

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

Revision History

버전	날짜	작성자	변경내역
0.5	2011/6/30	이동훈, 문장완, 김슬기	초안 0.5
1.0b1	2011/7/11	문장완	1차 수정
1.0b2	2011/7/14	문장완	2차 수정
1.0b3	2011/7/25	문장완	3차 수정(MPI 추가)
1.0b4	2011/7/28	문장완	4차 수정(mapreduce 추가)
1.0b5	2011/7/29	문장완	5차 수정(hadoop 컴파일)
1.0b6	2011/7/29	PSC Group	정리

Document Status

유형	Draft (초안)	보안	Confidential
----	------------	----	--------------

OpenCL is a registered trademark of Apple Inc. used by permission by Khronos Group. All references to OpenCL components in this document are referenced from the publicly available OpenCL specification on the Khronos web-site at: <http://www.khronos.org/opencv>

NVIDIA, the NVIDIA logo, CUDA, and GeForce are trademarks or registered trademarks of NVIDIA Corporation.

Information furnished is believed to be accurate and reliable. However, PSC Group assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of PSC Group. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. PSC Group products are not authorized for use as critical components in life support devices or systems without express written approval of PSC Group.

The PSC Group logo is a registered trademark of PSC Group.
All other names are the property of their respective owners

© 2011 PSC Group - All rights reserved

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

내용

Introducing.....	3
GPGPU 개발환경 구축 : Windows 7 64-bit, CUDA 기준.....	3
GPGPU 개발환경 구축 : Linux 64-bit Ubuntu 11.04, CUDA 기준.....	17
GPGPU 개발환경 구축 : MPI 환경 설치.....	22
GPGPU 개발환경 구축 : Hadoop-mapreduce 환경 설치(Linux).....	30
GPGPU 개발환경 구축 : Hadoop 빌드 방법.....	38

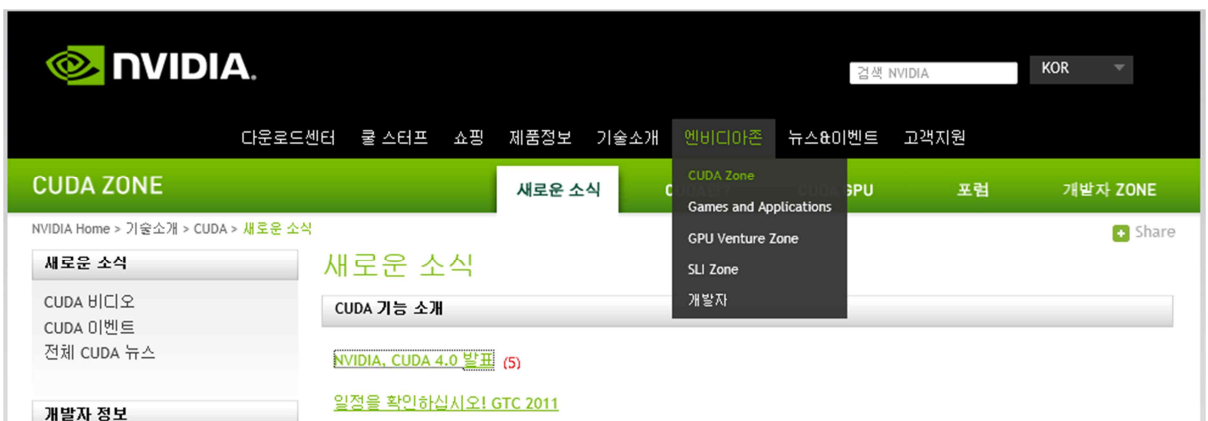
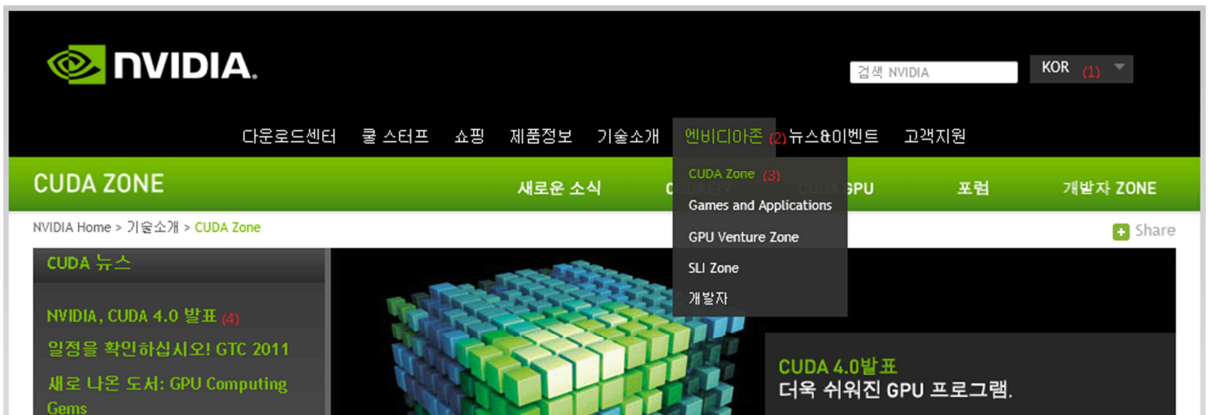
개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

Introducing

본 문서는 '공개소프트웨어 기반의 개인용 슈퍼 컴퓨팅 플랫폼 구축 및 커뮤니티 운영' 과제에 대한 개발환경 구축과 가이드에 대한 내용을 서술한 것임..

