

# *Let's Stay Close*



**Ruud van der Pas**

**Distinguished Engineer**

**SPARC Microelectronics**

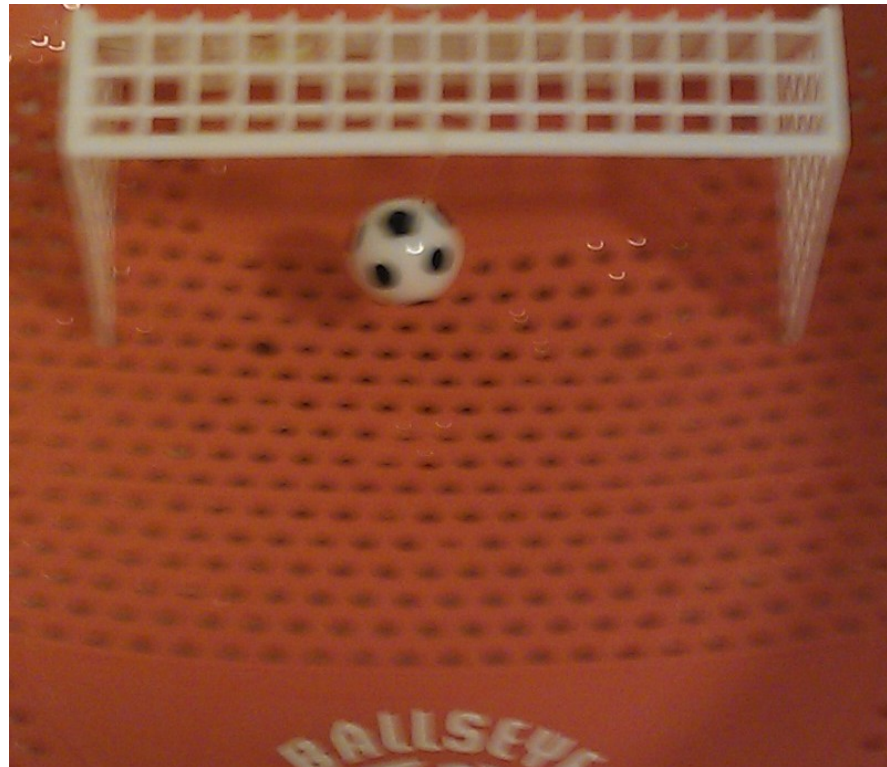
**ORACLE**

**Santa Clara, CA, USA**

*SC'14 Talk at OpenMP Booth*

*Wednesday, November 19, 2014*

***We Will Look At The Big Picture  
(which can be an eye opener)***



*Put Things In The Right Perspective*



