrupt?
  - Prior to Driver 15 (zBC12 and zEC12 GA2) the arrival of a signal on a Coupling Facility link did not generate an interrupt.
    - This meant that the users of Coupling Links (Coupling Facilities and z/OS systems) needed some other mechanism to detect the presence of something in the link buffer that needed to be attended to.



Coupling Thin Interrupts

- Driver 15 introduced the ability for the link hardware (under program control) to generate an interrupt when something arrives in the link buffer.
- Because the CF signals are simpler than other forms of I/O (DASD I/O, for example) the processing associated with them is less complex.
- Hence:
  - Coupling — Because they are for Coupling Links
  - Thin — Because they are “light weight”
  - Interrupts — Because they generate an interrupt.

Coupling Thin Interrupts

- Well, that's it. Now you know what Coupling Thin Interrupts are.

- Thanks for coming.

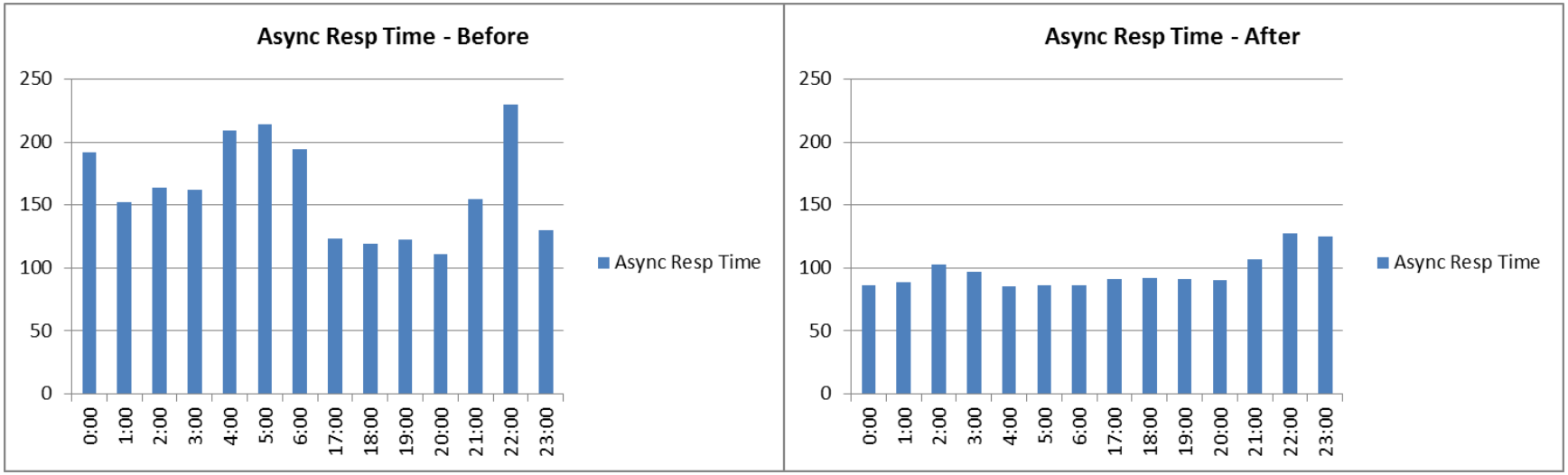
Coupling Thin Interrupts

- I suppose you want a little more?