GPGPU 개발환경 구축 : Windows 7 64-bit, CUDA 기준

1. 우선 Microsoft Visual Studio 2008 또는/그리고 Microsoft Visual Studio 2010을 install 함.
 - A. 실제 설치 순서에 있어서, 나중에 Install 하게 될 Parallel Nsight 보다는 우선적으로 Visual Studio를 Install하면 문제는 없음.
2. <http://www.nvidia.com> 에서 아래 두 개의 그림과 같이 (1), (2), (3), (4), (5)의 순서대로 선택해 나감.



3. 영문으로 설명하는 페이지가 표시되는데 그 중에서 다음과 같은 표를 확인.

Windows 7, VISTA, Windows XP	Downloads
------------------------------	-----------

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

Developer Drivers for WinXP (270.81)	32-bit 64-bit
Developer Drivers for WinVista and Win7 (270.81)	32-bit 64-bit
Notebook Developer Drivers for WinVista and Win7 (270.61)	please check again later
CUDA Toolkit <ul style="list-style-type: none"> • C/C++ compiler • Visual Profiler • GPU-accelerated BLAS library • GPU-accelerated FFT library • GPU-accelerated Sparse Matrix library • GPU-accelerated RNG library • Additional tools and documentation 	32-bit 64-bit documentation
CUDA Tools SDK	32-bit 64-bit
GPU Computing SDK code samples	32-bit 64-bit
Parallel Nsight 2.0	download
Other Tools and Libraries	link to page

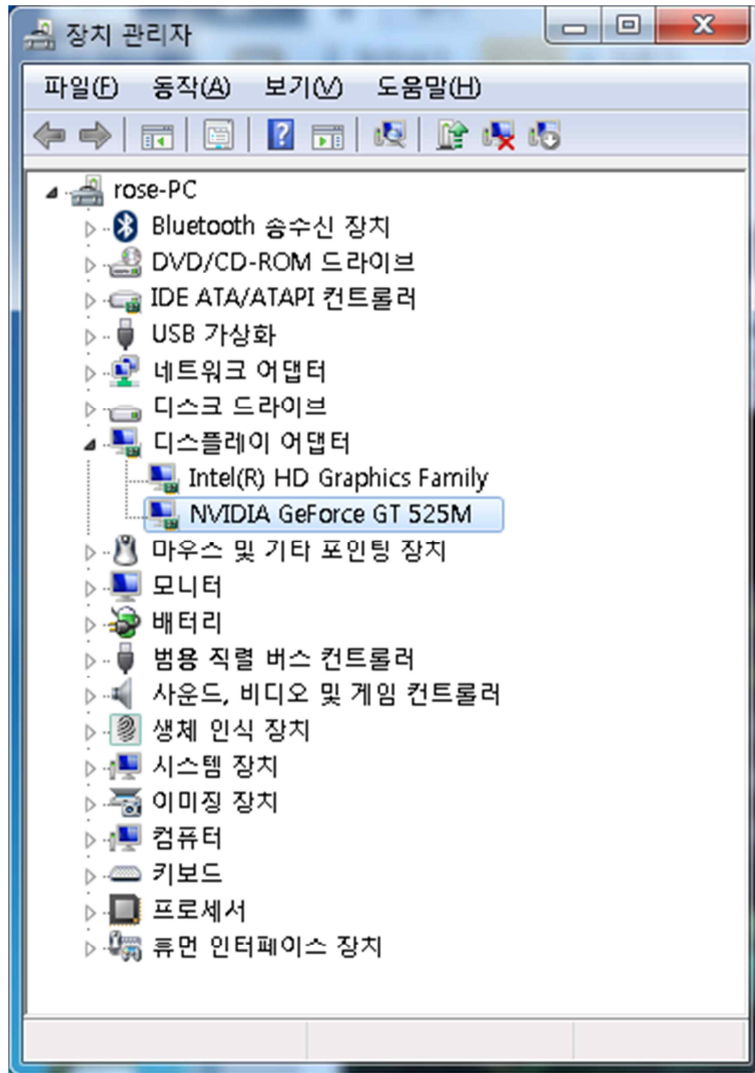
< Table 1 : WINDOWS, VISTA, WINDOWS XP Download >

4. Developer Driver를 설치하기 위해서는 위의 <Table 1>보다는 아래와 같이 다운로드

A. 그래픽 칩을 수동으로 확인하는 방법

- i. [컴퓨터]->[시스템 속성]->[장치 관리자]->[디스플레이 어댑터]에서 NVIDIA 그래픽 카드를 확인(여기에서는 NVIDIA GeForce GT 525M)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)



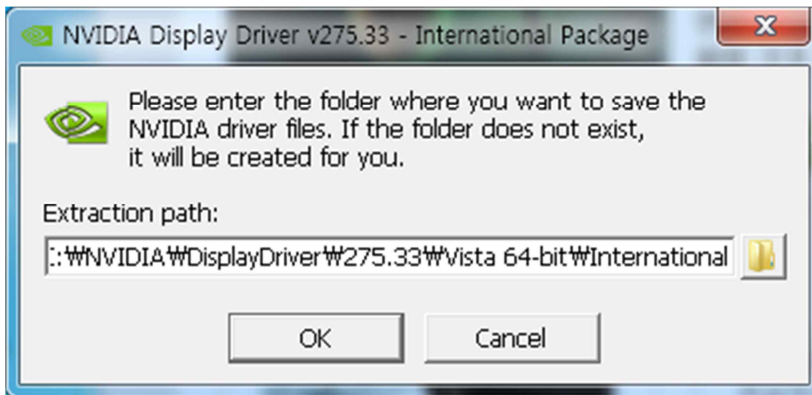
- ii. 아래 그림과 같이 (1), (2)를 선택 후 옵션 1과 같이 (3), (4), (5), (6), (7), (8)의 순서대로 또는 옵션 2와 같이 (a)의 자동 검색을 사용하여 드라이버 다운로드.

