

HPC na biurku

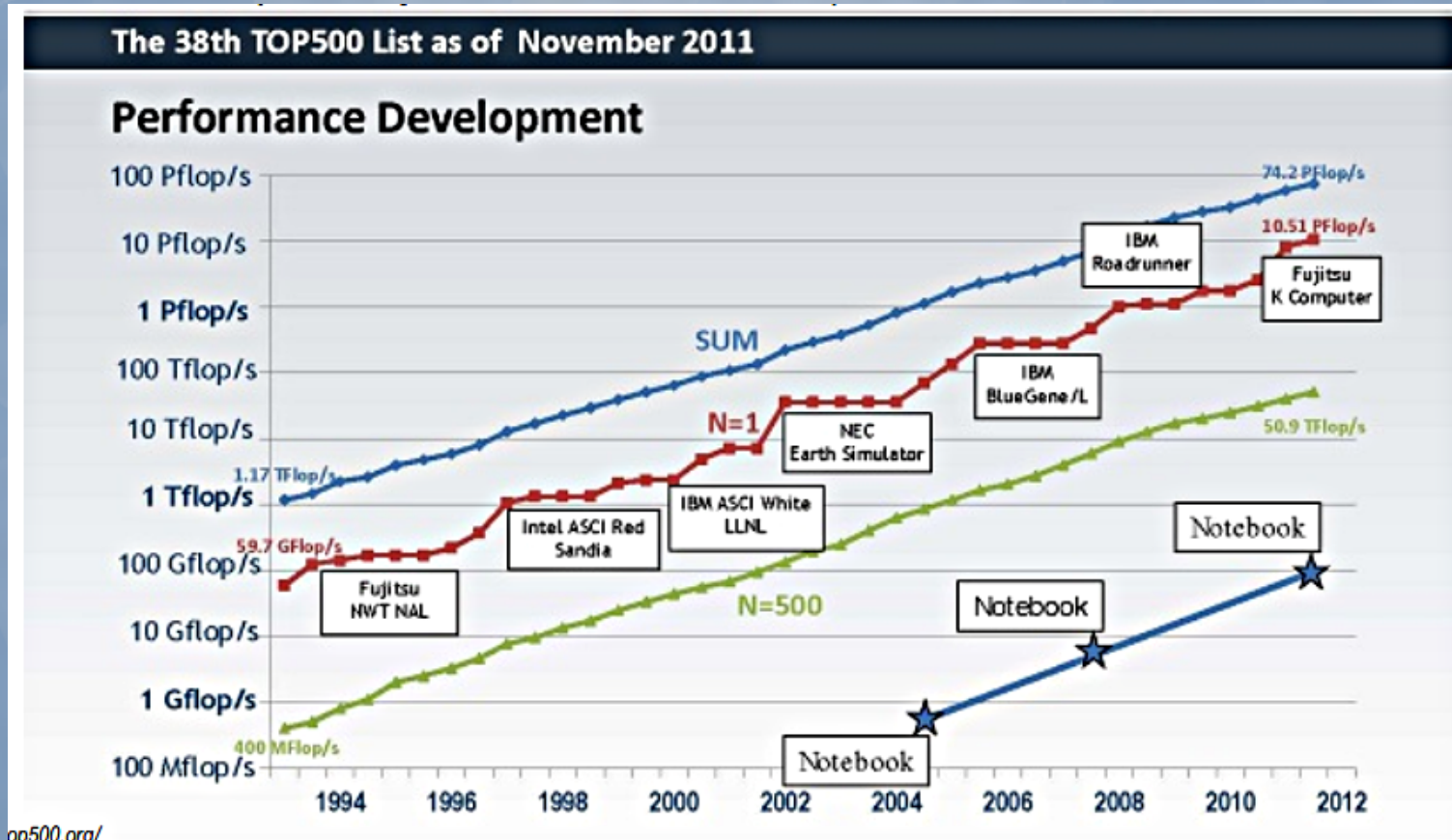
Wojciech Dębski

22.01.2015

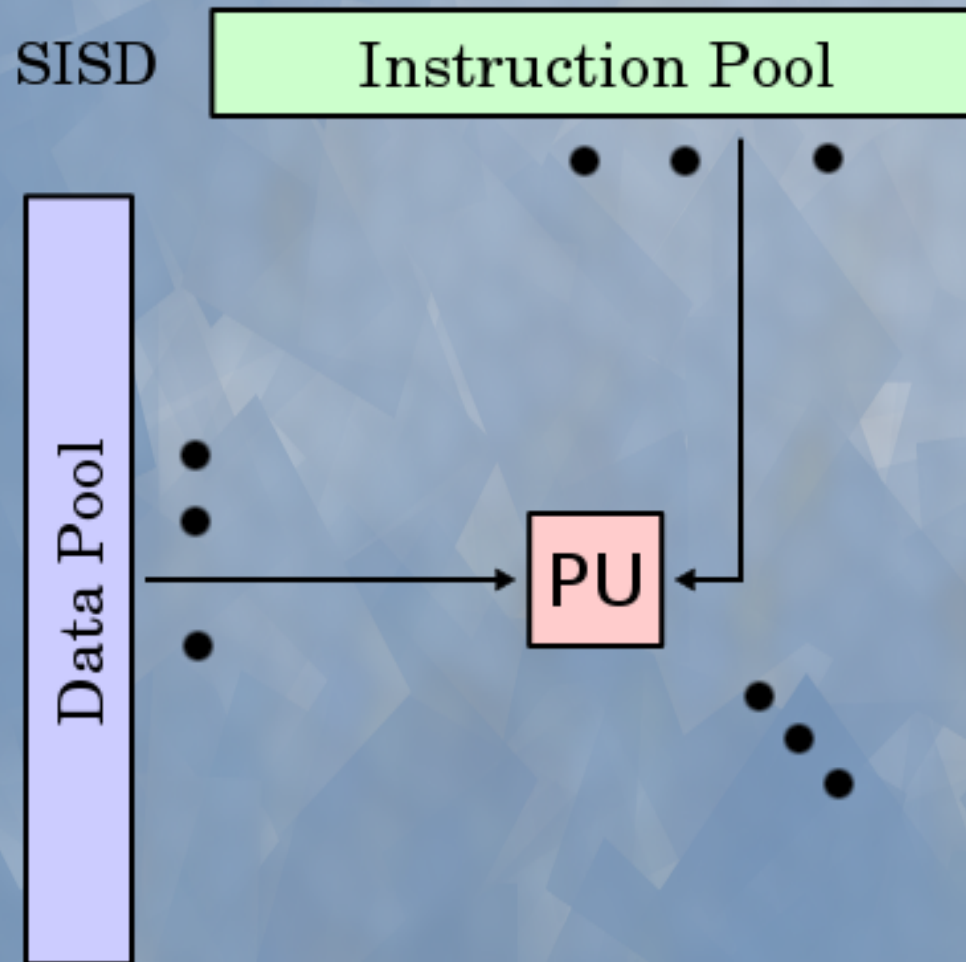
HPC - co to jest?