Coupling Thin Interrupts

- This is what Coupling Thin Interrupts can do for YOU

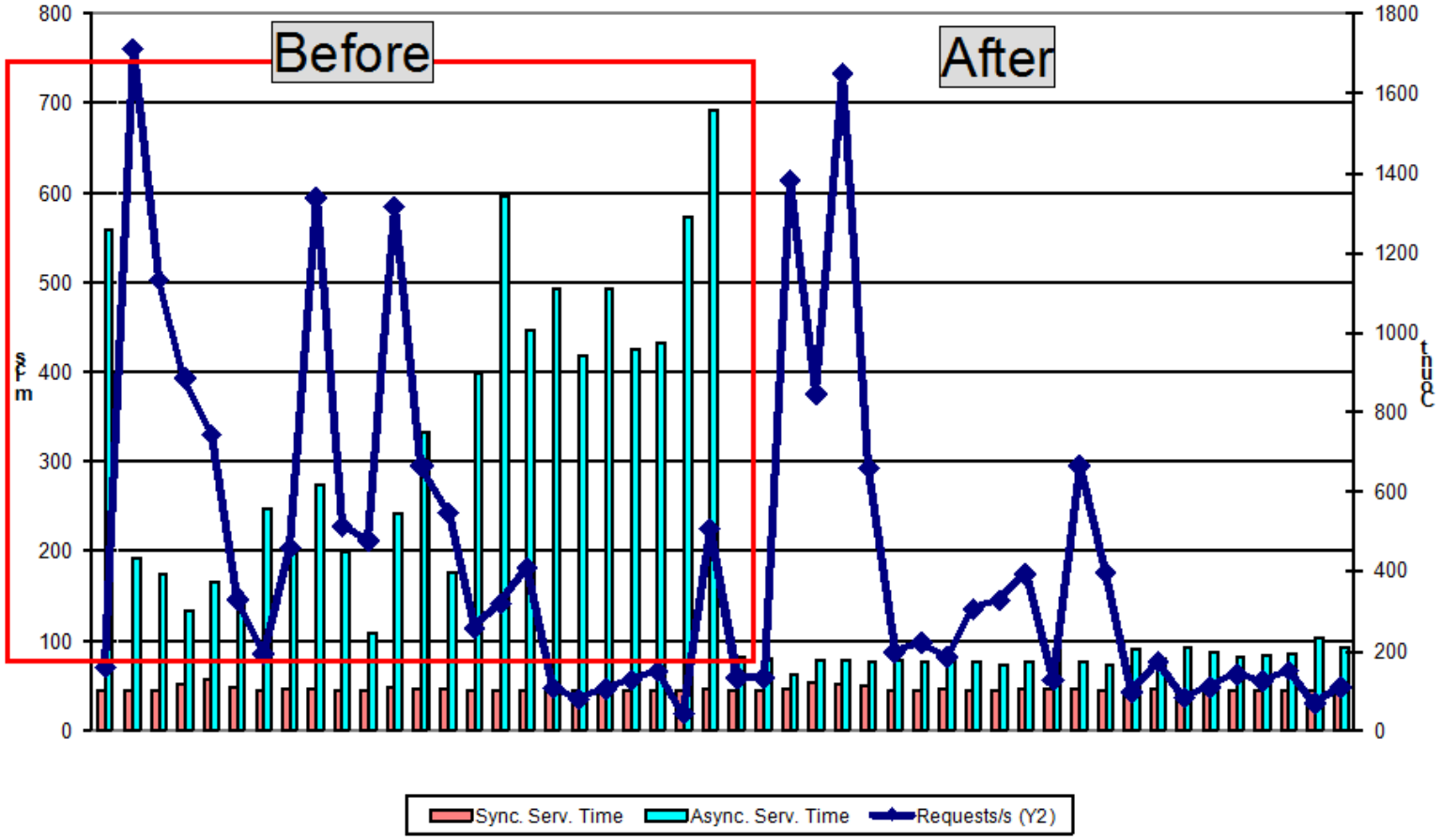
Production environment, Customer 1



Coupling Thin Interrupts

- Or maybe this:

Test environment, Customer 2



Coupling Thin Interrupts

- The Coupling Thin Interrupt capability can be exploited by both Coupling Facility and by z/OS, but in different ways.

- Let's look at z/OS first.

Coupling Thin Interrupts

- We need a little background information first….
- On z/OS, Coupling Thin Interrupts can change:
  - How PR/SM dispatching works for a z/OS LPAR:
    - On the z/OS end, Coupling Thin Interrupts can be used regardless of whether z/OS using shared or dedicated engines.
  - How XCF becomes aware that he has some work to do.