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)



B. 또는 (1), (2)를 선택 후 옵션 2와 같이 (a)의 자동 검색을 사용하여 드라이버 다운로드

5. 다운로드된 Developer Driver 설치



A. 'C:\NVIDIA\...'에 압축을 풀어서 인스톨을 시작하려고 하는데 웬만하면 이 Default 디렉토리에서 install을 완료함.

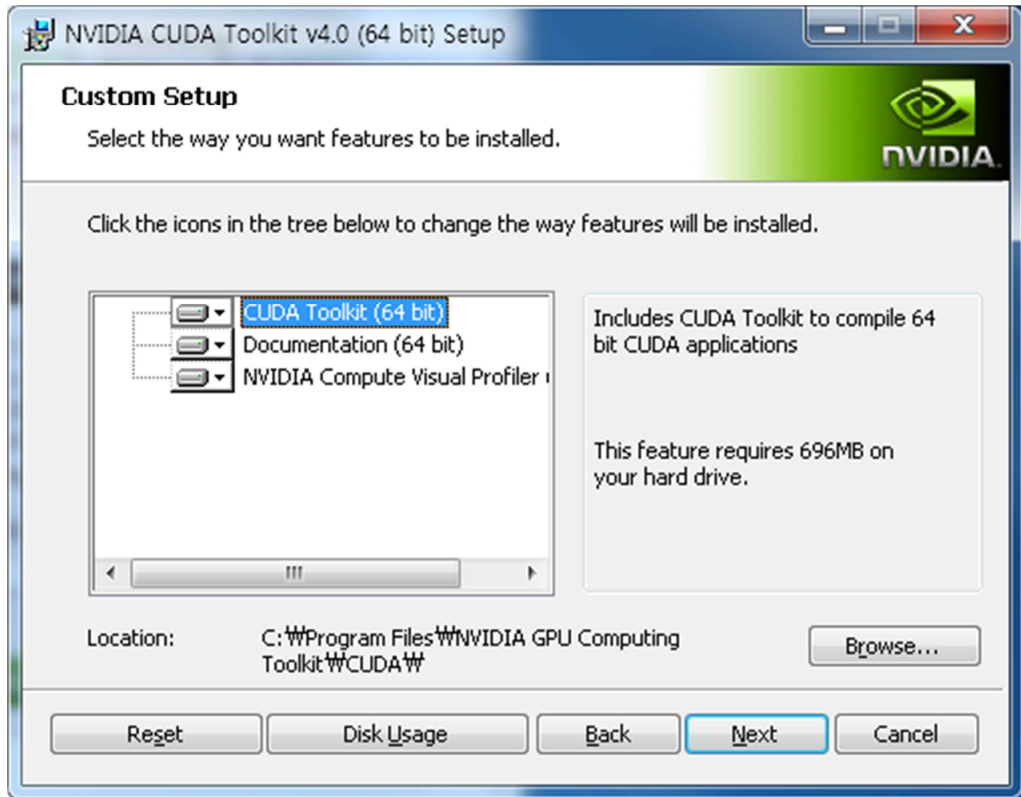
B. Install 완료와 동시에 컴퓨터 재시작을 요구함.

6. <Table 1>에서 CUDA Toolkit 중 OS에 맞게 32-bit 또는 64-bit를 다운로드 후 설치

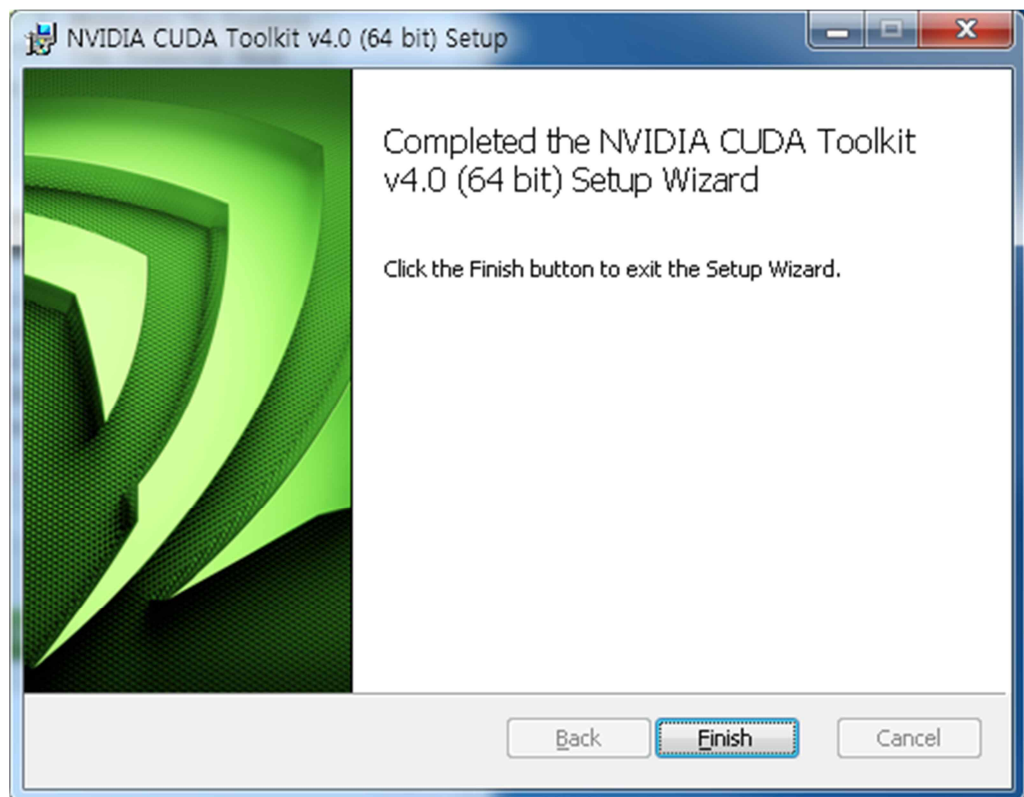
A. Default로 'C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA'에 설치됨.