High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

Historia HPC (Supercomputing)



Obliczenia szeregowe i równoległe, podział według M. Flyna



Klasyczny komputer

```
for(i=0; i<N; i++)  
    c[i] = a[i] + b[i];
```

```
c[0] = a[0] + b[0];  
c[1] = a[1] + b[1];  
.  
.  
.  
c[N] = a[N] + b[N];
```

Obliczenia szeregowe i równoległe

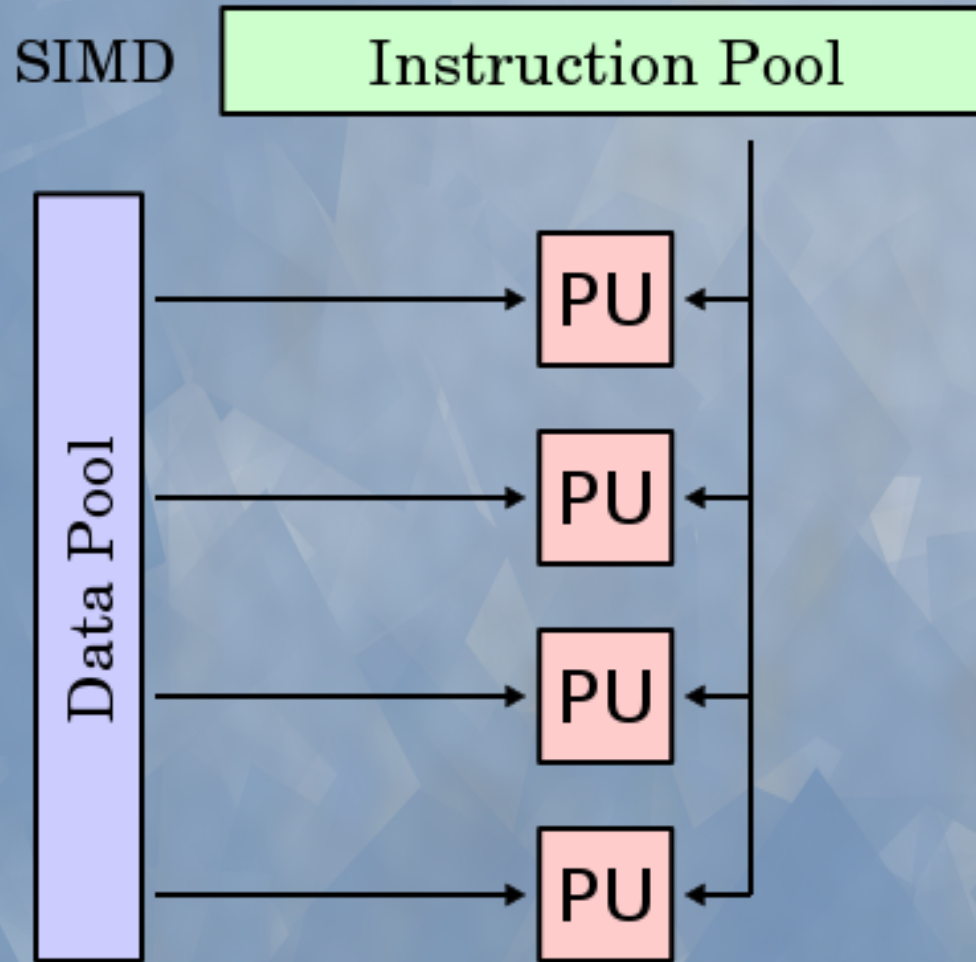


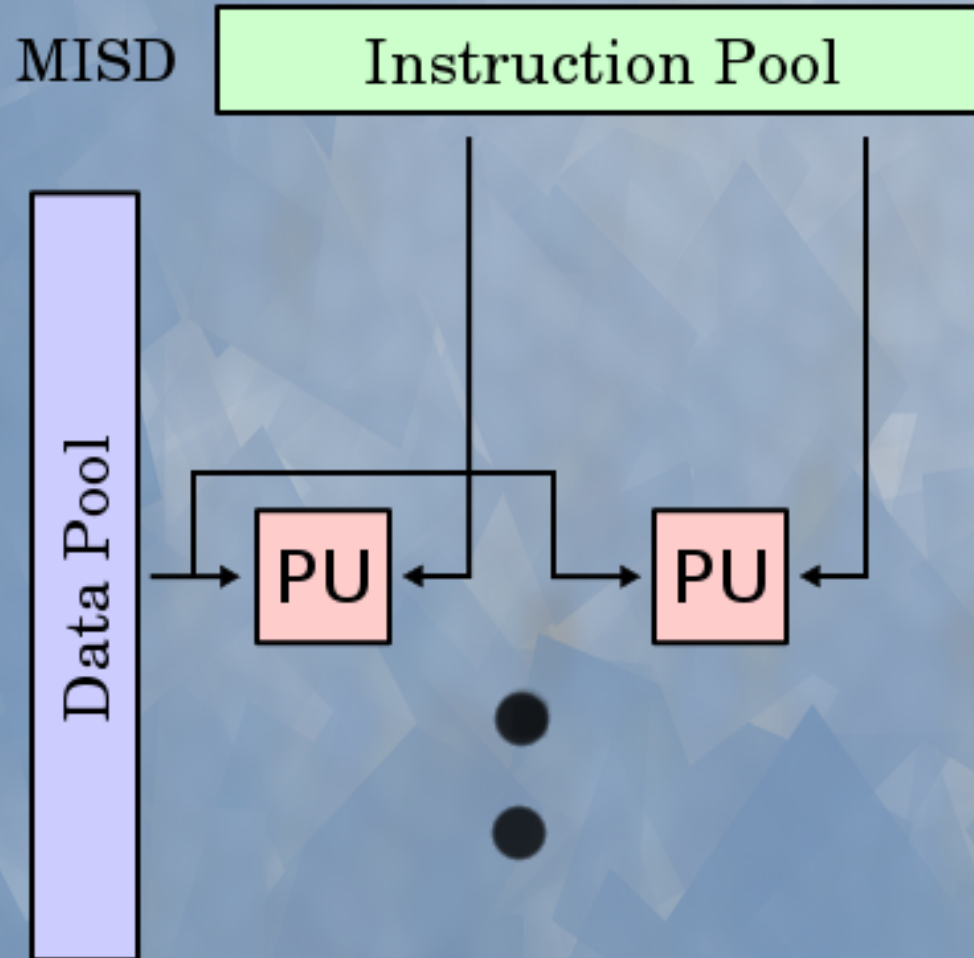
Image processing
(GPU)

$$a = a * 0.5$$

$a[0]$	$a[0] * 0.5$
$a[1]$	$a[1] * 0.5$
$a[2]$	$\Rightarrow a[2] * 0.5$
$a[3]$	$a[3] * 0.5$
$a[4]$	$a[4] * 0.5$