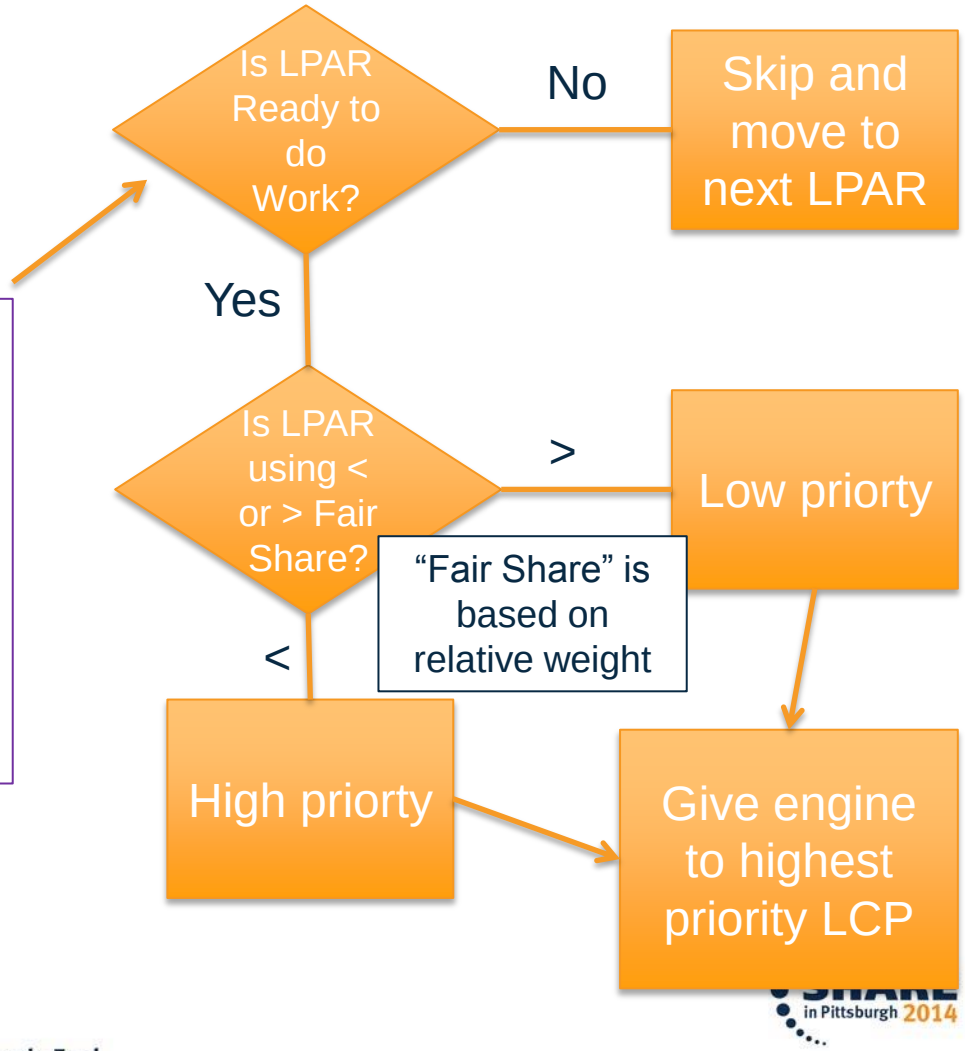
Coupling Thin Interrupts

PR/SM Dispatching

PR/SM Dispatching and CTI

The presence of an outstanding interrupt for an LPAR makes PR/SM aware that the LPAR is ready to do some work

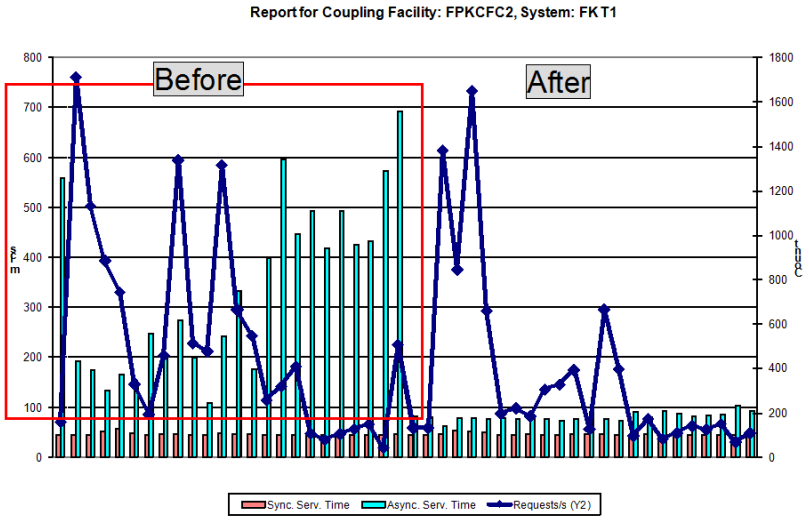
Without Coupling Thin Interrupts, an LPAR that has an outstanding CF signal might appear that it has no work to do, so it gets passed over