# *Summary*

## *(Dedicated To The Impatient)*





*Place : where threads can execute*



## *Hamburger Place*

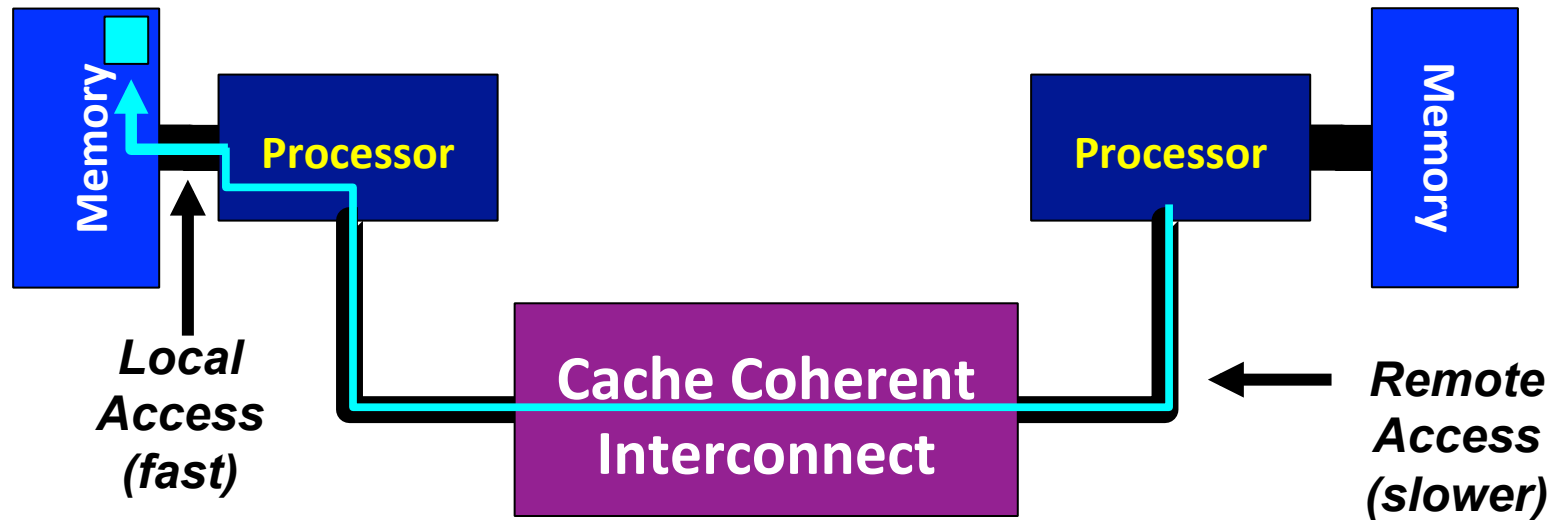


*Place : where threads can execute*  
*Affinity Policy: where threads go*

# *Why Worry ?*



# A Generic cc-NUMA Architecture



**Main Issue:**  
**How To Distribute The Data ?**

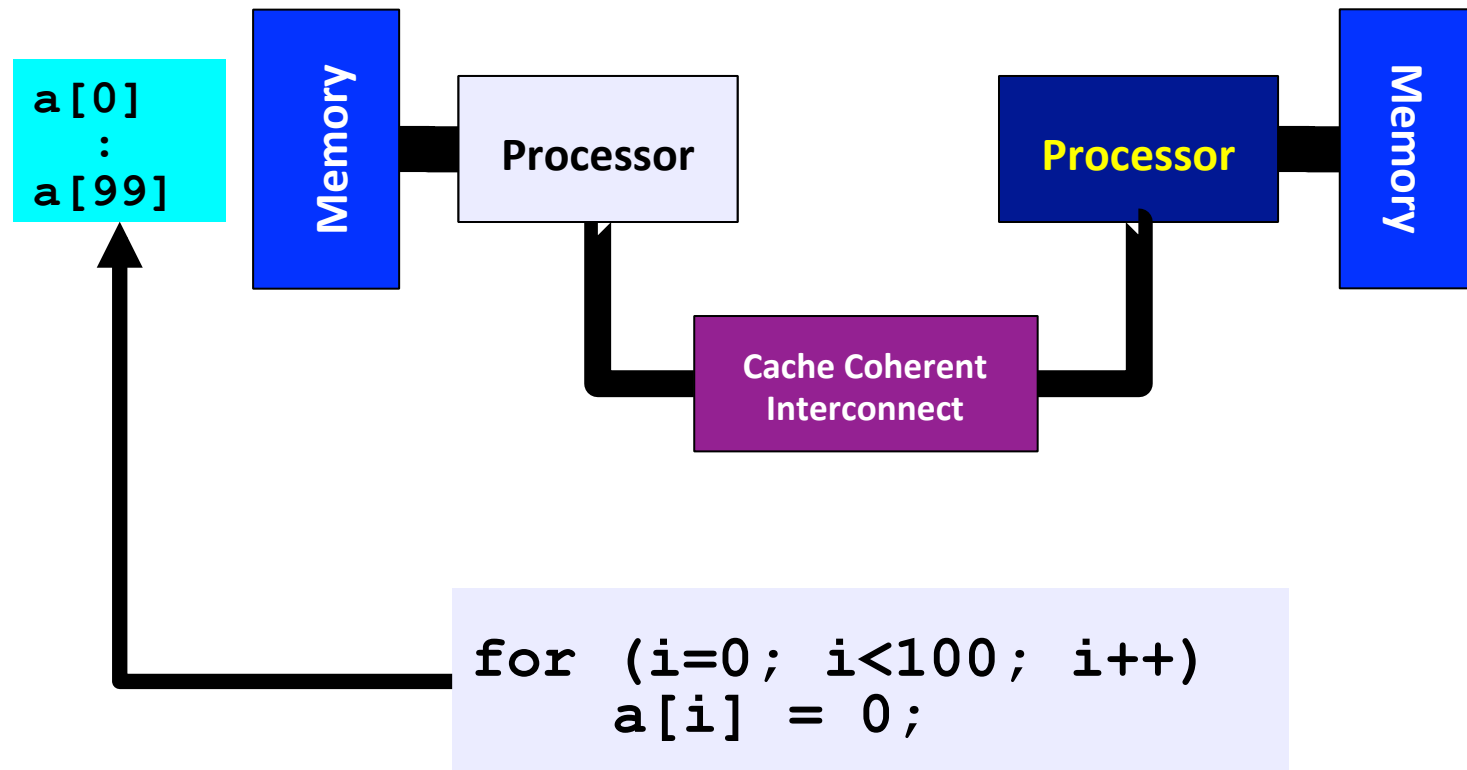
# Where Does Your Data Go ?