Obliczenia szeregowe i równoległe



Analiza sygnałów
(procesory DSP)

$a = \sin(x)$

$b = \cos(x)$

$a[0] = \sin(x) \quad b[0] = \cos(x)$

$a[1] = \sin(x) \quad b[1] = \cos(x)$

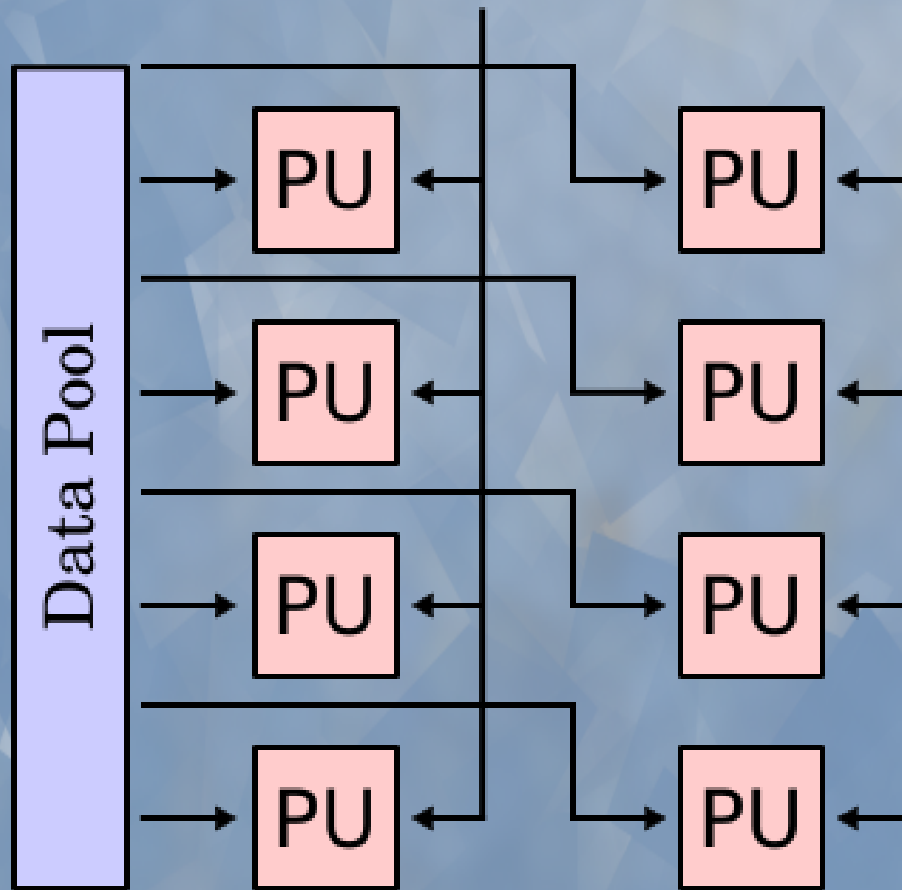
$a[2] = \sin(x) \quad b[2] = \cos(x)$

⋮

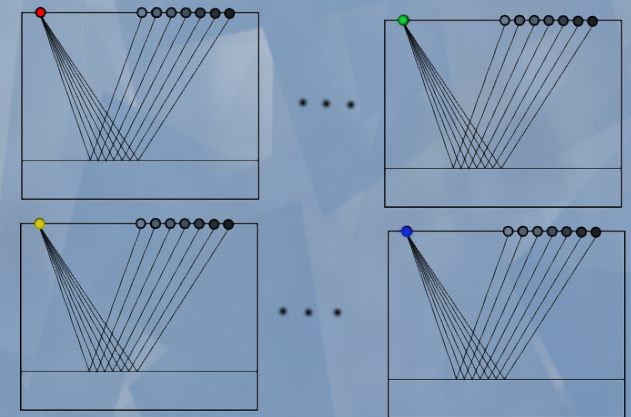
Obliczenia szeregowe i równoległe

MIMD

Instruction Pool



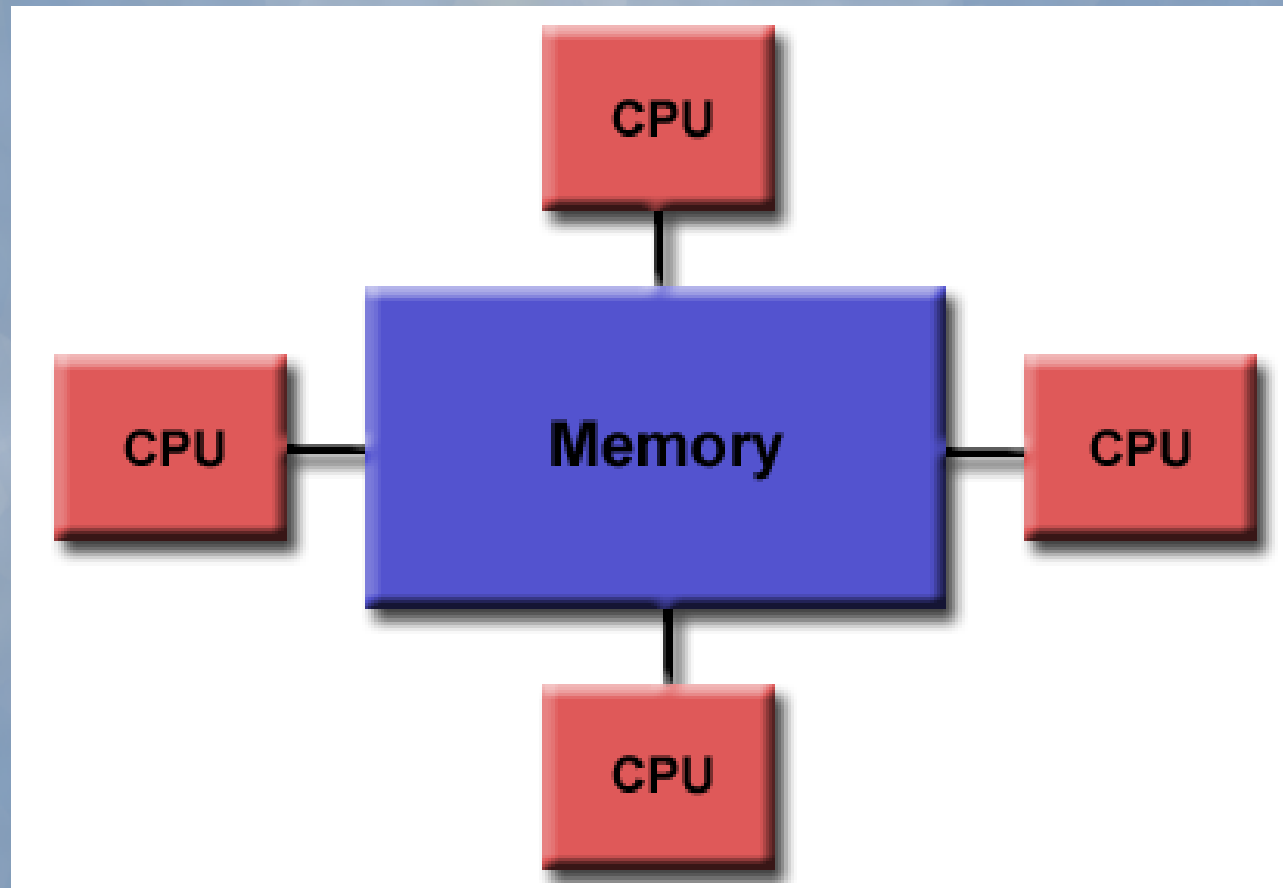
Symulacja pola falowego