Coupling Thin Interrupts

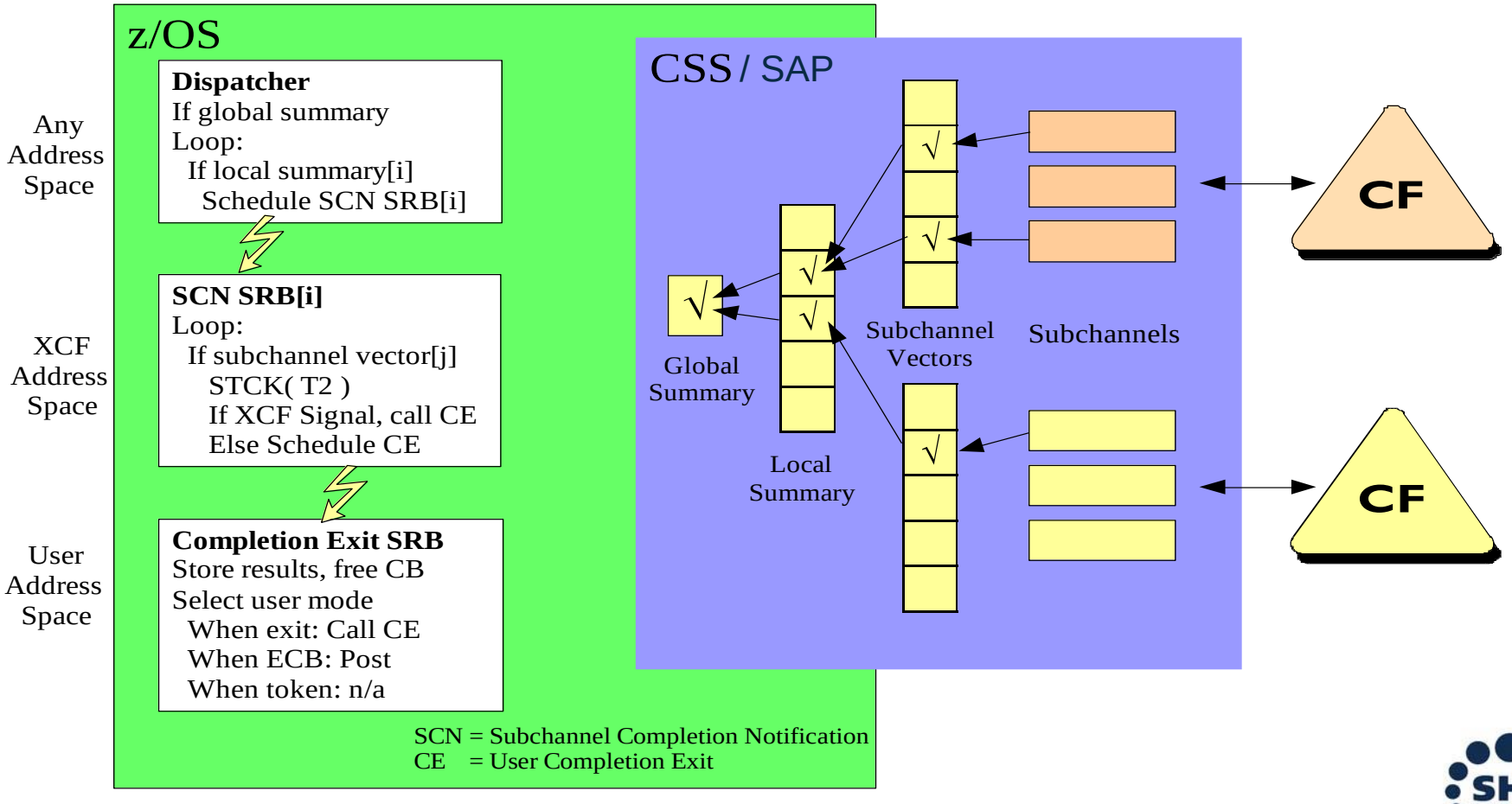
Benefit #1 (PR/SM/LPAR Level):

- If z/OS using a shared engine and either has a low weight or is not very busy, the existence of an outstanding interrupt can decrease the time that LPAR is waiting to get dispatched again.



