



***czyli programowanie równoległe
przy użyciu interfejsu programowania aplikacji
(API) Open Multi-Processing***

cz. I

Witold Dyrka

14.12.2009

Seminarium Zespołu Bioinformatyki i Biofizyki Nanoporów

Plan seminarium

- O OpenMP
- O wielowątkowości
- Wątki w OpenMP
- Rodzaje segmentów równoległych
- Zmienne w segmentach równoległych
- Planowanie podziału pracy w pętlach
- Materiały



Do czego służy?

Do czego nie służy?

- Do czego służy
 - programowanie równoległe układów wieloprocessorowych z pamięcią dzieloną, w tym popularne **procesory wielordzeniowe**
 - języki programowania **C/C++** i **Fortran**
- Do czego **NIE** służy
 - programowanie równoległe klastrów, czyli grupy niezależnych komputerów (węzłów) – w tym przypadku można użyć np. MPI lub PVM

Standard OpenMP

Members

Permanent Members of the ARB:

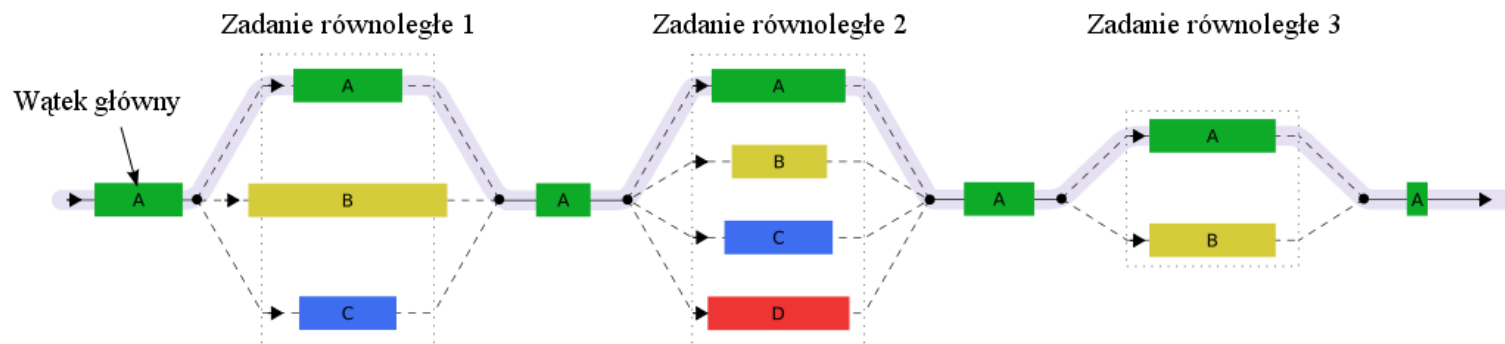
- **AMD** (David Leibs)
- **Cray** (James Beyer)
- **Fujitsu** (Matthijs van Waveren)
- **HP** (Uriel Schafer)
- **IBM** (Kelvin Li)
- **Intel** (Sanjiv Shah)
- **NEC** (Kazuhiro Kusano)
- **The Portland Group, Inc.** (Michael Wolfe)
- **SGI** (Lori Gilbert)
- **Sun Microsystems** (Nawal Copty)
- **Microsoft** (-)

Auxiliary Members of the ARB:

- **ASC/LLNL** (Bronis R. de Supinski)
- **cOMPunity** (Barbara Chapman)
- **EPCC** (Mark Bull)
- **NASA** (Henry Jin)
- **RWTH Aachen University** (Dieter an Mey)