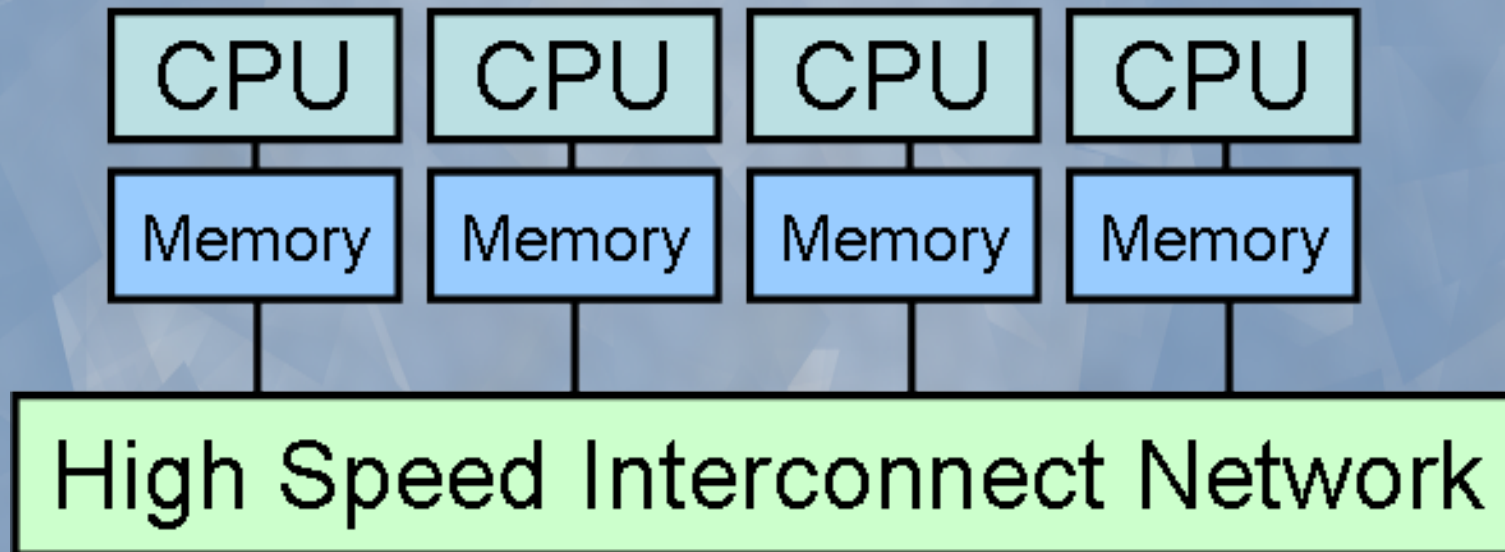
Dwie podstawowe konfiguracje “superkomputerów”

- ◆ systemy rozproszone - wiele “niezależnych” jednostek obliczeniowych komunikujących się ze sobą poprzez zewnętrzną sieć
- ◆ systemy “upakowane” gdzie wszystkie jednostki obliczeniowe dzielą wspólną pamięć przez którą wymieniają informacje

Architektury komputerów - wspólna pamięć



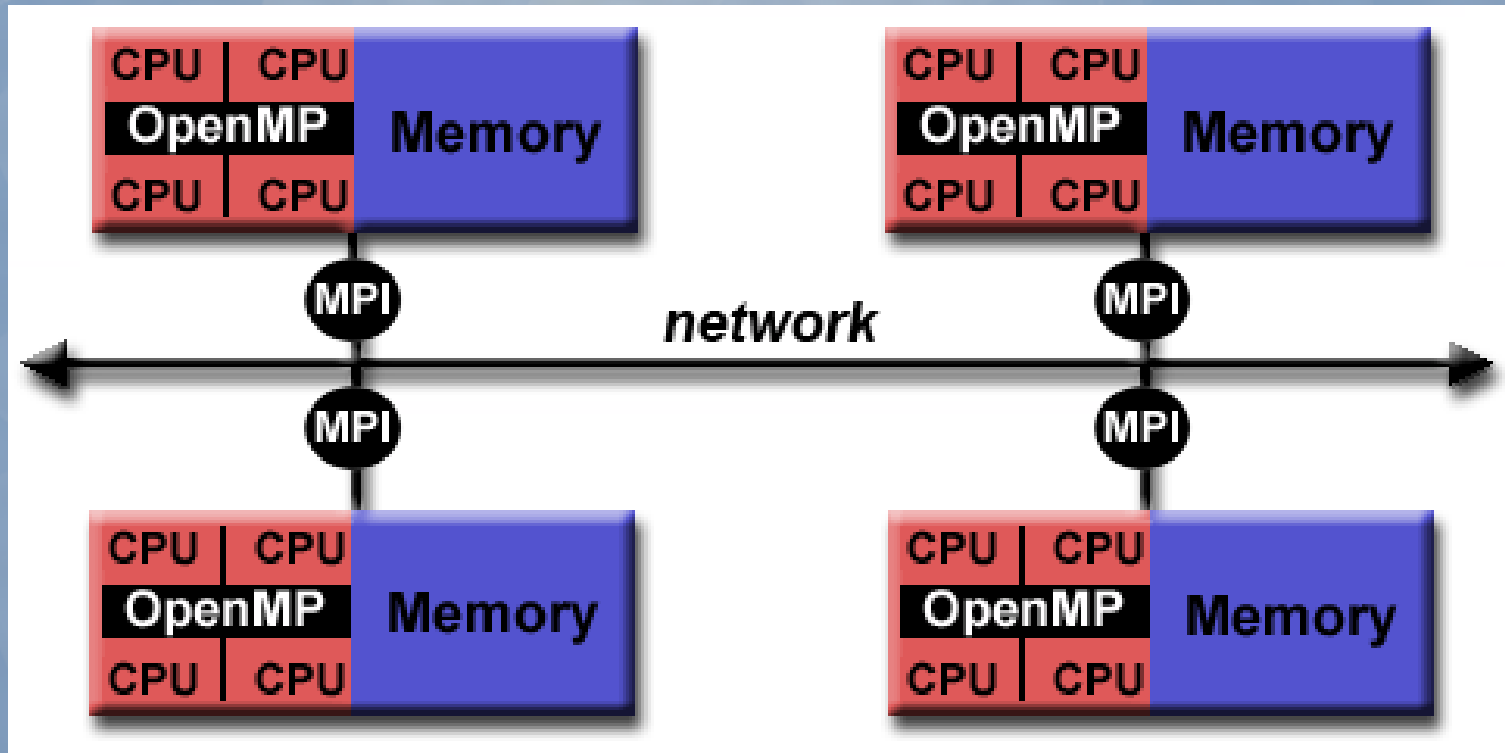
Architektury komputerów - oddzielna pamięć