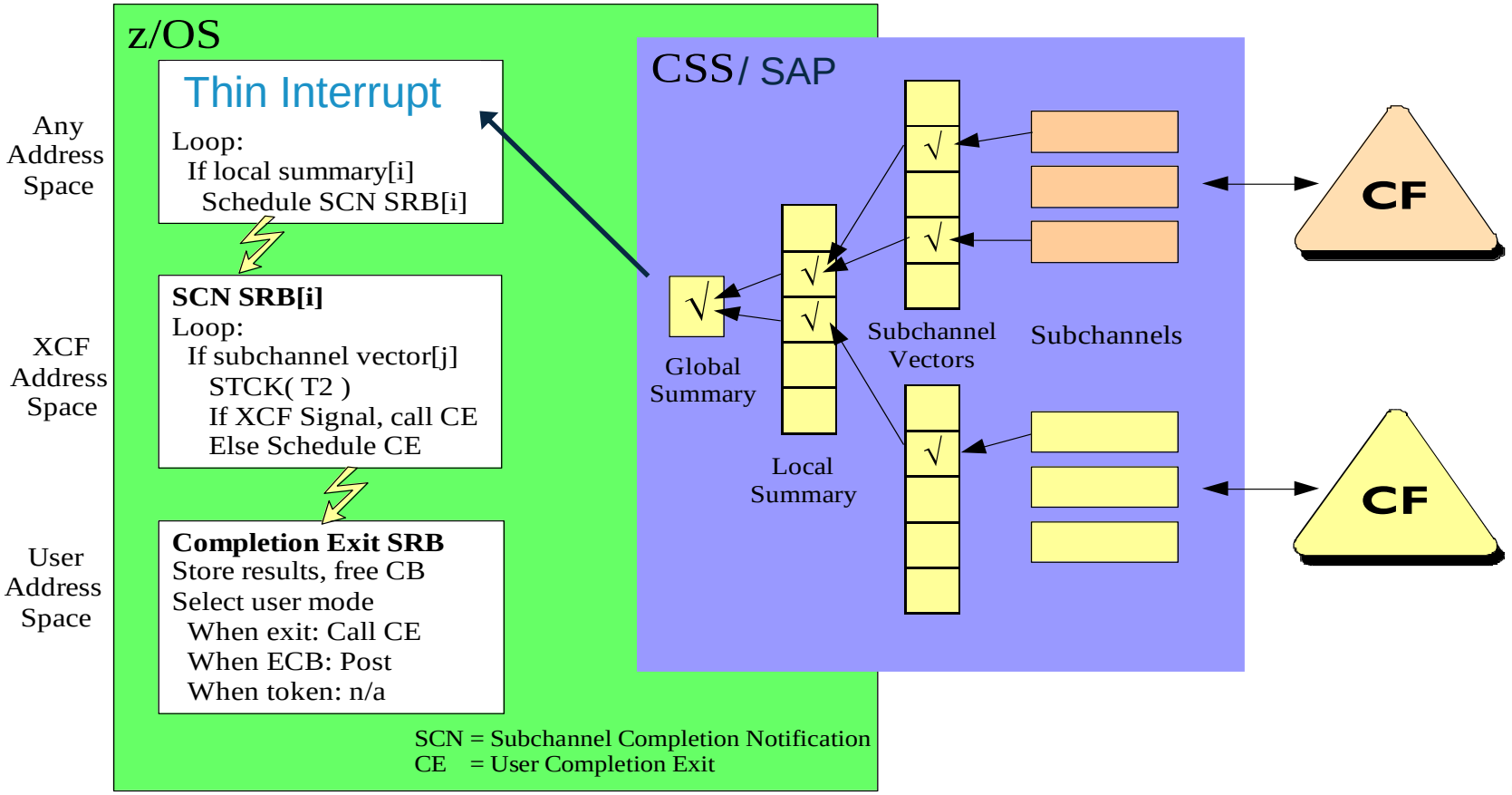
Coupling Thin Interrupts

Processing for asynch response or CF notification (**Before** CTI)



Coupling Thin Interrupts

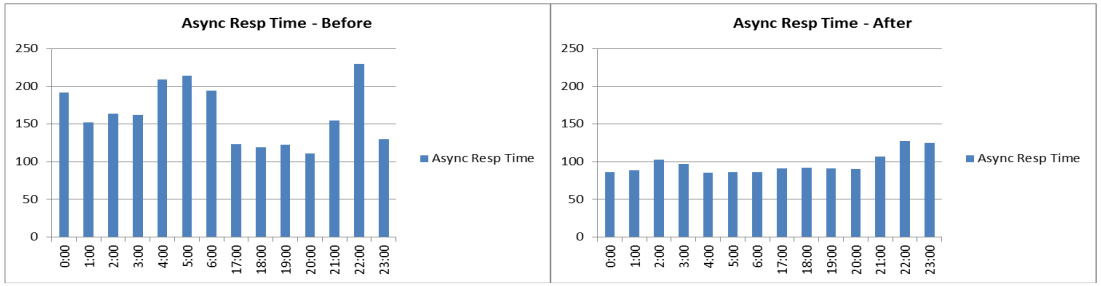
Processing for asynch response or CF notification (**After** CTI)



Coupling Thin Interrupts

Benefit #2 (z/OS/XCF Level):

- Because the arrival of a CF signal generates an interrupt, you no longer need to wait for the dispatcher to check the Global Summary bit.
  - Reduces interval between when signal arrives from CF and when XCF gets called to process it.
  - Delivers more consistent response times because interrupt will be processed more or less immediately rather than having to wait a variable amount of time for the dispatcher to get control and check Global Summary bit



Coupling Thin Interrupts

- Prerequisites – for z/OS exploitation of Coupling Thin Interrupts:
  - z/OS must be running on zEC12 GA2 or zBC12 GA1
    - LPAR can be using shared or dedicated engines
  - z/OS V2.1
  - z/OS V1.13 with APARs OA38734, OA37186, OA38781
  - z/OS V1.12 with APARs OA38734, OA37186, OA38781
  - COUPLINGTHININT XCF FUNCTION *is enabled by default* and can be turned on and off dynamically using the SETXCF FUNCTIONS command.

Coupling Thin Interrupts

- Summary for the z/OS end of CTI:
  - As long as you are on a CPC with Driver 15 or later, and running z/OS 1.12 or later with the required fixes, z/OS will AUTOMATICALLY use Coupling Thin Interrupts
  - LPARs most likely to benefit are:
    - Those where asynch response times are a LOT higher than synchronous response times.
    - LPARs with shared engines and low weights.
    - LPARs with large variances in asynch response time for different times of day.

Coupling Thin Interrupts

- We said that both z/OS and CFs can exploit Coupling Thin Interrupts, but also that the considerations are different for the two.

