Programowanie systemów pomiarowych
w.5

dr inż. Dawid Kucharski

Zakład Metrologii i Systemów Pomiarowych
Instytut Technologii Mechanicznej
Wydział Budowy Maszyn i Zarządzania
Politechnika Poznańska

19.02.2019 13:13

C

Wieloplatformowy, Strukturalny język programowania niskiego poziomu

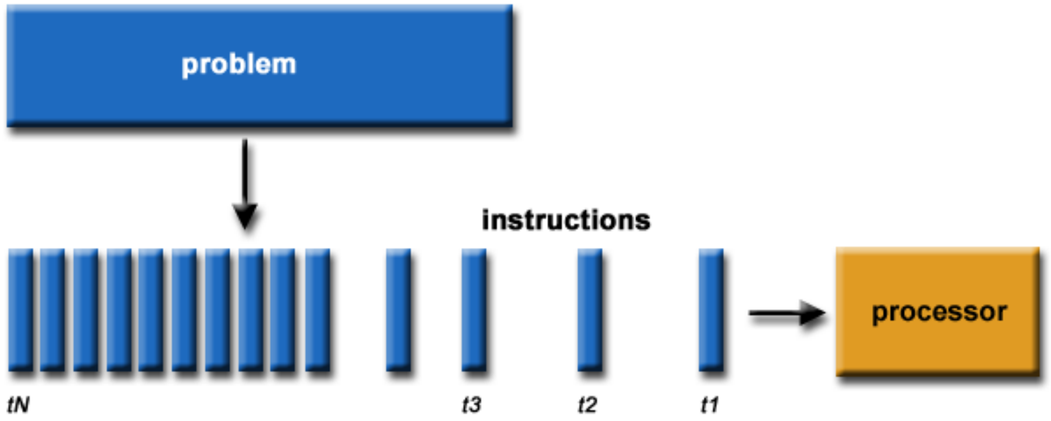
W programowaniu systemów pomiarowych wykorzystywany do sterowania i analizy danych

GCC

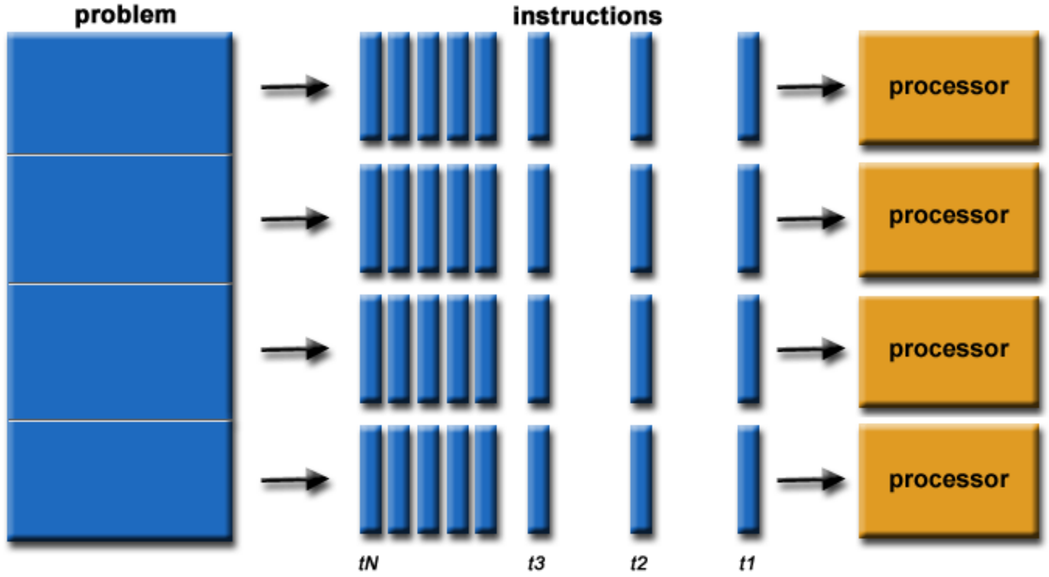
GNU Compiler Collection (GCC) – zestaw kompilatorów o otwartym kodzie źródłowym rozwijany w ramach Projektu GNU. Rozpowszechniany jest na licencji GPL oraz LGPL.