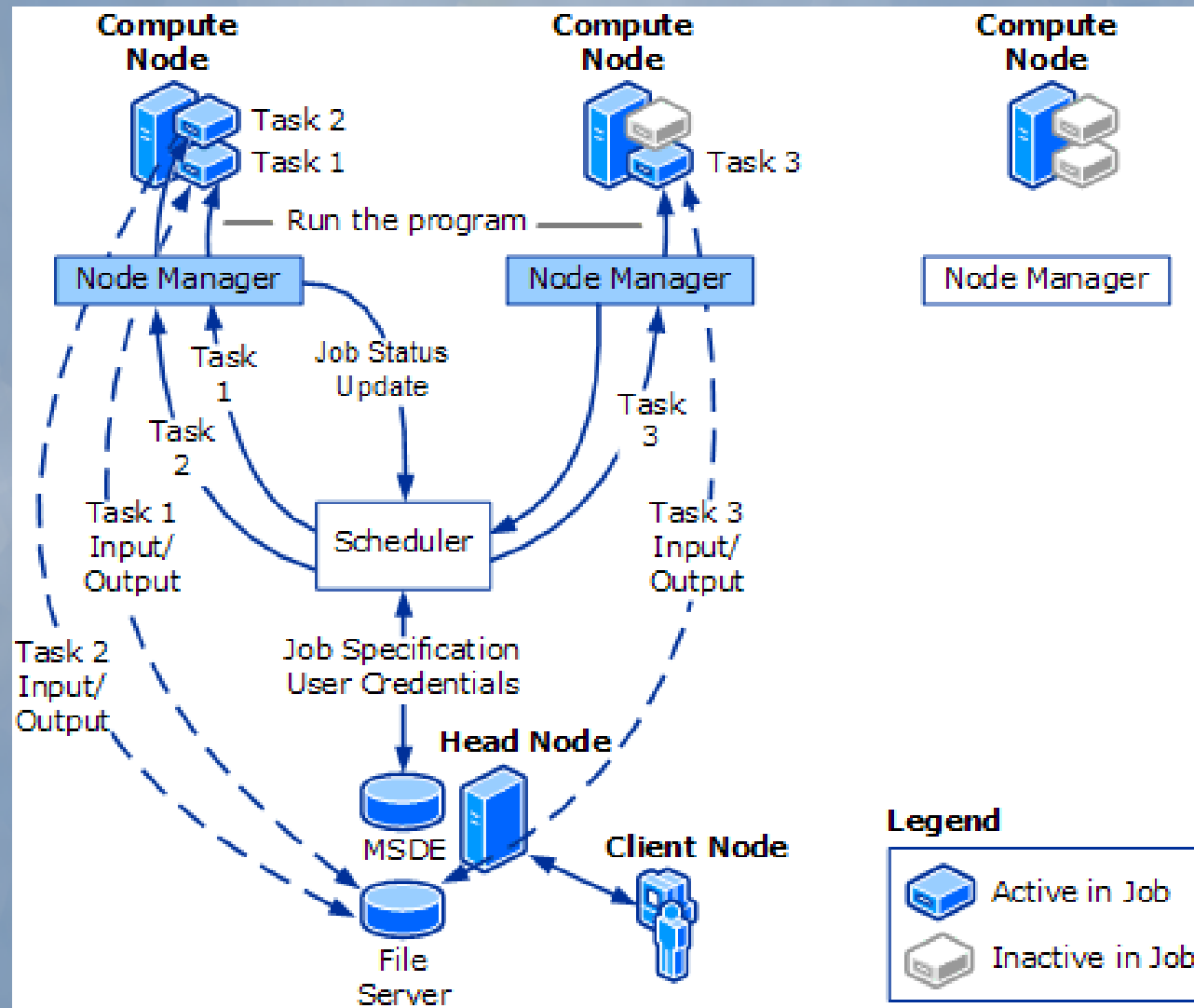
Klastry obliczeniowe - klasyczne rozwiązanie z dzieloną pamięcią