B. 설치 초기 화면 (모두 설치)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)



C. 설치 완료 화면



D. 설치 완료 후 바탕화면에 생성된 아이콘

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)



7. <Table 1>에서 CUDA Toolkit 중 필요한 Document 다운로드
 - A. 아래 문서들은 모든 install이 완료된 후 필요에 따라 속독.
 - B. 제공되는 Document (NVIDIA GPU Computing Documentation - <http://developer.nvidia.com/nvidia-gpu-computing-documentation> 참조)

CUDA Getting Started Guide (Windows)

[Download](#)

This guide will show you how to install and check the correct operation of the CUDA development tools in Windows.

CUDA Getting Started Guide (Linux)

[Download](#)

This guide will show you how to install and check the correct operation of the CUDA development tools in Linux.

CUDA Getting Started Guide (Mac OS X)

[Download](#)

This guide will show you how to install and check the correct operation of the CUDA development tools in Mac OS X.

Getting Started with CUDA SDK samples

[Download](#)

This guide covers the introductory CUDA SDK samples beginning CUDA developers should review before developing your own projects.

SDK Code Sample Guide New Features in CUDA Toolkit 4.0

[Download](#)

This guide covers what is new in CUDA Toolkit 4.0 and the new code samples that are part of the CUDA SDK 4.0.

CUDA Toolkit 4.0 Release Notes

[Download](#)

NVIDIA CUDA Toolkit version 4.0 Release Notes for all OS Platforms

CUDA Toolkit 4.0 Release Notes Errata

[Download](#)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

NVIDIA CUDA Toolkit version 4.0 Release Notes Errata for all OS Platforms

CUDA Toolkit 4.0 Readiness for CUDA Applications

In NVIDIA CUDA Toolkit version 4.0, a major emphasis has been placed on improving the programmability of multi-threaded and multi-GPU applications and on improving the ease of porting existing code to CUDA C/C++. This document describes the key API changes and improvements that have been made toward that end, particularly where they have the potential to impact existing applications. This document also highlights a few of the improvements that have been made to the libraries bundled with the CUDA Toolkit.

[Download](#)

CUDA C Programming Guide

This is a detailed programming guide for CUDA C developers.

[Download](#)

CUDA C Best Practices Guide

This is a manual to help developers obtain the best performance from the NVIDIA CUDA Architecture. It presents established optimization techniques and explains coding metaphors and idioms that can greatly simplify programming for the CUDA architecture.

[Download](#)

CUDA Occupancy Calculator

The CUDA Occupancy Calculator allows you to compute the multiprocessor occupancy of a GPU by a given CUDA kernel. This tool provides guidance for optimizing the best kernel launch configuration for the best possible occupancy for the GPU.

[Download](#)

CUDA Developer Guide for Optimus Platforms

This document provides guidance to CUDA developers and explains how NVIDIA CUDA APIs can be used to query for GPU capabilities in Optimus systems. It is strongly recommended to follow these guidelines to ensure CUDA applications are compatible with all notebooks featuring Optimus.

[Download](#)

OpenCL Programming Guide

This is a detailed programming guide for OpenCL developers.

[Download](#)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

OpenCL Best Practices Guide

[Download](#)

This is a manual to help developers obtain the best performance from OpenCL.

OpenCL Overview for the CUDA Architecture

[Download](#)

This whitepaper summarizes the guidelines for how to choose the best implementations for NVIDIA GPUs.

OpenCL Implementation Notes

This document describes the "Implementation Defined" behavior for the NVIDIA OpenCL implementation as required by the OpenCL specification Version: 1.0. The implementation defined behavior is referenced below in the order of it's reference in the OpenCL specification and is grouped by the section number for the specification.

[Download](#)

DirectCompute Programming Guide

[Download](#)

This is a detailed programming guide for DirectCompute developers.

CUDA API Reference Manual (HTML)

[Download](#)

This is the CUDA Runtime and Driver API reference manual, an online HTML version

CUDA API Reference Manual (PDF)

[Download](#)

This is the CUDA Runtime and Driver API reference manual in PDF format.

CUDA API Reference Manual (CHM)

[Download](#)

This is the CUDA Runtime and Driver API reference manual in CHM format (Microsoft Compiled HTML help).

PTX: Parallel Thread Execution ISA Version 2.3

This document describes PTX, a low-level parallel thread execution virtual machine and instruction set architecture (ISA). PTX exposes the GPU as a data-parallel computing device.

[Download](#)

CUDA-memcheck User Manual

The CUDA debugger tool, cuda-gdb, includes a memory-checking feature for

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

detecting and debugging memory errors in CUDA applications. This document [Download](#) describes that feature and tool, called cuda-memcheck. The cuda-memcheck tool is designed to detect such memory access errors in your CUDA application.

CUDA-gdb Debugger User Manual

CUDA-GDB is the NVIDIA tool for debugging CUDA applications running on Linux and Mac. The tool provides developers with a mechanism for [Download](#) debugging CUDA applications running on actual hardware. CUDA-GDB runs on Linux and Mac OS X, 32-bit and 64-bit. The Linux edition is based on GDB 6.6 whereas the Mac edition is based on GDB 6.3.5

Compute Visual Profiler

The Compute Visual Profiler is a graphical user interface based profiling tool that can be used to measure performance and find potential opportunities for [Download](#) CUDA and OpenCL optimizations, to achieve maximum performance from NVIDIA GPUs. Compute Visual Profiler provides metrics in the form of plots and counter values presented in tables and as graphs. It tracks events with hardware counters on signals in the chip; this is explained in detail in the chapter entitled, "Compute Visual Profiler Counters."