GCC jest podstawowym kompilatorem w systemach uniksopodobnych, przy czym szczególnie ważną rolę odgrywa w procesie budowy jądra Linuksa.

Obliczenia szeregowe

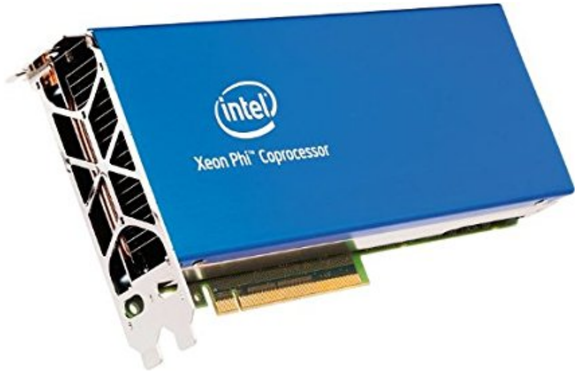


Obliczenia równoległe



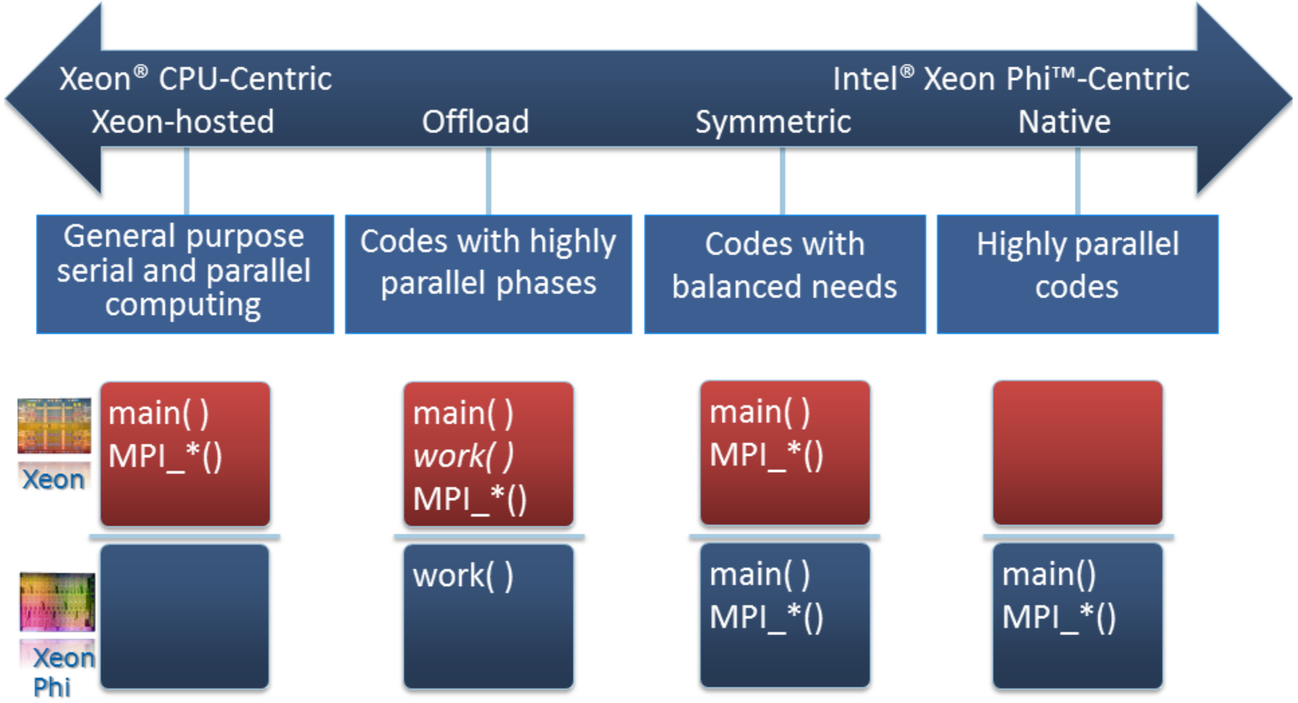
„Świat jest bardziej równoległy”