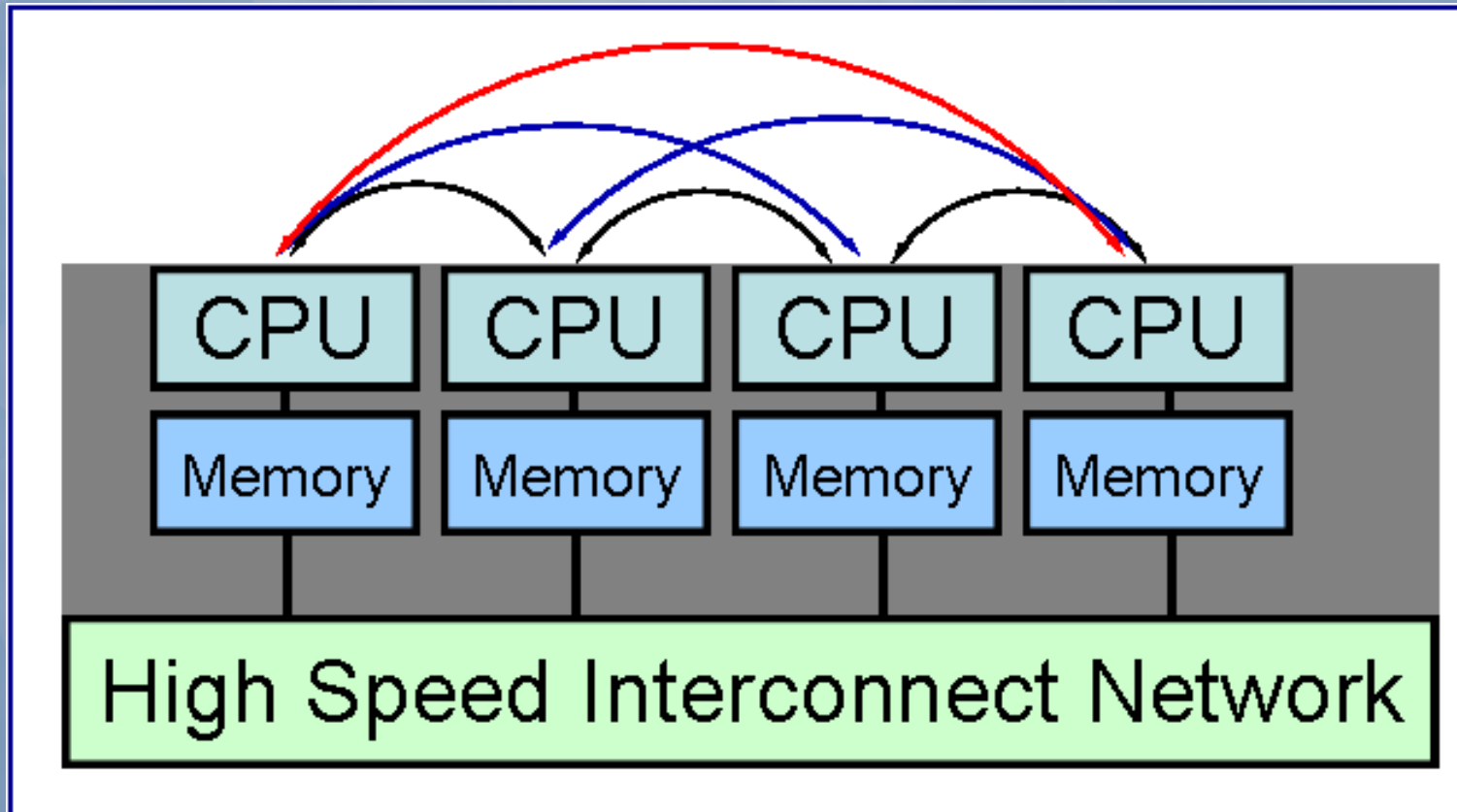
Architektury komputerów - hybrydy



Organizacja klastrów



MPI - Message Passing Interface



MPI - Message Passing Interface

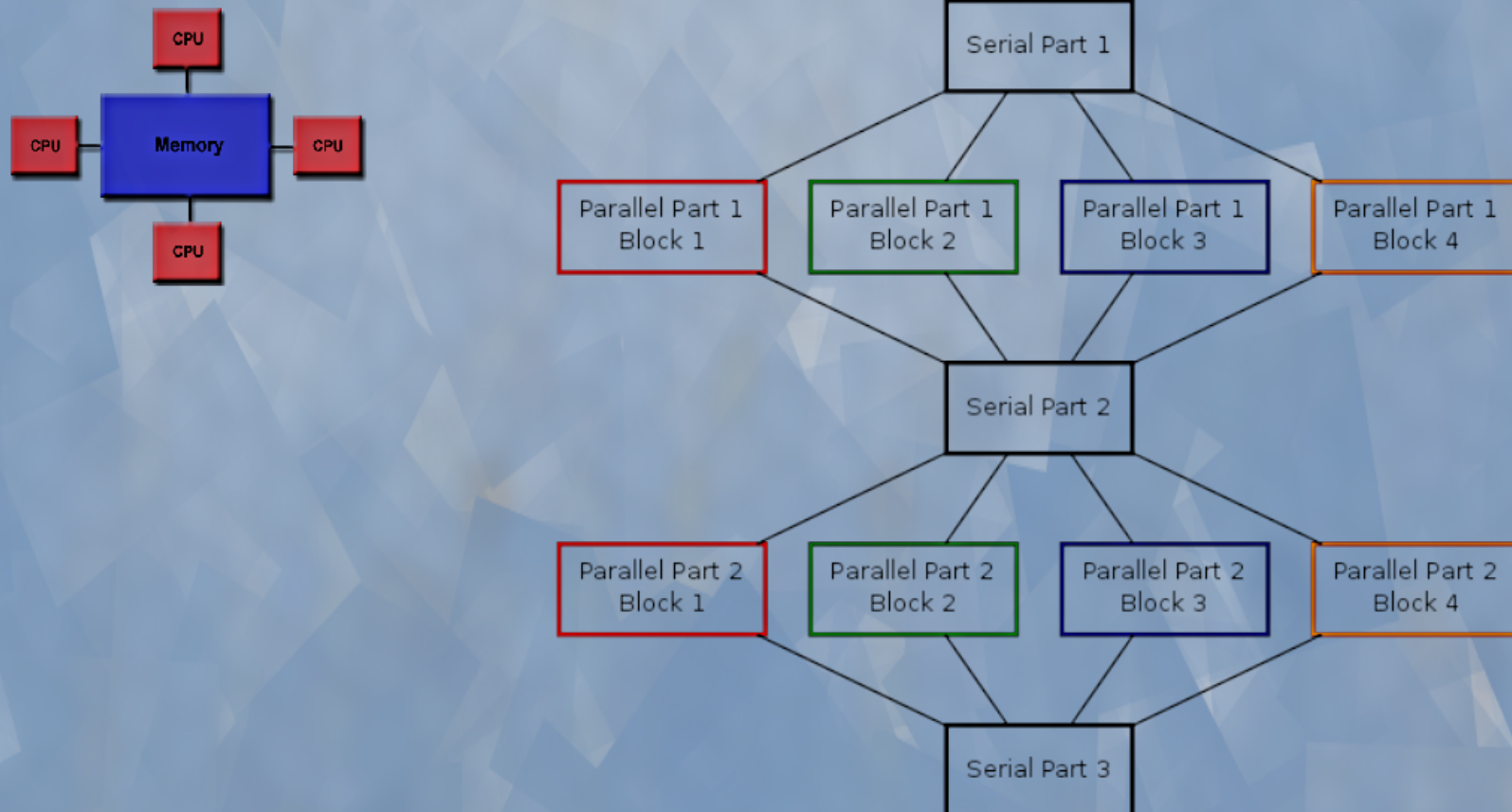
równoległe dodanie wektorów $\vec{c} = \vec{a} + \vec{b}$