CUDA Fermi Compatibility Guide

The Fermi Compatibility Guide for CUDA Applications is intended to help developers ensure that their NVIDIA CUDA applications will run effectively on GPUs based on the NVIDIA Fermi Architecture. This document provides [Download](#) guidance to developers who are already familiar with programming in CUDA C/C++ and want to make sure that their software applications are compatible with Fermi.

CUDA Fermi Tuning Guide

An overview on how to tune applications for Fermi to further increase these [Download](#) speedups is provided. More details are available in the CUDA C Programming Guide (version 3.2 and later) as noted throughout the document..

CUBLAS Library User Guide

[Download](#)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

The CUBLAS library is an implementation of BLAS (Basic Linear Algebra Subprograms) on top of the NVIDIA CUDA runtime. It allows the user to access the computational resources of NVIDIA Graphical Processing Unit (GPU), but does not auto-parallelize across multiple GPUs.

CUFFT Library User Guide

This document describes CUFFT, the NVIDIA CUDA Fast Fourier Transform (FFT) library. The FFT is a divide-and-conquer algorithm for efficiently computing discrete Fourier transforms of complex or real-valued data sets, and it is one of the most important and widely used numerical algorithms, with applications that include computational physics and general signal processing. The CUFFT library provides a simple interface for computing parallel FFTs on an NVIDIA GPU, which allows users to leverage the floating-point power and parallelism of the GPU without having to develop a custom, GPUbased FFT implementation.

[Download](#)

CUSPARSE Library User Guide

The NVIDIA CUDA CUSPARSE library contains a set of basic linear algebra subroutines used for handling sparse matrices and is designed to be called from C or C++. These subroutines can be classified in four categories.

[Download](#)

CURAND Library User Guide

The NVIDIA CURAND library provides facilities that focus on the simple and efficient generation of high-quality pseudorandom and quasirandom numbers.

[Download](#)

NVIDIA Performance Primitives (NPP) Library User Guide

NVIDIA NPP is a library of functions for performing CUDA accelerated processing. The initial set of functionality in the library focuses on imaging and video processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

[Download](#)

Thrust Quick Start Guide

Thrust is a C++ template library for CUDA based on the Standard Template

[Download](#)

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

Library (STL). Thrust allows you to implement high performance parallel applications with minimal programming effort through a high-level interface that is fully interoperable with CUDA C.

NVIDIA CUDA H.264 Video Encoder Library User Guide

The NVIDIA CUDA H.264 Video Encoder is a library for performing CUDA accelerated video encoding. The functionality in the library takes raw YUV frames as input and generates NAL packets. This encoder supports up to various profiles up to High Profile @ Level 4.1.

[Download](#)

NVIDIA CUDA Video Decoder Library User Guide

The CUDA Video Decoder API gives developers access to hardware video decoding capabilities on NVIDIA GPU. The actual hardware decode can run on either Video Processor (VP) or CUDA hardware, depending on the hardware capabilities and the codecs. This API supports the following video stream formats for Linux and Windows platforms: MPEG-2, VC-1, and H.264 (AVCHD).

[Download](#)

CUDA C SDK Release Notes

[Download](#)

DirectCompute SDK Release Notes

[Download](#)

OpenCL SDK Release Notes

[Download](#)

CUDA Toolkit Software License Agreement

This is the Software License Agreement for developers that use the CUDA Toolkit. This License agreement also include the distribution license for CUDA Accelerated Libraries.

[Download](#)

GPU Computing SDK End User License Agreement

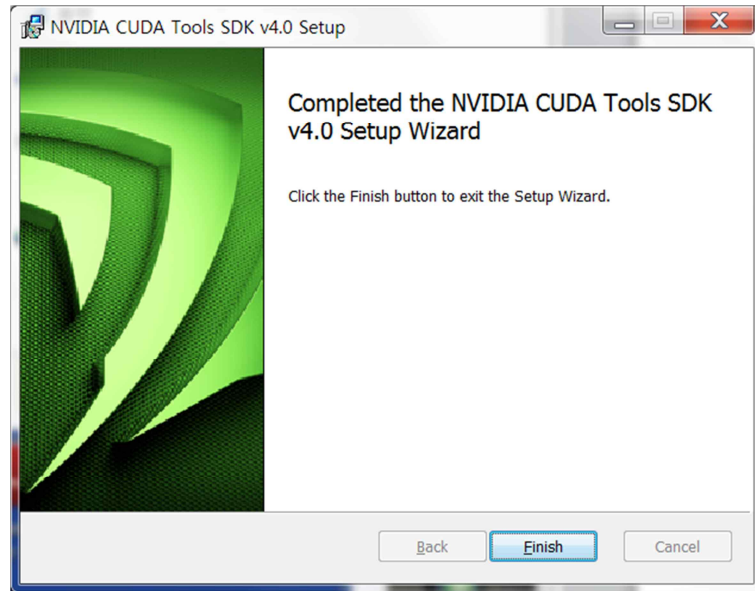
[Download](#)

This is the Software License Agreement for developers or licensees.

8. <Table 1>에서 CUDA Tools SDK 중 OS에 맞게 32-bit 또는 64-bit를 다운로드 후 설치
 - A. 'Choose Setup Type'에서 'Complete install' 실행
 - B. Default로 'C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA Tools SDK'에 install 됨.