C; programowanie równoległe



Intel® Manycore Platform

Software Stack (Intel® MPSS)



Intel® Manycore Platform

Software Stack (Intel® MPSS)

Supported Host OS Versions	Kernel Version
Red Hat* Enterprise Linux* 64-bit 6.6	2.6.32-504
Red Hat* Enterprise Linux* 64-bit 6.7	2.6.32-573
Red Hat* Enterprise Linux* 64-bit 7.0	3.10.0-123
Red Hat* Enterprise Linux* 64-bit 7.1	3.10.0-229
Red Hat* Enterprise Linux* 64-bit 7.2	3.10.0-327
SUSE* Linux* Enterprise Server 11 SP4 64-bit	3.0.101-63-default
SUSE* Linux* Enterprise Server 12 64-bit	3.12.28-4-default
SUSE* Linux* Enterprise Server 12 SP1 64-bit	3.12.49-11-default

Intel® Manycore Platform

Software Stack (Intel® MPSS)

Sprawdzenie obecności koprocesora

```
[host]$ lspci | grep -i Co-processor
08:00.0 Co-processor: Intel Corporation Device 225c (rev 20)
```

Sprawdzenie ustawień BIOS

```
[host]# lspci -s 08:00.0 –vv
08:00.0 Co-processor: Intel Corporation Device 225c (rev 20) Subsystem: Intel Corporation Device 2500
Physical Slot: 4
Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV-
VGASnoop- ParErr+ Stepping- SERR+ FastB2B- DisINTx+ Status: Cap+ 66MHz- UDF- FastB2B- ParErr-
DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx- Latency: 0, Cache Line Size: 64 bytes
Interrupt: pin A routed to IRQ 56
Region 0: Memory at 3c7e00000000 (64-bit, prefetchable)
[size=200000000]
Region 4: Memory at ec000000 (64-bit, non-prefetchable)
[size=128K]
        : <output truncated>
```

Instalacja sterowników

```
[host]$ tar xvf mpss-3.7-linux.tar
[host]$ cd mpss-3.7
```

Deinstalacja poprzednich wersji

```
[host]$ rpm -qa | grep -e intel-mic -e mpss
[host]$ cd $MPSS3X
[host]# ./uninstall.sh
```

Red Hat* Enterprise Linux*

```
[host]# yum remove intel-mic\*
```

SUSE* Linux* Enterprise Server

```
[host]# zypper remove intel-mic\*
```

Przebudowa składników

Red Hat* Enterprise Edition (RHEL*)

[host]# yum install kernel-headers kernel-devel

Regenerate the Intel® MPSS driver module package:

[host]$ cd $MPSS37/src/

[host]$ rpmbuild --rebuild mpss-modules*.src.rpm

[host]$ cd $HOME/rpmbuild/RPMS/x86_64

[host]$ cp mpss-modules*`uname -r`*.rpm $MPSS37/modules

SUSE* Linux* Enterprise Server (SLES*)

[host]# zypper install kernel-default-devel rpm-build

[host]$ cd $MPSS37/src/

[host]# rpmbuild --rebuild mpss-modules*.src.rpm

[host]$ cd /usr/src/packages/RPMS/x86_64

[host]$ cp mpss-modules*`uname -r`*.rpm $MPSS37/modules

Instalacja MPSS

```
[host]$ cd $MPSS37
[host]$ cp ./modules/*`uname -r`*.rpm .
```