- Why?

Coupling Thin Interrupts

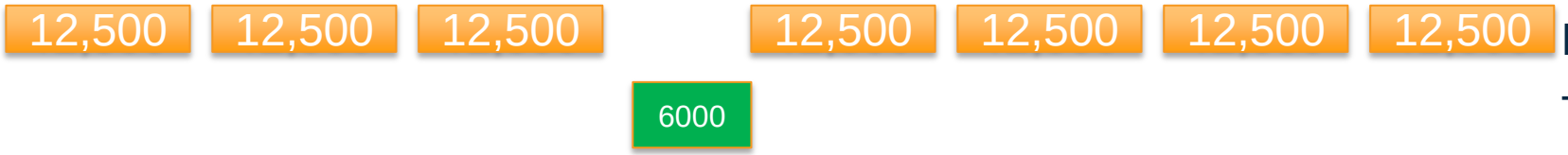
- 1) z/OS systems have millions of things to do – processing CF signals is just one of them – hence the old model where the MVS Dispatcher would only check the Global Summary bit every so often. CF's on the other hand ONLY process CF requests. To do that as quickly as possible, Coupling Facility Control Code spends its time either processing a request or looking in the link buffers, for some work to do.
- If the CF LPAR is always dispatched (that is, it has a dedicated engine), it is always looking in the link buffers anyway, so interrupts would not provide any response time benefit in that situation.
- As a result, Coupling Thin Interrupts can only be used for CFs with shared engines.

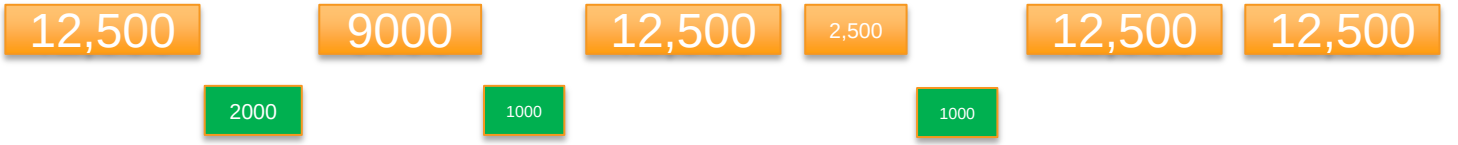
Coupling Thin Interrupts

- 2) CFs role in life is to deliver the best response time it can – let's say 5 mics.
- CF knows that when it lost the engine, it would be waiting THOUSANDS of mics before it was dispatched again – meaning that requests that arrived during that time would have to wait a LONG time until the CF LPAR is dispatched again. In an attempt to avoid this delay, CFCC tries to hold on to the engine as long as possible:
  - For a CF LPAR running with DYNDISP OFF, it will never release the engine until PR/SM takes it away.
  - For a CF LPAR running with DYNDISP ON, it will finish all its work, then hang on for a while longer (hoping that some more work shows up) then finally go to sleep and release the engine back to PR/SM.
- This behavior was based on the fact that there was no interrupt mechanism for CF signals.