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

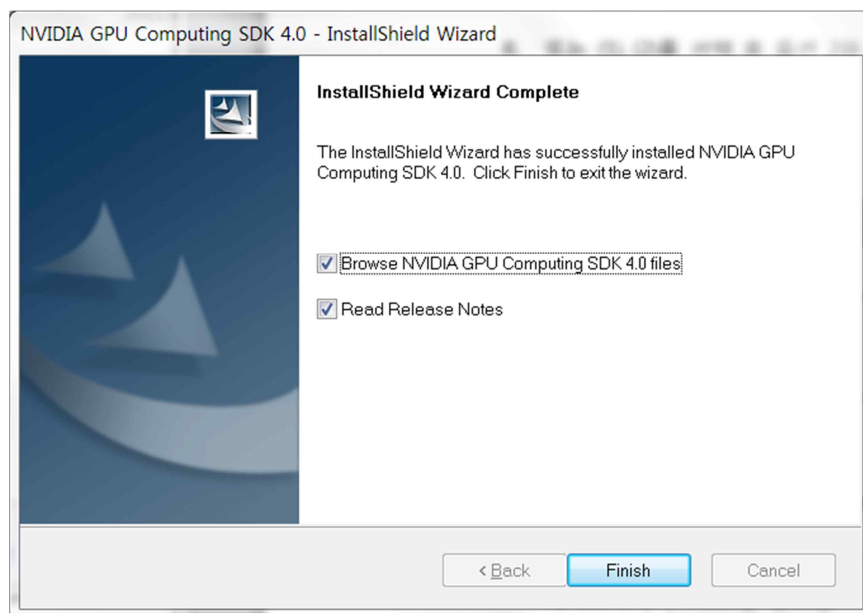
C. 다음과 같은 화면과 함께 install 완료됨



9. <Table 1>에서 'GPU Computing SDK code samples' 중 OS에 맞게 32-bit 또는 64-bit를 다운로드 후 설치

A. Default로 'C:\ProgramData\NVIDIA Corporation\NVIDIA GPU Computing SDK 4.0'에 install 됨.

B. 다음과 같은 화면과 함께 완료 됨.



10. <Table 1>에서 'Parallel Nsight 2.0'를 Download

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- A. 등록 또는 Login 후 OS에 맞게 32-bit 또는 64-bit 버전 다운로드
- B. 관련 Document와 참고 자료 (<http://developer.nvidia.com/nvidia-parallel-nsight>)

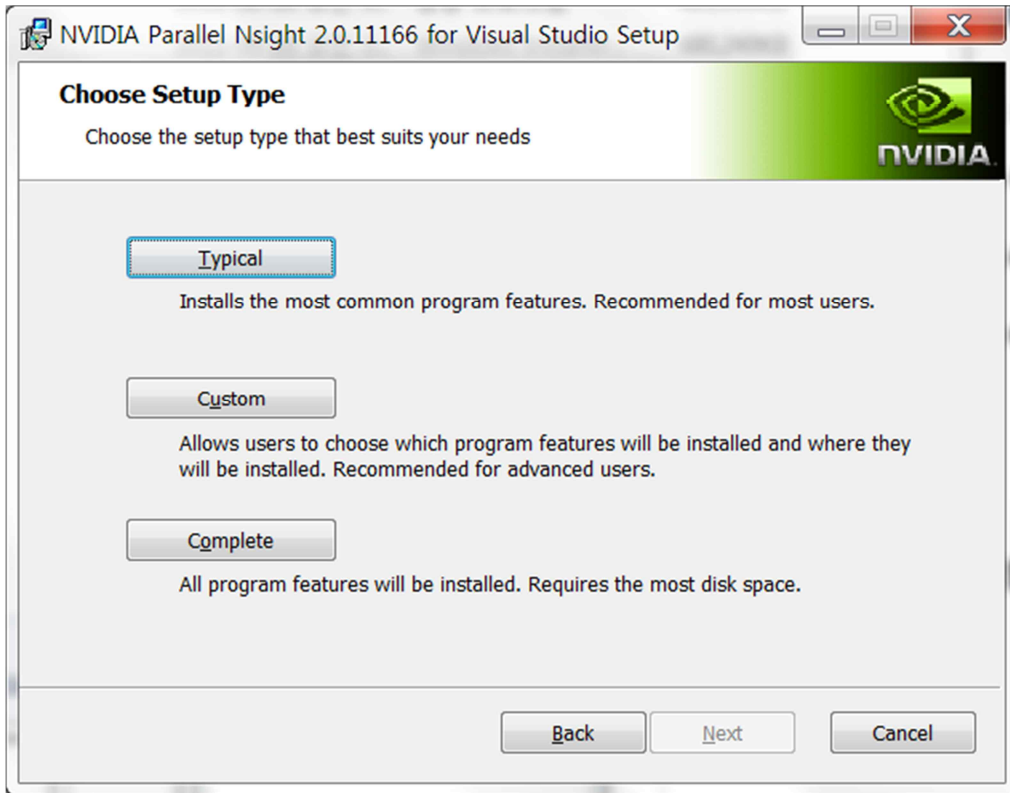
NVIDIA Parallel Nsight

NVIDIA Parallel Nsight brings GPU Computing into Microsoft Visual Studio. Debug, profile and analyze GPGPU or graphics applications using CUDA C, OpenCL, DirectCompute, Direct3D, and OpenGL.