`for (i=0; i<N; i++) c[i] = a[i] + b[i]`

```
for (i=0; i<=nx; i++)
    MPI_Send(&c33[i][0], nz+1, MPI_FLOAT, dest, 21, MPI_COMM_WORLD);
for (i=0; i<=nx; i++)
    MPI_Send(&c44[i][0], nz+1, MPI_FLOAT, dest, 21, MPI_COMM_WORLD);
    .
    .
    for (i=0; i<=nxs; i++)
        sxm[i][0] = xs + (float) i * dxs;
    .
    .
for (dest=1; dest<=all_proc-1; dest++)
{ MPI_Recv(&pom, 1, MPI_INT, dest, 101, MPI_COMM_WORLD, &status);
  printf ("Proc:%d Transfer succeed!\n", pom); }
```

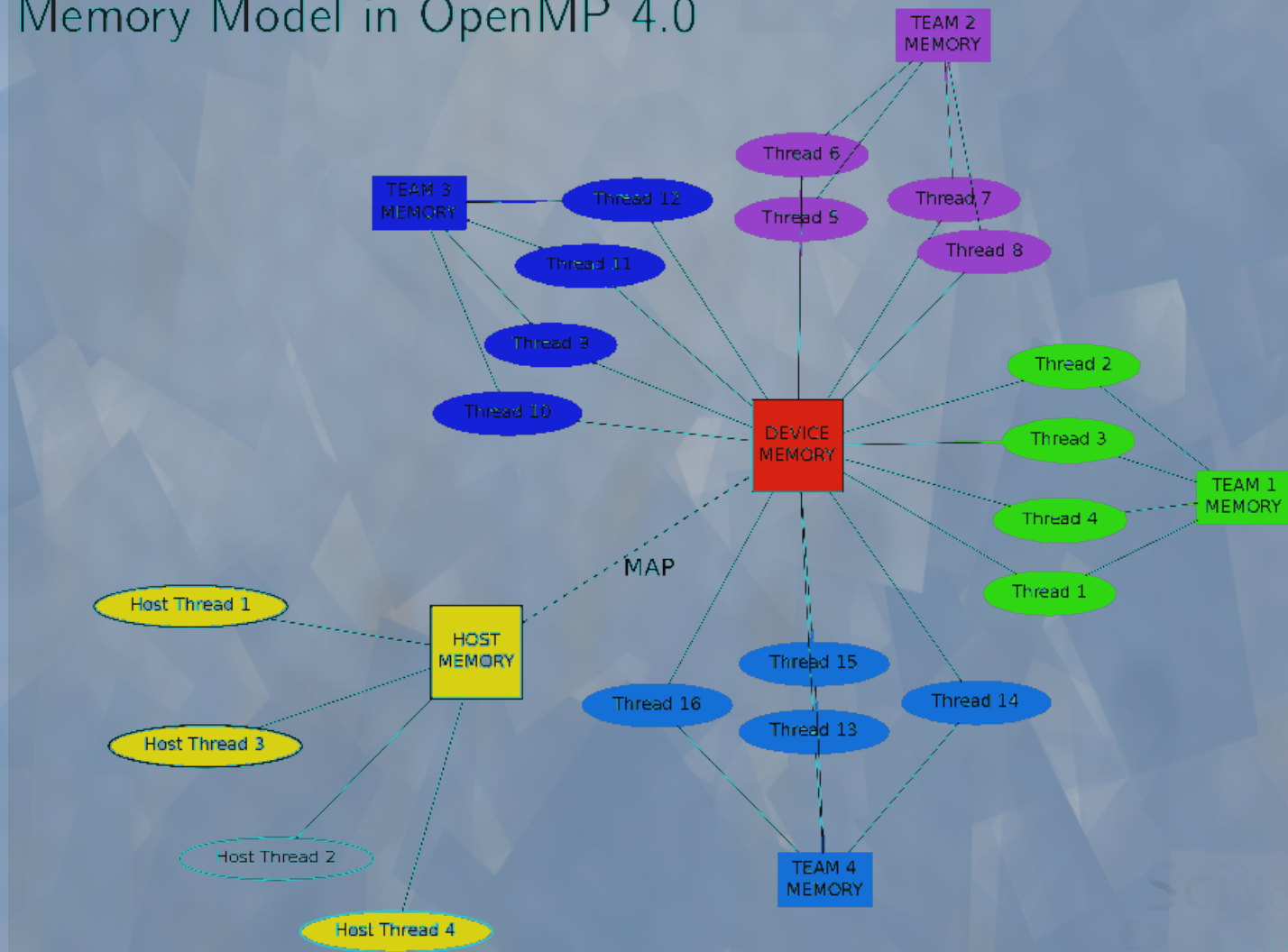
Shared memory model

Execution Model in OpenMP



Shared memory + heterogeneous environment

Memory Model in OpenMP 4.0



GPGPU - general purpose graphic processing units



GPGPU - Tesla K80

- ◆ 4992 procesory potokowe
- ◆ pamięć 24Gb
- ◆ zegar ≈ 1 GHz
- ◆ wydajność ≈ 6 Tflops

Cel seminarium - pytania

1. Jak wykorzystać wielo-rdzeniowość naszych PC'tów?

2. Jak wykorzystać posiadane karty graficzne jako akceleratory obliczeniowe (GPGPU)?

bez konieczności stawiania się zawodowym programistą, uczenia nowego języka programowania, budowy kart graficznych, itp.

Odpowiedź

Wykorzystać modele programowania równoległego dla architektury współdzielonej pamięci:

◆ **OpenMP3**

◆ **OpenMP4**

Wsparcie dla OpenMP

- ◆ kompilator gcc ver. 3.*, ver 4.*
(OpenMP3)
- ◆ kompilator gcc ver 5.* (styczeń 2015)
(OpenMP4)

Co to jest OpenMP ?

OpenMP (*ang. Open Multi-Processing*) jest interfejs programowania aplikacji (API) umożliwiający tworzenie programów komputerowych dla systemów wieloprocessorowych z pamięcią dzieloną. Składa się ze zbioru dyrektyw kompilatora, bibliotek oraz zmiennych środowiskowych mających wpływ na sposób wykonywania programu.

Informacje o OpenMP: <http://openmp.org/wp/>



THE OPENMP® API SPECIFICATION FOR PARALLEL PROGRAMMING



Subscribe to the News Feed

»OpenMP Specifications

»About the OpenMP ARB

»Frequently Asked Questions

»Compilers

»Resources

»Who's Using OpenMP?

»Press Releases

»Videos

»Discussion Forums

Events

»Public OpenMP Calendar

Input Register

Alert the OpenMP.org webmaster about new products, events, or updates and we'll post it here.

»webmaster@openmp.org

 Follow @OpenMP_ARB

Search OpenMP.org

Archives

- January 2015
- December 2014

OpenMP News

»PPCES Videos

A number of high performance computing training videos from the HPC Group at RWTH Aachen University (Germany) are now available on »[YouTube](#). These workshops were held in March 2014.