Red Hat* Enterprise Linux*

```
[host]# yum install *.rpm
[host]# yum install --nogpgcheck *.rpm
```

SUSE* Linux* Enterprise Server

```
[host]# zypper install *.rpm
[host]# modprobe mic
```

Aktualizacja pamięci flash koprocesora

[host]# micflash -getversion

[host]$ micctrl -s

[host]# micctrl –rw

[host]# micflash -update -device all

[host]# micflash -update -device all -smcbootloader (kontroler zarządzania)

Zainicjowanie MPSS

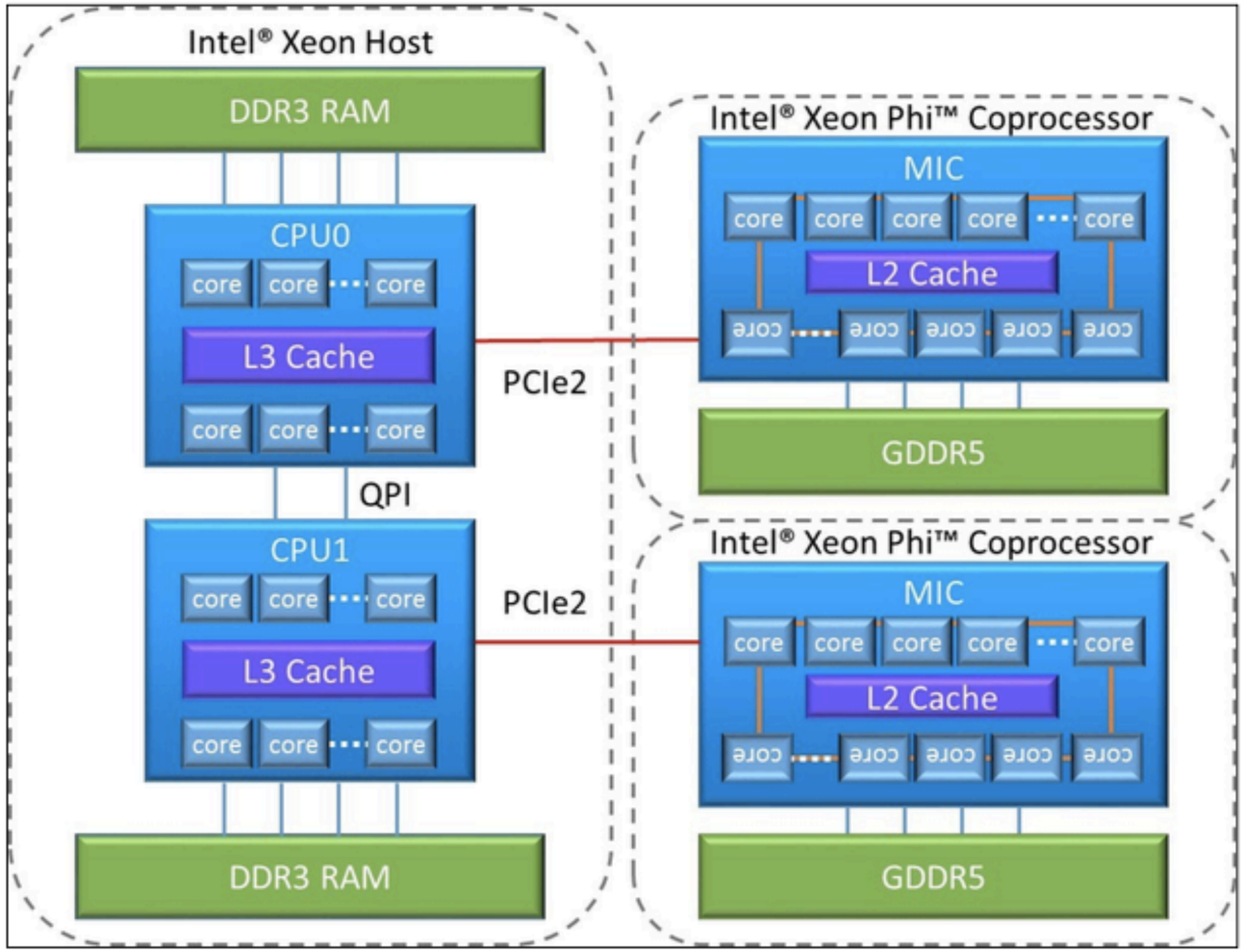
```
mic0:
==========================================================
    Config Version: 1.1

    Linux Kernel:   /usr/share/mpss/boot/bzImage-knightscorner
    BootOnStart:    Enabled
    Shutdowntimeout: 300 seconds

    ExtraCommandLine: highres=off
    PowerManagment: cpufreq_on;corec6_off;pc3_on;pc6_off

Root Device:   Dynamic Ram Filesystem /var/mpss/mic0.image.gz
from:
Base:      CPIO /usr/share/mpss/boot/initramfs-
           knightscorner.cpio.gz
CommonDir: Directory /var/mpss/common
Micdir:    Directory /var/mpss/mic0

Network:       Static Pair
    Hostname:  snhondo-desktop7-mic0.dd.domain.com
    MIC IP:    172.31.1.1
    Host IP:   172.31.1.254
    Net Bits:  24
    NetMask:   255.255.255.0
    MtuSize:   64512
    MIC MAC:   4c:79:ba:15:00:1e
    Host MAC:  4c:79:ba:15:00:1f

Cgroup:
Memory:    Disabled

Console:        hvc0
VerboseLogging: Disabled
CrashDump:      /var/crash/mic 16GB
```