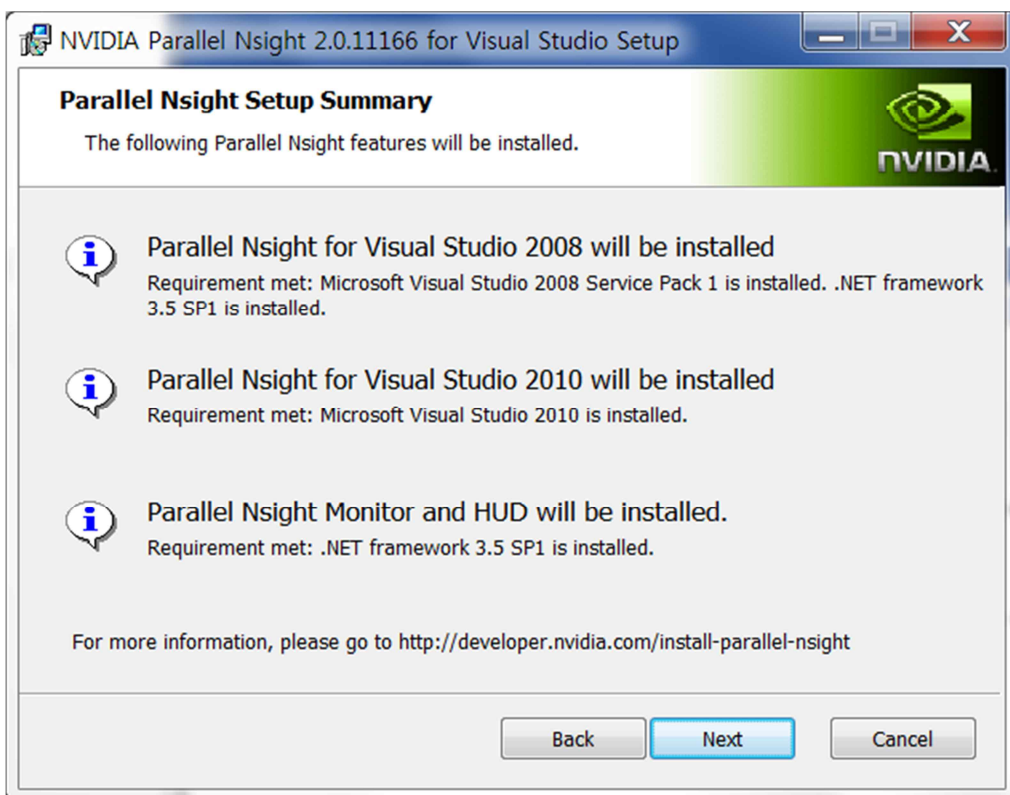
- [Product Overview](#)
Introduction to Parallel Nsight on NVIDIA.com
- [Support and Documentation](#)
User Guide, Forums, and more
- [Videos](#)
Teaser and Instructional Videos showing Parallel Nsight in action
- [Webinars](#)
Past Parallel Nsight Webinars
- [Licensing](#)
License and Pricing information

- C. 'Choose Setup Type'에서 'complete' 선택

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)



- D. 'Parallel Nsight Setup Summary' 다이얼로그 박스에서 아래와 같이 이미 install 되어 있는 MS Visual Studio 버전을 찾아서 진행 함.

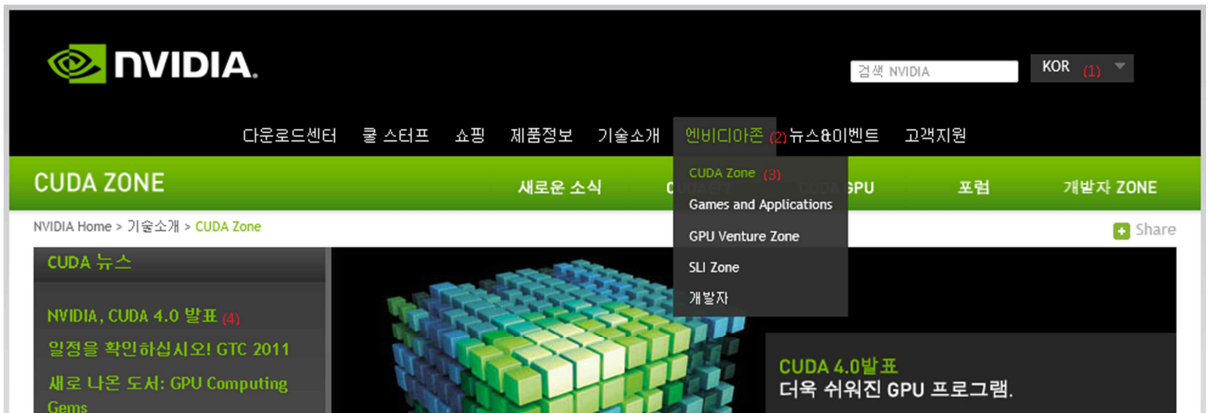


개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

11. <Table 1>에서 'Other Tools and Libraries' link로 가서 필요한 툴 Download

GPGPU 개발환경 구축 : Linux 64-bit Ubuntu 11.04, CUDA 기준

- 1 우선 Ubuntu 를 인스톨 함(본 문서에서는 version 11.04, 64-bit)
- 2 <http://www.nvidia.com> 에서 아래 두 개의 그림과 같이 (1), (2), (3), (4), (5)의 순서대로 선택해 나감.



- 3 위 Windows 7 과 같은 방식으로 Nvidia 웹 메뉴를 선택
- 4 영문으로 설명하는 페이지가 표시되는데 그 중에서 다음과 같은 표를 확인.