*The OS decides on the placement of data*

*A common default is to use the “First Touch” policy*

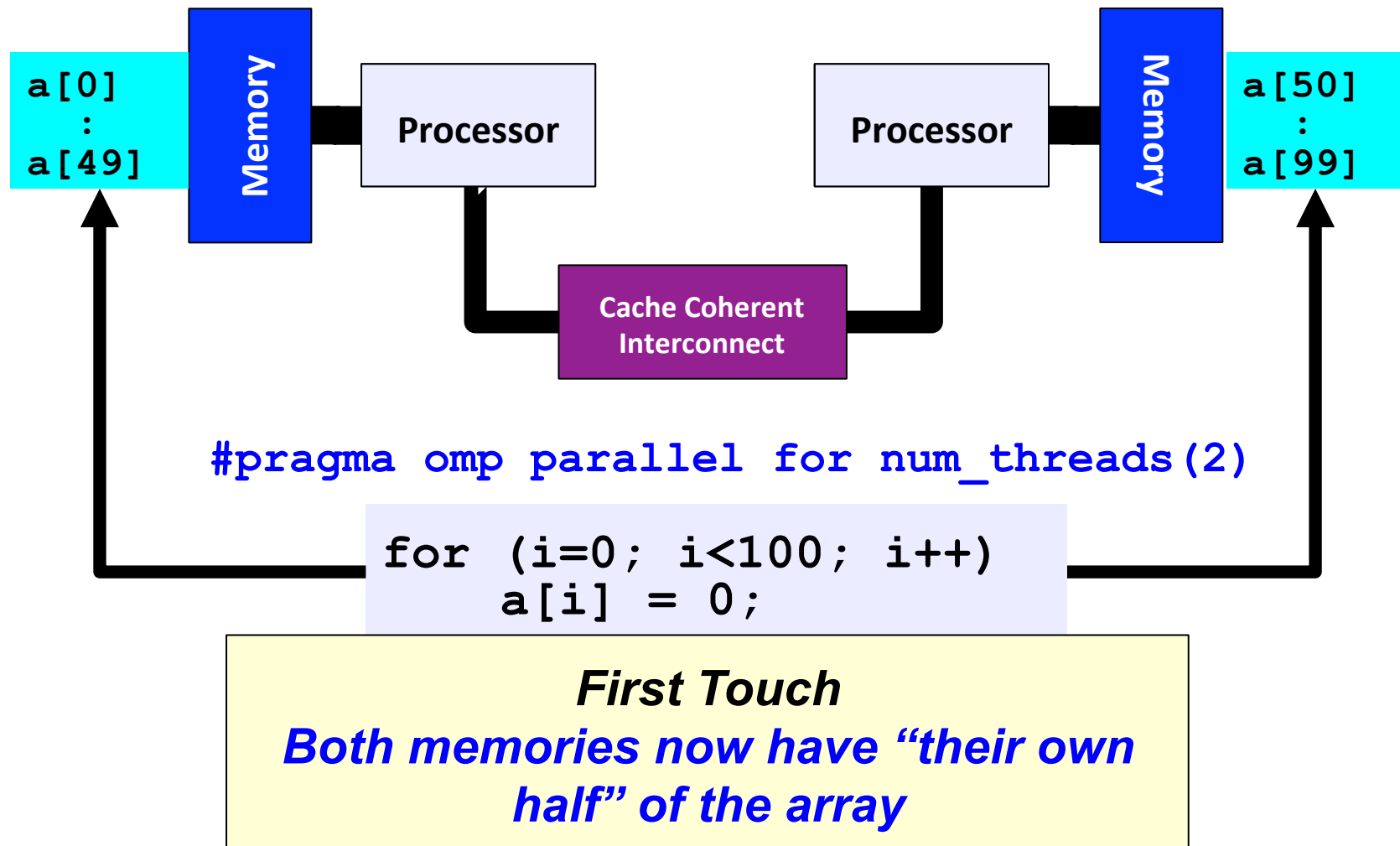
# Example First Touch Placement/1



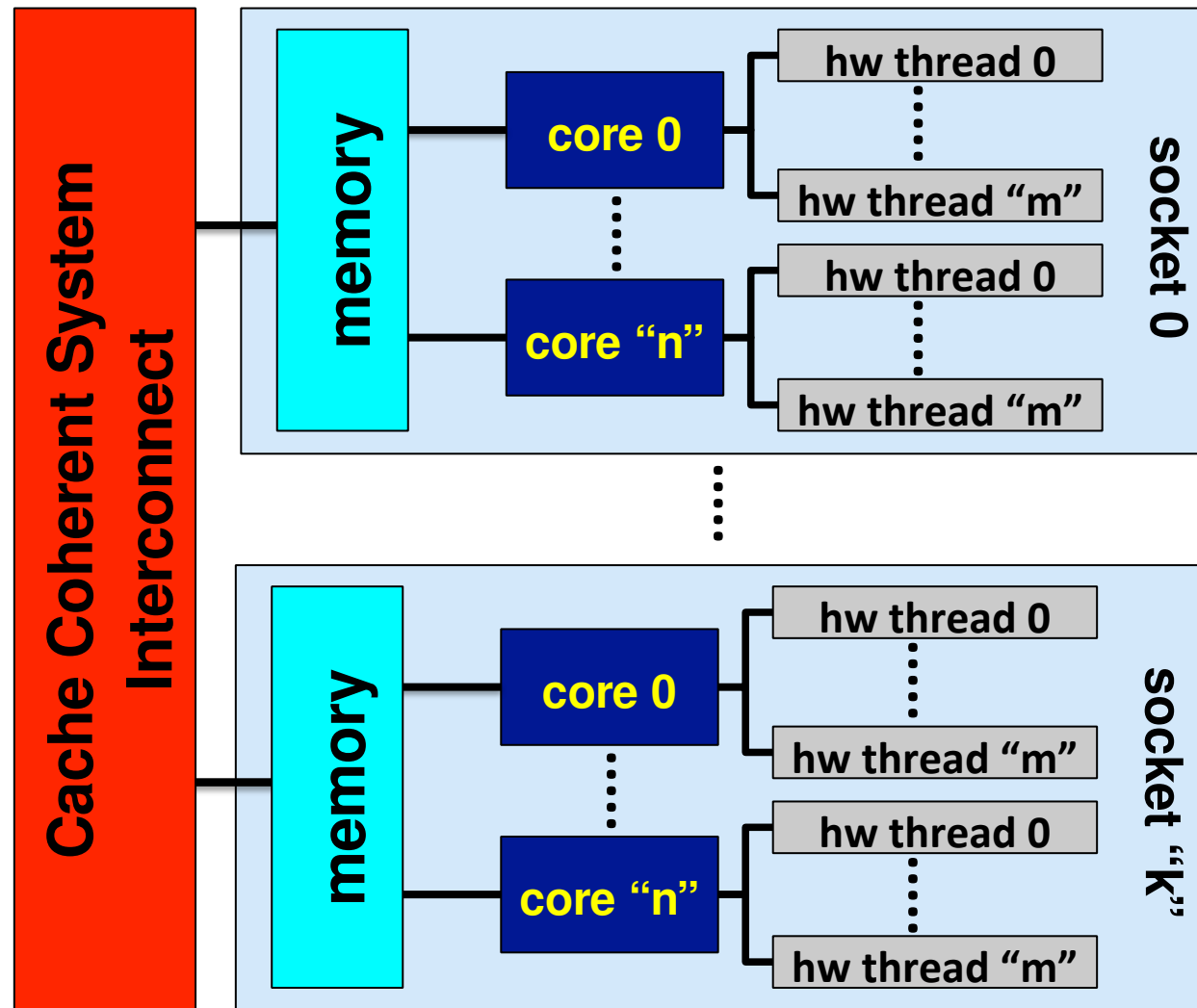
***First Touch***  
***All array elements are in the memory of the processor executing this thread***