[host]$ micctrl --config

Uruchomienie koprocesora

```
[host]# service mpss start
[host]$ ssh mic0
[mic0]$
```

Progress in parallel computation with Xeon Phi architecture



Typical Intel® Xeon Phi™ Based Workstation Configuration

[dkucharski@zmisp-253 ~]$ lspci | grep Co-processor
03:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 3120 series (rev 20)
[dkucharski@zmisp-253 ~]$ micinfo

MicInfo Utility Log
Created Mon Nov 7 13:17:08 2016
System Info
HOST OS : Linux
OS Version : 3.10.0-327.28.3.el7.x86_64
Driver Version : 3.7.2-1
MPSS Version : 3.7.2
Host Physical Memory : 31897 MB

Device No: 0, Device Name: mice
Version
Flash Version : 2.1.02.0391
SMC Firmware Version : 1.17.6900
SMC Boot Loader Version : 1.8.4326
Coprocessor OS Version : 2.6.38.8+mpss3.7.2
Device Serial Number : ADKC32800470

Cores
Total No of Active Cores : 57
Voltage : 0 uV
Frequency : 1100000 kHz

Progress in parallel computation with Xeon Phi architecture

[dkucharski@zmisp-253 ~]$ gcc '/home/dkucharski/hello.c' -fopenmp

[dkucharski@zmisp-253 ~]$ '/home/dkucharski/a.out'

HOST: Hello world from HOST. I am zmisp-253.mt.put.poznan.pl and I have 12 logical cores. I was called from host: zmisp-253.mt.put.poznan.pl

[dkucharski@zmisp-253 ~]$ /opt/mpss/3.7.2/sysroots/x86_64-mpsssdk-linux/usr/bin/k1om-mpss-linux/k1om-mpss-linux-gcc '/home/dkucharski/hello.c' -o hello_mic

[dkucharski@zmisp-253 ~]$ sudo scp hello_mic mic0:
hello_mic   100%   11KB 11.0KB/s   00:00

[dkucharski@zmisp-253 ~]$ sudo ssh mic0

[root@zmisp-253-mic0 ~]# ls
hello_mic

[root@zmisp-253-mic0 ~]# ./hello_mic
MIC: Hello world from MIC. I am zmisp-253-mic0.mt.put.poznan.pl and I have 228 logical cores. I was called from host: zmisp-253-mic0.mt.put.poznan.pl
[[[[[[[[[[[[[r[root@[root@zmisp-[root@zm[root@zmisp-253-[root@zmisp-253-mic0 ~]#