Coupling Thin Interrupts

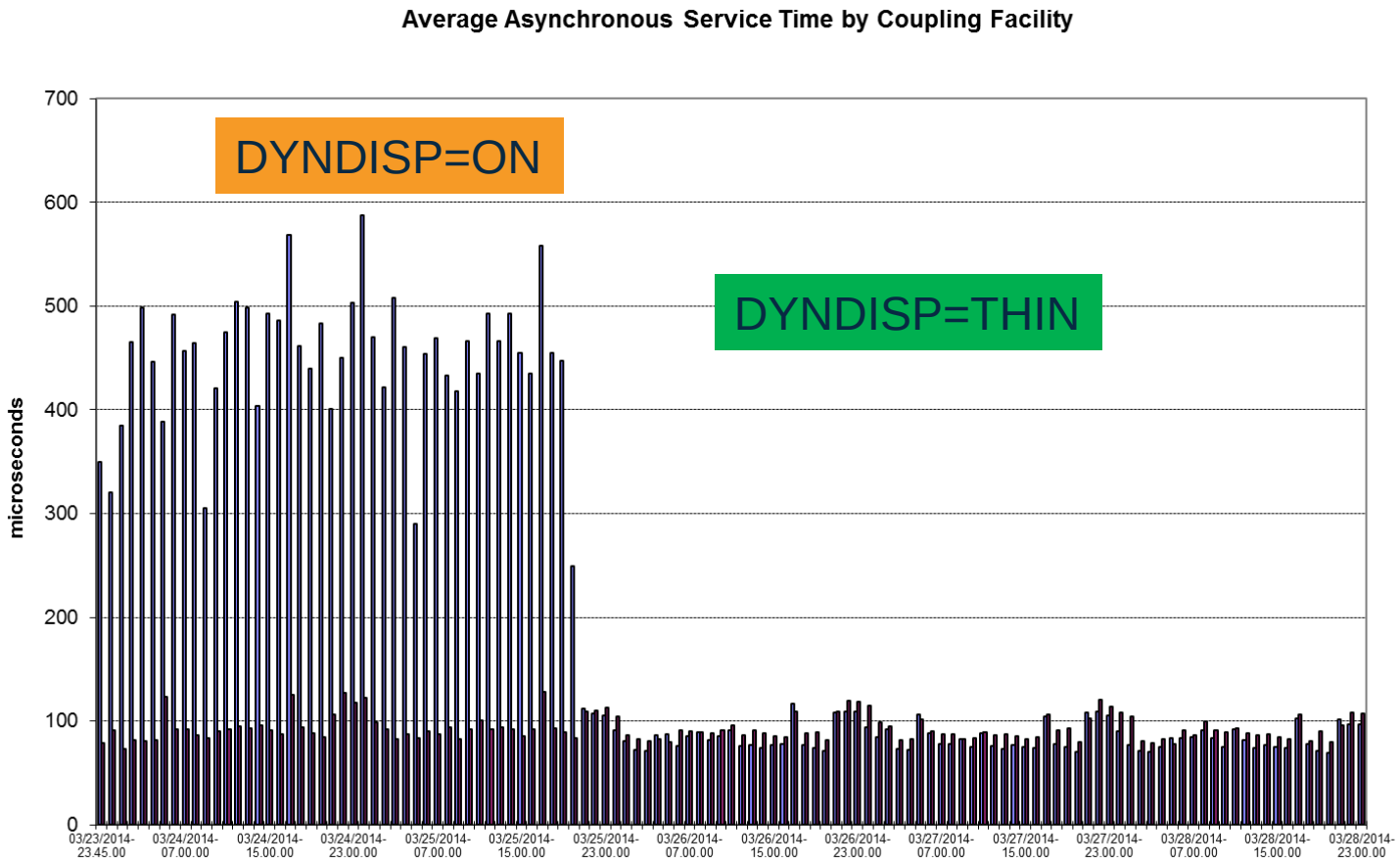
- So how does Coupling Thin Interrupts change things?
- 1) Because there is an interrupt mechanism, requests to a CF that is not currently dispatched will be picked up much sooner.