# Example First Touch Placement/2



# Thread Affinity – Machine Model



# Why Thread Affinity Support ?



***For good performance and scalability it is key that***

***Threads are close to the data they need most often***

## **Benefits**

***Reduces remote memory references***

***Maximizes bandwidth and reduces latency***



# Basic OpenMP Philosophy



*Data is wherever it may be*

*Threads are moved to the data they need most*

# Two Key Concepts



***The Place List***

***The Thread Affinity Policy***

# *The Place List*



# Places – Definition and Notation



*A place consists of a (set of) numbers*

*Each number represents a scheduling unit*

*That is, something a thread can run on*

*For example, a hardware thread*

*Single element place*



*Multi element place*



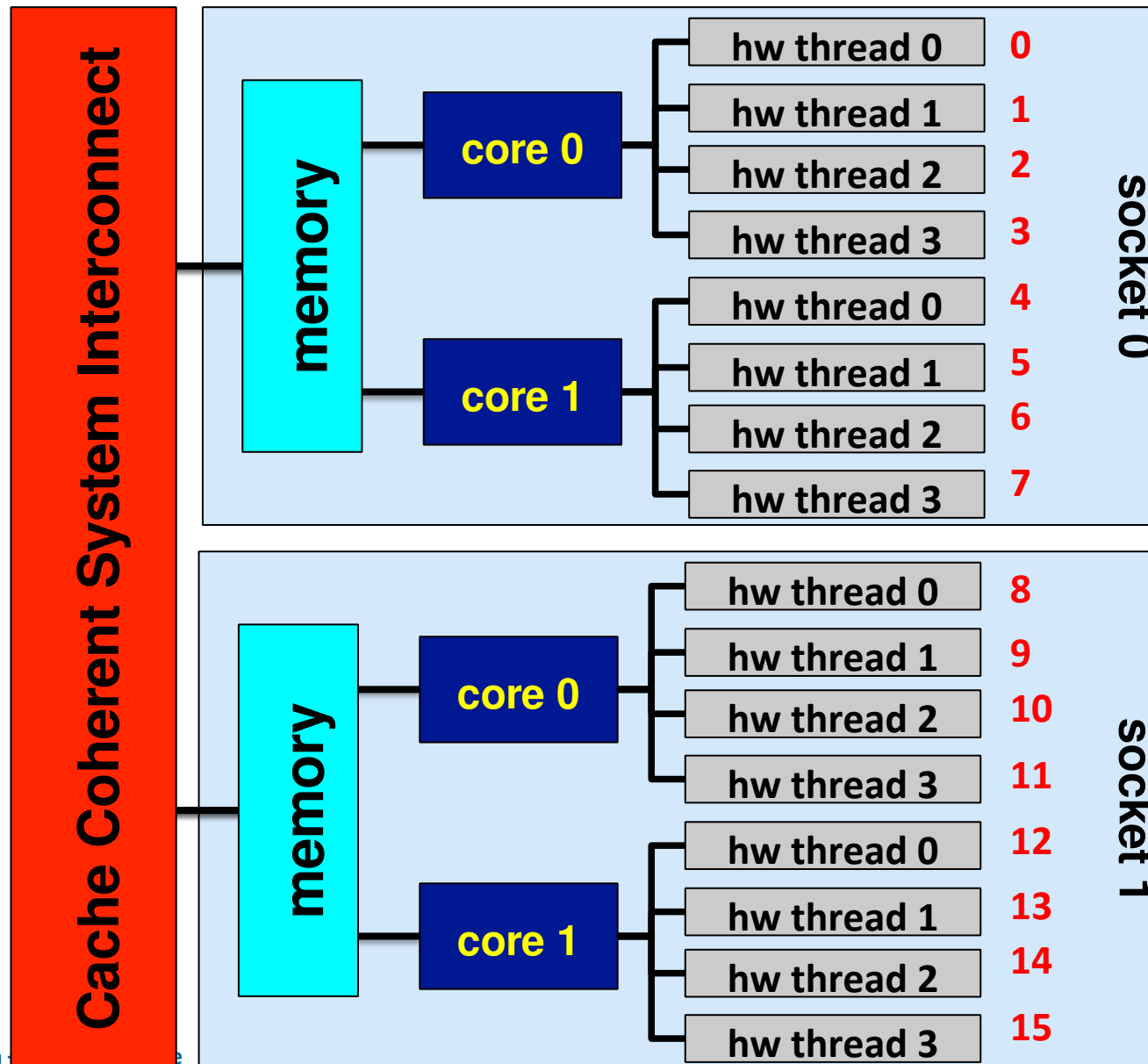
# Example Multicore System



Component	Count	Total
Sockets	2	2
Cores/socket	2	4
Threads/core	4	16

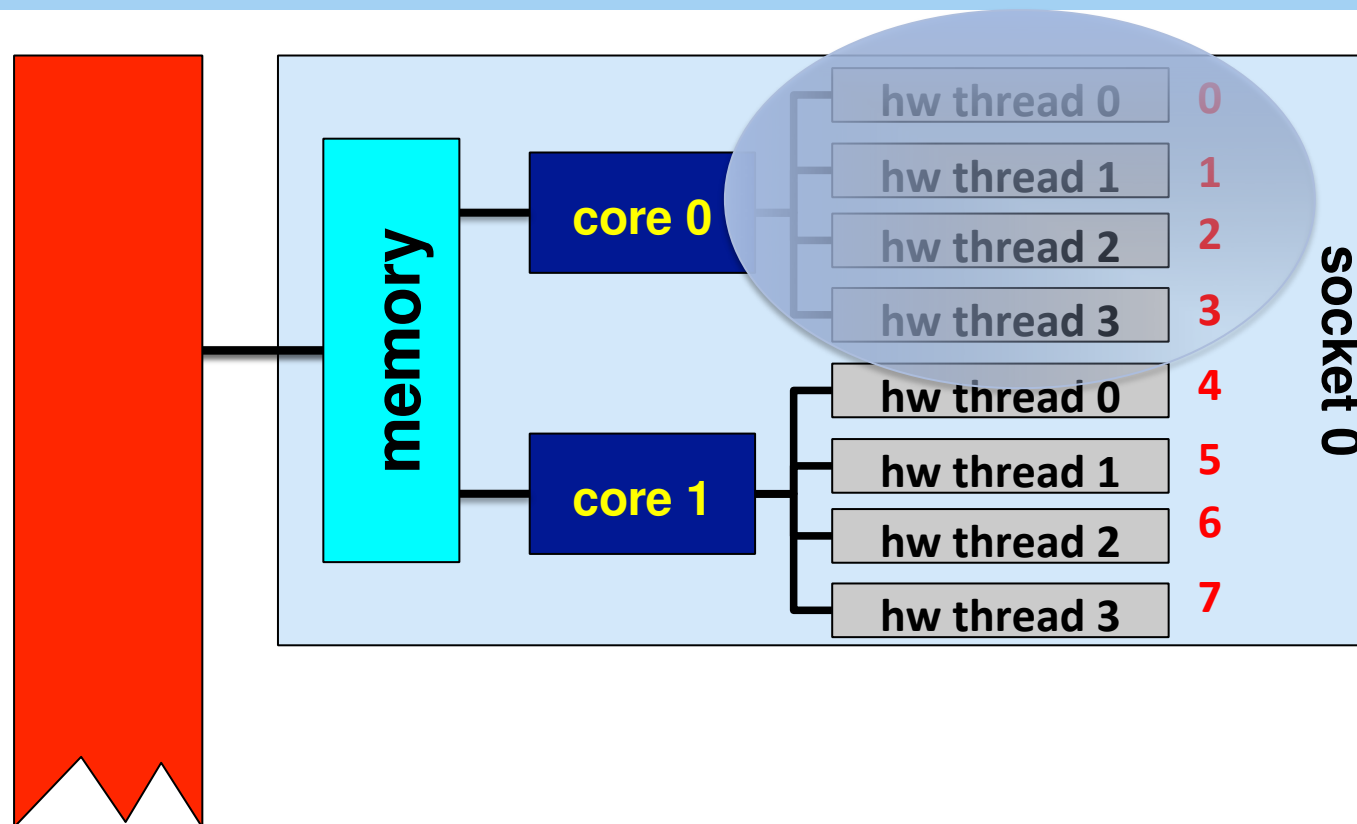