Progress in parallel computation with Xeon Phi architecture

$$\nabla^2 \mathrm{A} + k^2 \mathrm{A} = 0$$

[dkucharski@zmisp-253 ~]$ '/home/dkucharski/helmholtz'

HELMHOLTZ
C/OpenMP version

A program which solves the 2D Helmholtz equation.

This program is being run in parallel.

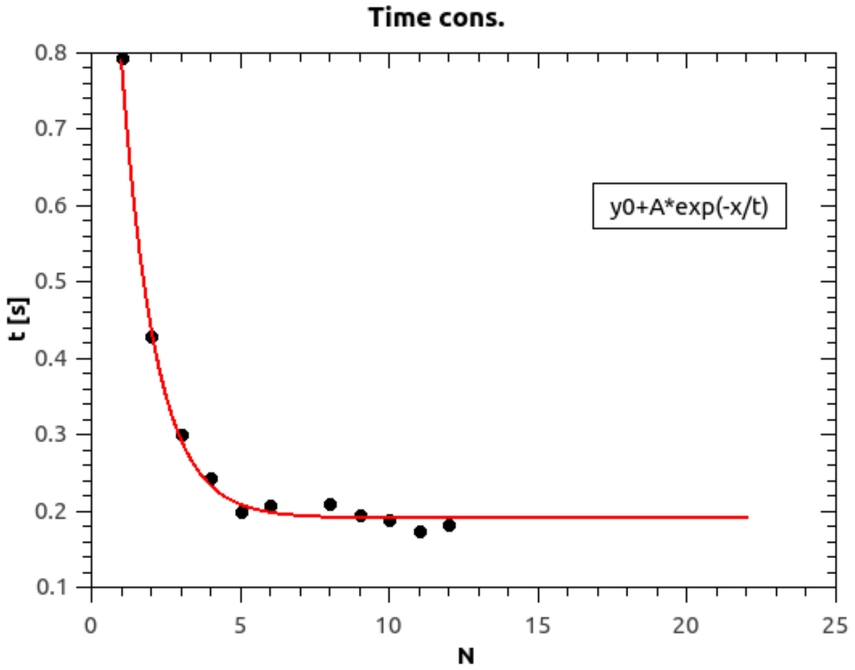
Number of processors available = 12
Number of threads = **12**