»GNU	gcc (4.3.2)	Free and open source - Linux, Solaris, AIX, MacOSX, Windows Compile with -fopenmp »More information
»IBM	XL C/C++ / Fortran	Windows, AIX and Linux. http://www.ibm.com/software/rational/cafecommunity/ccpp - the IBM C/C++ Community http://www-01.ibm.com/software/awdtools/fortran/forXLFortran http://www-01.ibm.com/software/awdtools/xlcpp/forXL_C/C++ . »More information
»Sun Microsystems	C/C++ / Fortran	Sun Studio compilers and tools - free download for Solaris and Linux. Compile with -xopenmp »More information »Documentation »Tools: Thread Analyzer/Performance Analyzer
»Intel	C/C++ / Fortran (10.1)	Windows, Linux, and MacOSX. Compile with -Qopenmp on Windows, or just -openmp on Linux or Mac OS X »More information
»Portland Group Compilers and Tools	C/C++ / Fortran	»More Information on PGI Compilers Compile with -mp »More Information on PGI OpenMP
»Absoft Pro FortranMP	Fortran	Version 11.0 of the Fortran 95 compiler for 32 and 64 bit Linux includes integrated support for OpenMP 2.5.
»Lahey/Fujitsu Fortran 95	C/C++ / Fortran	»More Information
»PathScale	C/C++ / Fortran	Linux 32/64 bit. »PathScale Compiler Suite User Guide
»HP	C/C++ / Fortran	»More Information
»MS	Visual Studio 2008 C++	Implements OpenMP 2.0 »More Information

Wielowątkowość



- operacje, które można przeprowadzić równoległe są wykonywane niezależnie
- sekwencja operacji, która może być wykonana równoległe z inną sekwencją nazywa się wątkiem (ang. *thread*)
- na przykład?

Wątki w OpenMP

- domyślna ilość wątków w programie?
zmienna środowiskowa `OMP_NUM_THREADS`
- ile wątków może być wykonywanych
rzeczywiście równoległe?

```
#include <omp.h>  
#include <stdio.h>
```

biblioteka
OpenMP

- początek segmentu wykonywanego równoległe
- parametry domyślne

```
int main() {  
    #pragma omp parallel  
    printf("Hello from thread %d, nthreads %d\n", omp_get_thread_num(),  
        omp_get_num_threads());  
}
```

zwraca identyfikator (numer) wątku;
wątek główny ma numer 0 („zero”)

Rodzaje segmentów

- `omp for` lub `omp do`
 - rozdzielenie iteracji pętli pomiędzy wątki
- `sections`
 - przypisanie następujących po sobie niezależnych segmentów do niezależnych różnych wątków
- `single/master`
 - segment kodu przetwarzany przez jeden wątek

Zmienne w segmentach równoległych

- `shared (zm1, zm2, ...)`
 - wątki operują na wspólnej kopii zmiennej
 - domyślne, z wyjątkiem licznika pętli
- `private (...)`
 - wątek operuje na własnej lokalnej kopii zmiennej
 - zmienne są niezainicjowane!
- `firstprivate (...)`
 - zmienna prywatna inicjalizowana wartością zmiennej o tej samej nazwie z wątku głównego
- `lastshared (...)`
 - zmienna prywatna, która zwraca wartość do zmiennej o tej samej nazwie z wątku głównego

Zmienne w segmentach równoległych (2)

- `threadprivate, copyin, copyprivate`
- `default (shared|private|none)`
 - ustalenie domyślnego traktowania zmiennych

Planowanie podziału pracy w pętlach

- składnia: `schedule (type, chunk)`
- `type`:
 - `static` – iteracje (w blokach o rozmiarze `chunk`) przydzielane są wątkom „z góry”
 - `dynamic` – wątek pobiera blok iteracji, gdy skończył poprzedni blok
 - `guided` – jak wyżej ale rozmiar bloków zmniejszany wykładniczo
 - `auto`
 - `runtime`

Materiały

- <http://www.OpenMP.org>
 - strona oficjalna
- <http://pl.wikipedia.org/wiki/OpenMP>
 - polskojęzyczne źródło informacji
- <https://computing.llnl.gov/tutorials/openMP/>
 - by Blaise Burney, LLNL, USA
 - przykłady „hello” i „workshare1”
 - ćwiczenia „bug1” i „bug2”

Kompilacja pod GCC (>4.2): gcc -fopenmp omp_hello.c -o hello