# Example System Architecture



# Places – Example

***{0,1,2,3} identifies the threads in socket 0, core 0***

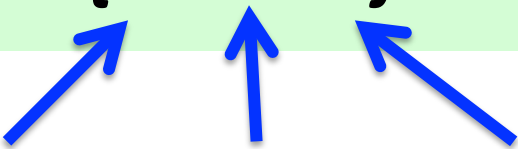


# Places – Example

***{0,1,2,3} identifies the threads in socket 0, core 0***

***Convenient interval notation: {0 : 4 : 1}***

***start : count : stride***



# Places – Order Does Not Matter



*The order of the numbers within a place does not matter*

*$\{0,1,2,3\}$  is the same as  $\{3,2,1,0\}$*

*Assumption*

*No preference regarding memory access time*



# The Place List - Definition

*The Place List consists of a comma separated list of places*

*For example: {0,1,2,3} , {8,9,10,11}*

*The Order Of The Places In The List DOES Matter*

*{0,1,2,3} , {8,9,10,11} **≠** {8,9,10,11} , {0,1,2,3}*

# The Place List – How To Set It



*Environment variable OMP\_PLACES is used to define the place list*

*Example: OMP\_PLACES="{0,1,2,3} , {8,9,10,11}"*

*The interval notation is very convenient*