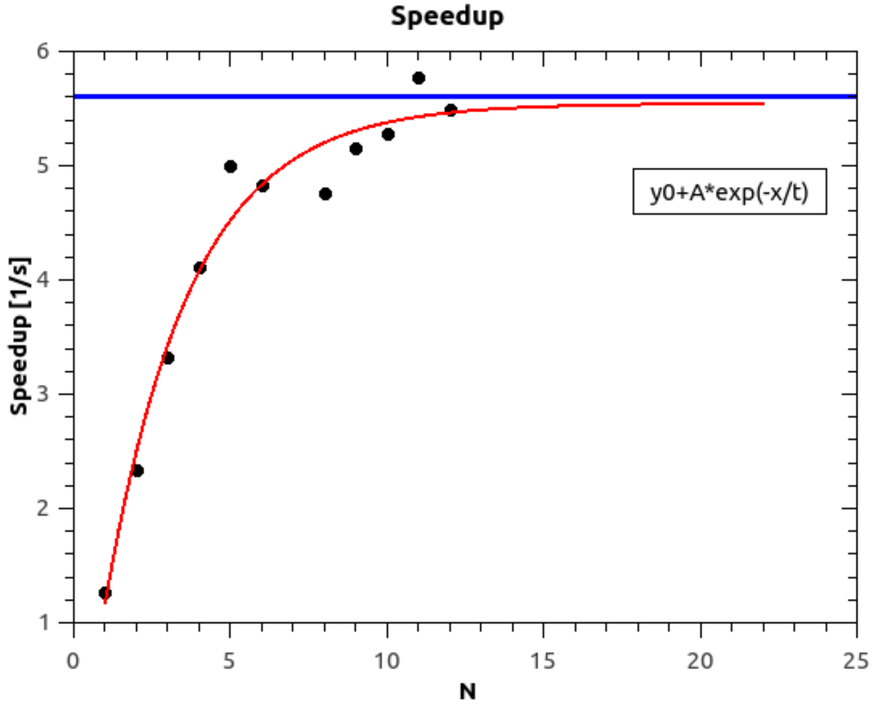
Total number of iterations 101

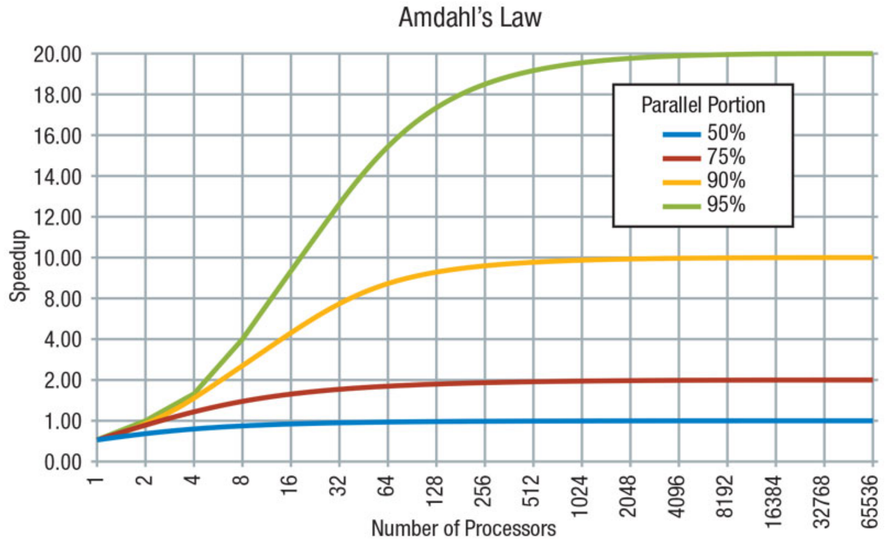
Elapsed wall clock time = 0.185227

HELMHOLTZ
Normal end of execution.
[dkucharski@zmisp-253 ~]$

Progress in parallel computation with Xeon Phi architecture







http://rtcmagazine.com/articles/view/103209

1. Amdahl, Gene M. (1967). "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities". AFIPS Conference Proceedings (30): 483–485. doi:10.1145/1465482.1465560.

2. Rodgers, David P. (June 1985). "Improvements in multiprocessor system design". ACM SIGARCH Computer Architecture News archive. New York, NY, USA: ACM. 13 (3): 225–231. doi:10.1145/327070.327215. ISBN 0-8186-0634-7. ISSN 0163-5964.

3. Dong Hyuk Woo and Hsien-Hsin S. Lee (December 2008). Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era, IEEE Computer, vol. 41, No. 12, pp.24-31,

4. Mark D. Hill and Michael R. Marty (April 2007). Amdahl's Law in the Multicore Era, IEEE Computer, vol. 41, pp. 33–38, July 2008. Also UW CS-TR-2007-1593,

Uruchominie kodu na GPU z poziomu Host

```
[host]$ cat hello_offload.c
#include <stdio.h>
#include <stdlib.h>
void
main()
{
    #pragma offload target (mic:0)
    {
        printf("hello_world from offloaded code running on the
coprocessor \n");
    }
}
```

[host]$ icc -offload hello_offload.c -o hello_offload

[host]$ ./hello_offload

hello_world from offloaded code running on the coprocessor

Dziękuję za uwagę