Coupling Thin Interrupts

- How this impacts response times for the THIN CF

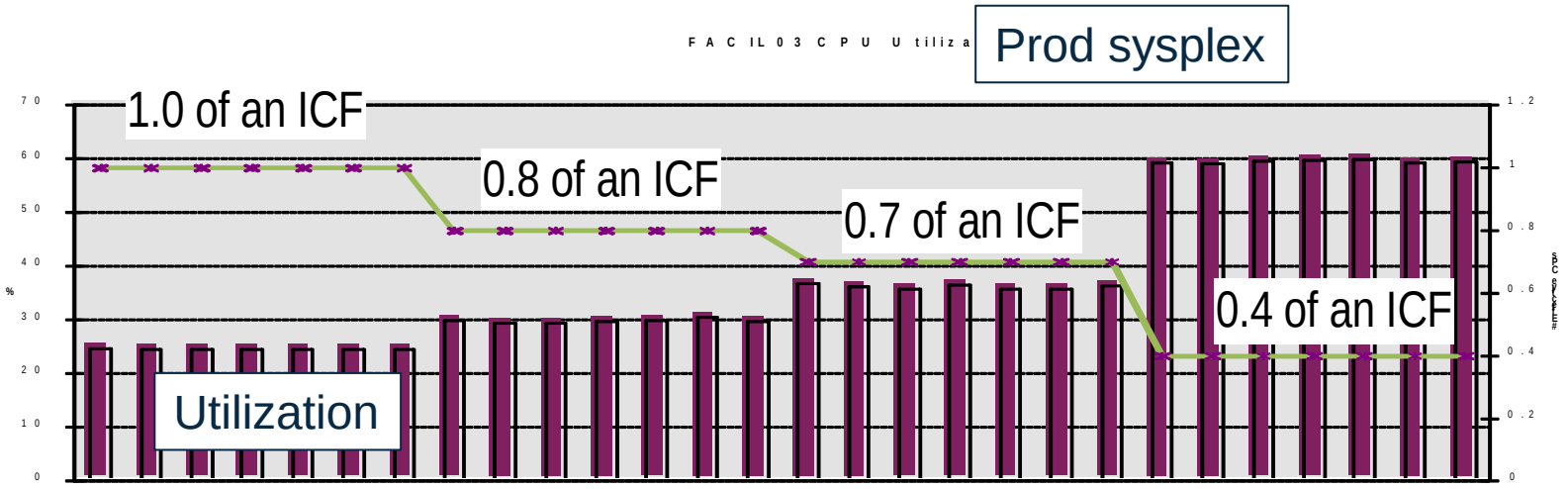


Coupling Thin Interrupts

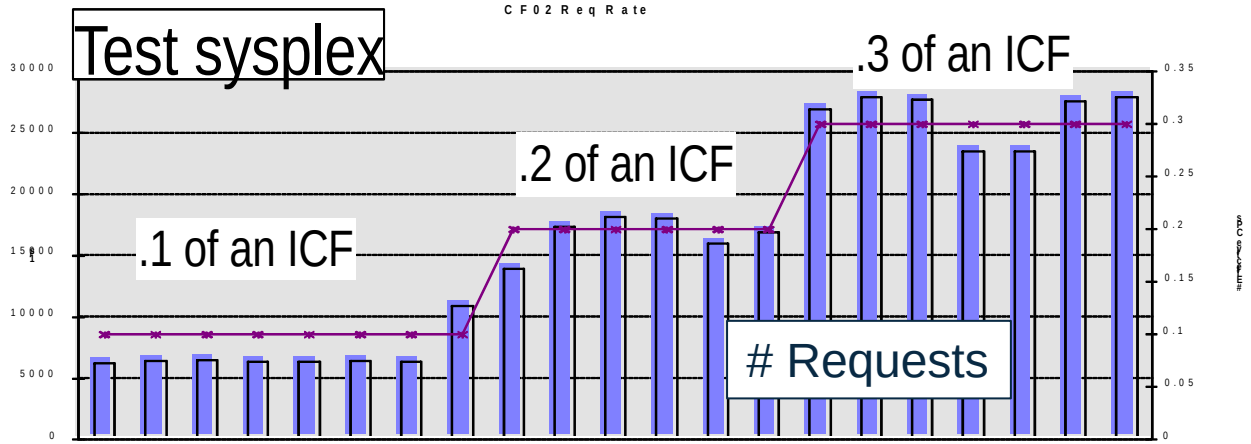
What else? Does it make the coffee?

Because CFCC knows that it will be able to react to newly arriving requests much sooner when DYNDISP is set to THIN, it is willing to give up its engine much sooner. This frees up capacity that can be used by its neighbors.

Coupling Thin Interrupts



Total number of requests



Coupling Thin Interrupts

And that is a very important point about the impact of Coupling Thin Interrupts on CFs – it doesn't only impact the CF LPAR where it is enabled, it ALSO impacts any other CF LPARs that it is sharing engines with.

Coupling Thin Interrupts

- Prerequisites – for CF exploitation of Coupling Thin Interrupts:
  - CF must be running on CPC with Driver 15 or later.
  - CF LPAR must be using shared engines.
  - Coupling Thin Interrupts must be explicitly turned on for that LPAR using the DYNDISP THIN command.

  - A Swiss bank account to hold all the awards and bonuses you will get after implementing this.

Coupling Thin Interrupts

- Coupling Thin Interrupts work with any type of CF link – ISC, ICP, PSIFB 1X/12X/IFB Mode/IFB3 Mode.
- Whatever type of CEC is at the other end of the CF link is irrelevent.
- It is irrelevant whether Couping Thin Interrupts are turned on on the z/OS LPAR.

Coupling Thin Interrupts

- Recommendations:
  - For CFs that are using DYNDISP ON today, we recommend that you switch to DYNDISP THIN.
  - If your CFs are using DYNDISP OFF today, we recommend that you at least try DYNDISP THIN (you can switch back and forth non-dispuptively).
  - If your production CF has a dedicated engine AND runs at extremely low utilizations (peak <10%), you MIGHT consider testing it with a shared engine and DYNDISP THIN before the next time you upgrade the CF CPC.
    - You might find that you can get acceptable response times without needing a dedicated engine.
    - But you can only do this if at least one of your CFs is already in a zEC12/zBC12.
    - And changing your CF engine from dedicated to shared and back requires an LPAR deactivate/reactivate so is somewhat disruptive.

Coupling Thin Interrupts

- References:
  - Setting Up a Sysplex
  - PR/SM Planning Guide (for EC12 or later)
  - Excellent White Paper 102400 – '*Coupling Thin Interrupts and Coupling Facility Performance in Shared Processor Environments*' by Barbara Weiler

Coupling Thin Interrupts

Thank you for coming!

If you liked this session, please check out others from Cheryl and Frank:

- 15836 – Tue, 3 pm – Cheryl & Frank – *Why Is the CPU Time for a Job so Variable?*
- 16251 – Thu, 3 pm – Cheryl & Frank – *Hot Tips From Cheryl and Frank*
- 15567 – Fri, 10 am – Cheryl & Frank – *Exploiting z/OS Tales From the MVS Survey*

If you like SMF data, please see our new *SMF Reference Summary* at www.watsonwalker.com/references.html