*Example: OMP\_PLACES="{0:4:1} , {8:4:1}"*

# The Place List – Abstract Names



*Three abstract names are always available:*

***sockets , cores , threads***

*Example: OMP\_PLACES=cores*

*Example: OMP\_PLACES="cores(4)"*

*Note: Implementation can add names*

# Example – OMP\_PLACES



***OMP\_PLACES=cores***

***Equivalent To***

***OMP\_PLACES="{0,1,2,3},{4,5,6,7},{8,9,10,11},{12,13,14,15}"***

***Equivalent To***

***OMP\_PLACES="{0:4:1},{4:4:1},{8:4:1},{12:4:1}"***

***Equivalent To***

***OMP\_PLACES = "{0:4:1}:4:4"***

# *The Affinity Policy*



*Illegal Place*

4

5

*Balcony Place*

3

*Hamburger Place*

6

7

8



9

10

11

*Affinity Policy: Which Place(s) To Use*

*Field Place*

0

*Cheap Place*

1

2

# Thread Affinity Policies



*The Affinity Policy defines which places to use*

*This is defined in a symbolic way:*

*master, close, or spread*

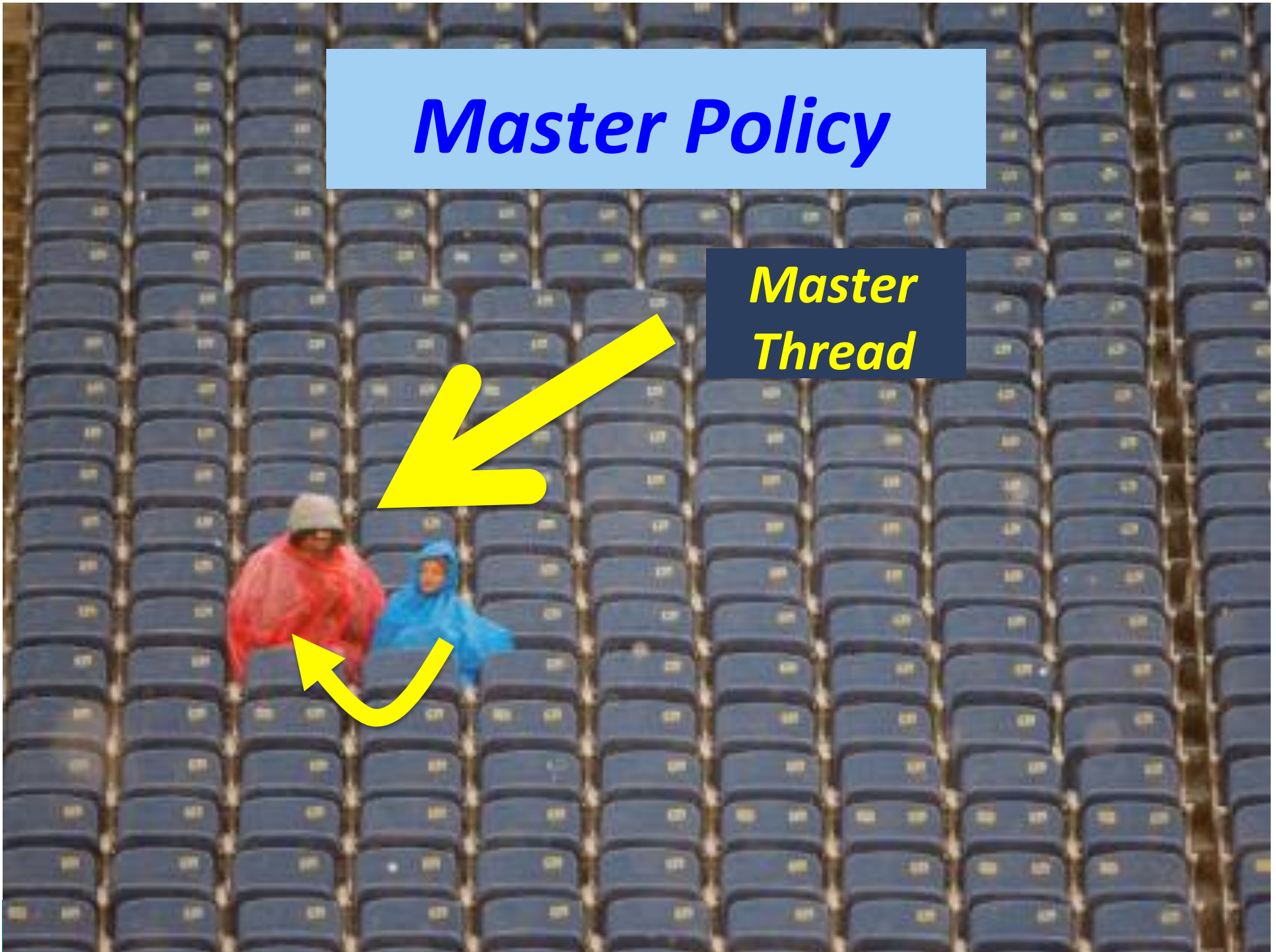
*Each parallel region has such an affinity policy*