PPCES (Parallel Programming in Computational Engineering and Science) is the annual introductory HPC event offered by the IT Center of RWTH Aachen University. The 2014 event continues the tradition of previous week-long events that take place in Aachen every spring since 2001.

A wide spectrum of topics were covered, ranging from serial programming and architectures (Monday) to parallel programming using MPI (Tuesday) and OpenMP (Wednesday). Furthermore, they introduced the participants to modern features of the OpenMP standard like vectorization and programming for accelerators and for the Many Integrated Core (MIC) Architecture (Thursday) as well as GPGPU programming with OpenACC (Friday).

Posted on January 13, 2015

»SC14 Videos!

Videos from the in-the-booth and Birds of a Feather sessions at Supercomputing 2014 (New Orleans) are now available »[here](#) and on our »[YouTube channel](#).

Posted on December 15, 2014

»OpenMPCon and IWOMP 2015 Announced

The OpenMP® Architecture Review Board (ARB) announces a new style OpenMP User Conference called **OpenMPCon**. The first OpenMPCon will be held on **September 28-30, 2015 at RWTH Aachen University in Germany**. It will be directly followed by the research-oriented **IWOMP on Oct 1-2, 2015**.

OpenMPCon

OpenMPCon will be the annual, face-to-face gathering organized by the OpenMP community, for the community. The aim is to learn about and showcase usage of OpenMP. Attendees will enjoy keynotes, inspirational talks, and a friendly atmosphere that helps them meet interesting people, learn more about OpenMP from each other, and have a stimulating experience. Multiple diverse technical tracks are being formulated that will appeal to anyone: from the OpenMP novice to the seasoned expert.

Call for Contributions

The **Call for Contributions** has been published, and can be found on the **OpenMPCon website**.

The OpenMP API

supports multi-platform shared-memory parallel programming in C/C++ and Fortran. The OpenMP API defines a portable, scalable model with a simple and flexible interface for developing parallel applications on platforms from the desktop to the supercomputer.

»[Read about OpenMP.org](#)

Get

»OpenMP specs

Use

»OpenMP Compilers

Learn



»[Using OpenMP](#) – the book

»[Using OpenMP](#) – the examples

»[Using OpenMP](#) – the forum

»[Wikipedia](#)

»[OpenMP Tutorial](#)

»[More Resources](#)

Model programowania równoległego OpenMP

- ◆ Zadanie dzielone jest pomiędzy wątki
- ◆ Wątki do obliczeń używają prywatnych kopii zmiennych które są niedostępne dla innych wątków
- ◆ Wątki komunikują się przez zmienne współdzielone (zmienne shared) lub dodatkowe mechanizmy synchronizacji wątków (np. tzw barrier)
- ◆ niebezpieczeństwo tzw. *data race*

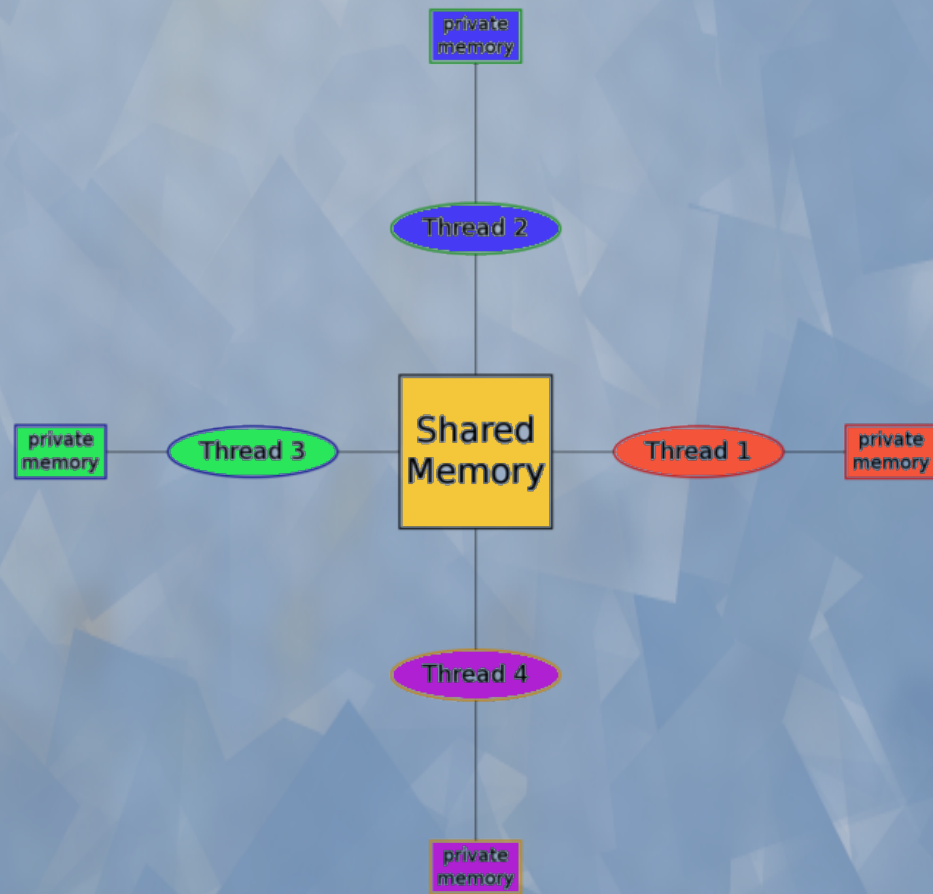
Podstawowa struktura: wątki


$$\textit{for}(i = 0; i < N; i++) c[i] = a[i] + b[i]$$

$$\textit{for}(i = 0; i < 10; i++)$$
$$c[i] = a[i] + b[i]$$
$$\textit{for}(i = 11; i < 20; i++)$$
$$c[i] = a[i] + b[i]$$
$$\textit{for}(i = 21; i < 30; i++) \dots$$
$$c[i] = a[i] + b[i]$$


Używanie pamięci

Memory Model in OpenMP (3.1)



Co musi zrobić programista

- ◆ podać która część ma być zrównoleglona
- ◆ określić które zmienne są “prywatne” dla każdego z wątków”
- ◆ określić warunki synchronizacji

Resztę pracy wykona za nas kompilator !

Przykład pętla “for”

```
#include<stdio.h>
#include<math.h>
#define NS 10000
int main()
{
    double sum=0,x;
    long unsigned int i;

    #pragma omp parallel for private (i), reduction(+:sum)

    for (i=0; i<NS; i++)
        sum+=exp(-i*i);

    printf("Sum=%g\n", sum);

    return;

}
```

Kompilacja:

```
gcc e1.c -fopenmp -lm -o e1
```

Przykład 2

Uwaga na “hazard czasowy: nigdy nie wiadomo który wątek skończy pracę pierwszy

KONIEC