*Thread binding to a place is implied*



***Master Policy***

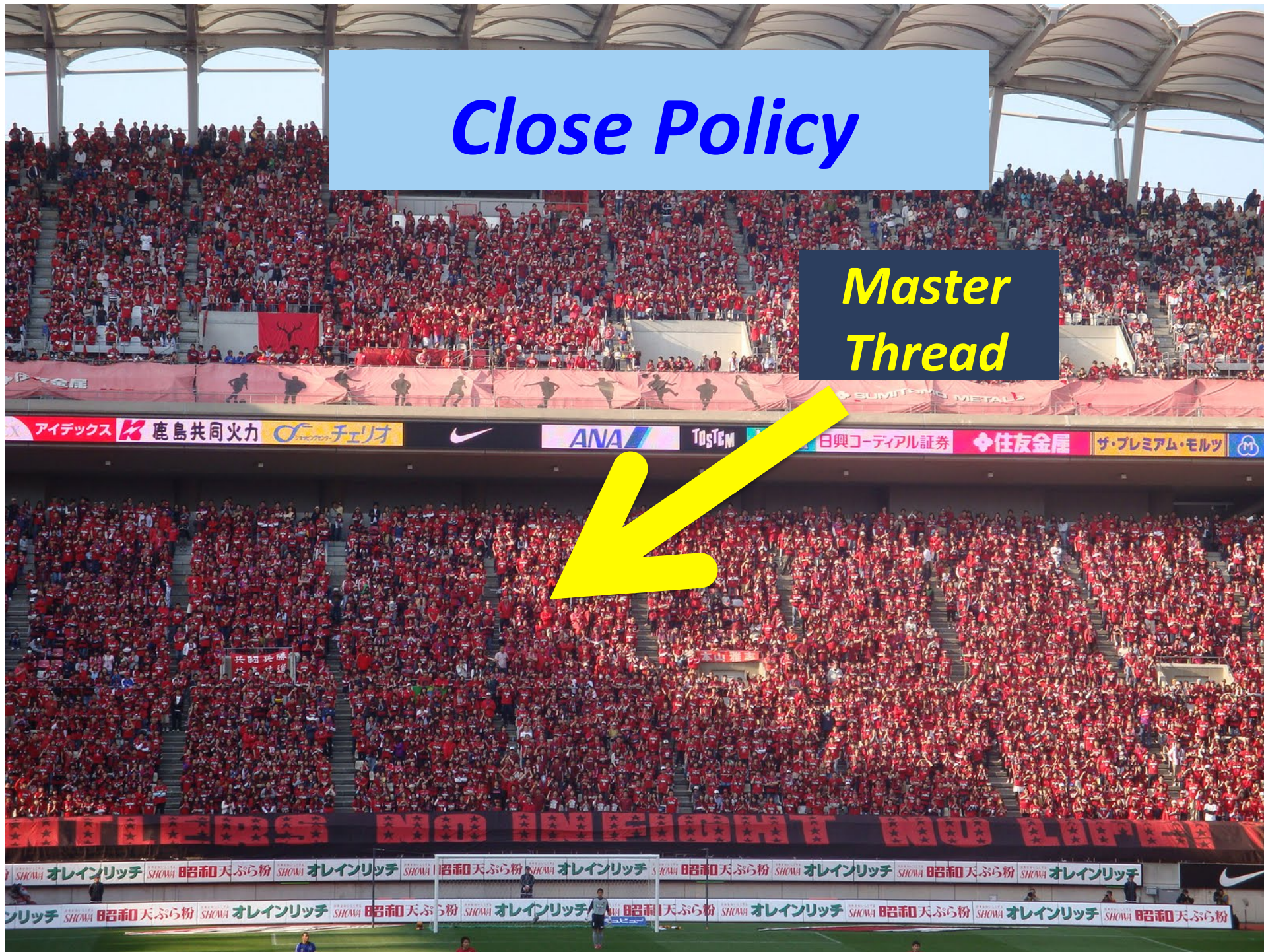
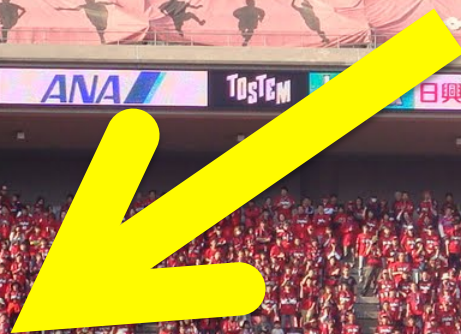
***Master Thread***





*Close Policy*

*Master  
Thread*





# *Spread Policy*

*Master  
Thread*





# Setting The Thread Affinity Policy

*Defined through OMP\_PROC\_BIND*

*Example: OMP\_PROC\_BIND="spread,close"*

*Can also use the "proc\_bind" clause*

*Applied to the current parallel region*

# Recap - Places And Affinity



***The Place List defines what is available  
(fixed for the duration of the program)***

***The Affinity Policy defines thread placement  
(can be set for each parallel region)***

# Example – Spread Policy



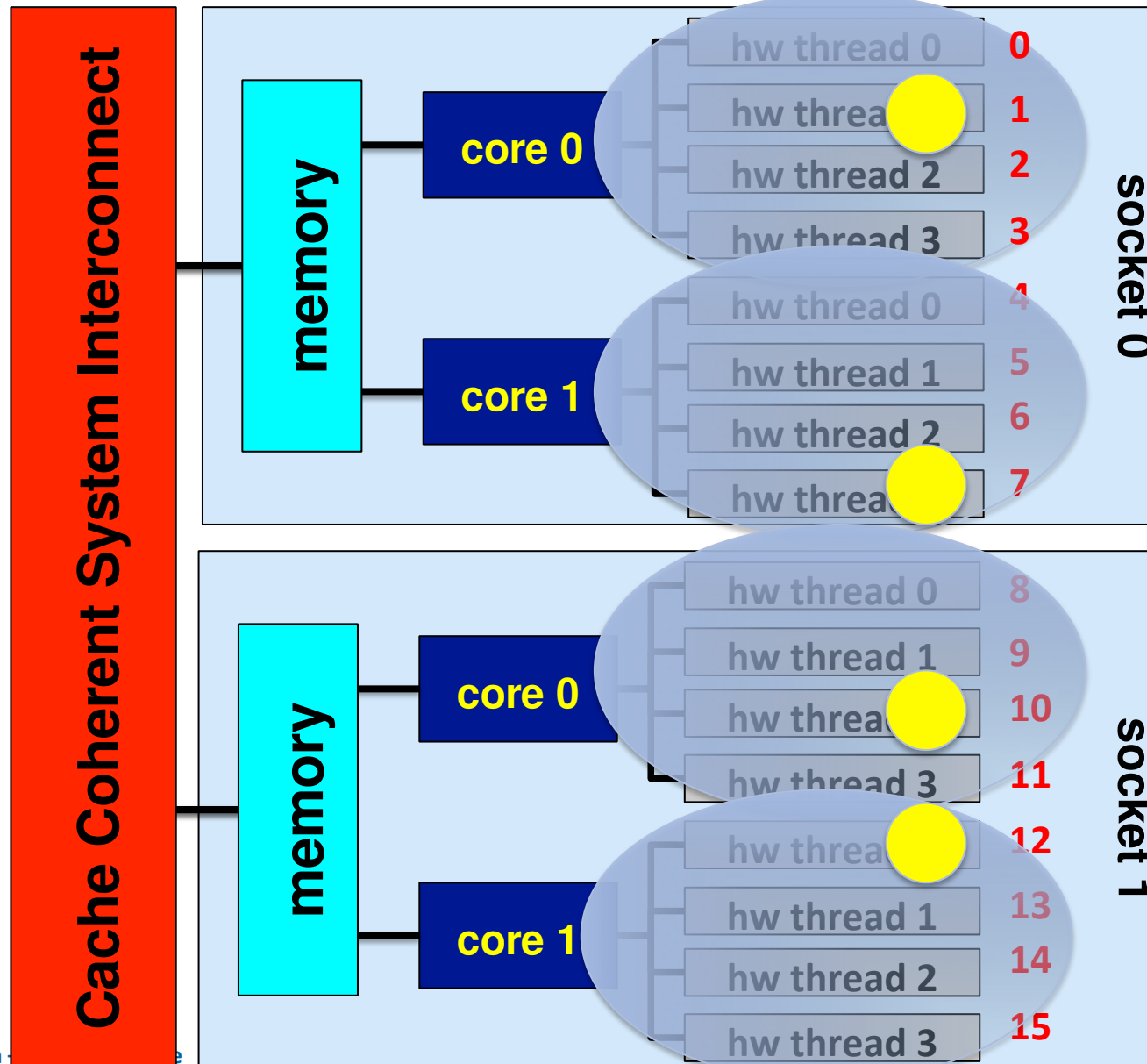
***OMP\_PLACES=cores***  
*(this means 4 places in the place list)*

***OMP\_PROC\_BIND=spread***

***OMP\_NUM\_THREADS=4***

***Result: One OpenMP thread per place***

# Example Spread Affinity Policy



***Thank You And ..... Stay Tuned !***

***[ruud.vanderpas@oracle.com](mailto:ruud.vanderpas@oracle.com)***

The Oracle logo, consisting of the word "ORACLE" in white sans-serif capital letters on a red rectangular background.

ORACLE®