Linux	Downloads
Developer Drivers for Linux (270.41.19)	32-bit 64-bit
CUDA Toolkit	documentation

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

Linux	Downloads
<ul style="list-style-type: none"> • C/C++ compiler • CUDA-GDB debugger • Visual Profiler • GPU-accelerated BLAS library • GPU-accelerated FFT library • GPU-accelerated Sparse Matrix library • GPU-accelerated RNG library • Additional tools and documentation 	
CUDA Toolkit for Fedora 13	32-bit 64-bit
CUDA Toolkit for RedHat Enterprise Linux 6.0	64-bit
CUDA Toolkit for RedHat Enterprise Linux 5.5	32-bit 64-bit
CUDA Toolkit for RedHat Enterprise Linux 4.8	64-bit
CUDA Toolkit for Ubuntu Linux 10.10	32-bit 64-bit
CUDA Toolkit for OpenSUSE 11.2	32-bit 64-bit
CUDA Toolkit for SUSE Linux Enterprise Server 11 SP1	32-bit 64-bit
CUDA Tools SDK	32-bit 64-bit
GPU Computing SDK code samples	download
Other Tools and Libraries	link to page

< Table 3 : Linux Download >

1 위 <Table 3>의 'Developer Drivers for Linux (270.41.19)'을 download

2 설치 전에 혹시 설치되어 CUDA 가 설치 되어있다면 지워주어야 함

A. apt-get -purge remove nvidia-*

3 'Developer Drivers for Linux (270.41.19)' 를 설치

A. 참조문서

i. CUDA Getting Started Guide (Linux)

http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Getting_Started_Linux.pdf

ii. NVIDIA Accelerated Linux Graphics Driver README and Installation Guide

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

http://us.download.nvidia.com/XFree86/Linux-x86/256.35/README_index.html

- B. GUI 환경에서 빠져 나가야 함
- i. 'Terminal' 프로그램을 시작한 후 'sudo /etc/init.d/gdm stop' 또는 '/sbin/init 3' 실행해서 GUI 를 빠져나감.
 - ii. 또는 재부팅 후 recovery 에서 netroot 로 부팅
- C. Download 한 Driver 프로그램의 폴더를 확인한 후, 'sudo sh <directory>/devdriver_4.0_linux_64_270.41.19.run' 를 실행(Driver file name 은 버전에 따라 변경될 수 있음)
- D. 맞는 버전의 Driver 가 설치되었는지 'cat /proc/driver/nvidia/version' 명령어로 확인

예) 다음과 같이 표시됨.

```
NVRM version: NVIDIA UNIX x86-64 Kernel Module 270.41.06 Mon Apr 18
14:53:56 PDT 2011
```

```
GCC version: gcc version 4.5.2 (Ubuntu/Linaro 4.5.2-8ubuntu4)
```

- E. CUDA 가 돌아갈 수 있도록 환경변수를 지정해 주어야 함

```
sudo nano ~/.bashrc    또는    sudo nano ~/bash_profile 에
```

맨 아랫줄에 다음 명령어를 적음

```
export CUDA_HOME="/usr/local/cuda"
```

```
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${CUDA_HOME}/lib64"
```

//32 비트 이신 분들은 64 를 지워주시면 됩니다.

```
export PATH=${CUDA_HOME}/bin:${PATH}
```

그 후 적용

```
source ~/.bashrc    또는    source ~/bash_profile
```

4 CUDA Toolkit 을 설치

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

A. `sudo sh cudatoolkit_4.0.17_linux_64_ubuntu10.10.run`

5 CUDA Tools SDK 를 설치

A. `sudo sh cudatools_4.0.17_linux_64.run`

6 GPU Computing SDK code samples 를 설치

A. `sudo sh cudatoolkit_4.0.17_linux_64_ubuntu10.10.run`

7 현재(2011-07-11) 아직 CUDA 가 GCC 4.5 버전 이상을 지원하지 않으므로 GCC 4.4 버전을 받음

A. `sudo apt-get install build-essential gcc-4.4 g++-4.4 libxi-dev libxmu-dev freeglut3-dev`

B. 새로 폴더를 만들고 설치한 파일을 넣는다.

i. `mkdir gcc`

ii. `cd gcc`

iii. `ln -s /usr/bin/cpp-4.4 cpp`

iv. `ln -s /usr/bin/gcc-4.4 gcc`

v. `ln -s /usr/bin/g++-4.4 g++`

C. nvcc 에 등록

i. `nano /usr/local/cuda/bin/nvcc.profile`

ii. `compiler-bindir = /home/xxx/gcc //gcc 폴더가 있는 곳을 작성`

D. GUI 환경으로 돌아가기 위해 'startx' or 'init 5' or 'sudo /etc/init.d/gdm start' 등 시스템 환경에 맞는 명령 실행

8 컴파일을 하고 실행 확인

A. Sample SDK 가 설치된 폴더 -> C 폴더 -> `sudo make` 를 하시면 컴파일이 됨

B. 컴파일이 완료된 후 `bin -> linux -> release` 에서 `./deviceQuery` 하면 됨

9 기타 참고사항

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

A. SLI 를 인식하지 못하는 현상

- i. `sudo nvidia-xconfig --enable-all-gpus`
- ii. `sudo nvidia-xconfig --sl=On`
- iii. `sudo pico /etc/default/grub` 여기서
'GRUB_CMDLINE_LINUX_DEFAULT="quiet splash vmalloc=256M"' 라고 쳐주면 됨.
혹시 256M 으로 해서 안되면 192M 으로 바꾸면 됨.
- iv. `sudo update-grub`

B. 드라이버를 잘못 설치하여 부팅이 되는 않는 현상

- i. 전에 내용을 삭제 후 재설치
- ii. 개발자 드라이버를 설치했는데 이렇게 되면 자신의 그래픽 카드에 맞는 드라이버를 찾아 설치해도 됨

C. 컴파일 중 에러

- i. `can not found -lcuda`
 - ① cuda 라이브러리를 못 찾는 경우라 드라이버를 다시 설치

D. 실행 시 에러

- i. `Libcudart.so.4`
 - ① PATH 설정이 잘못 되었을 경우가 크므로 자신의 컴퓨터의 환경에 맞게 해주어야 함.

10 참고 문헌

- A. <http://ubuntuforums.org/archive/index.php/t-1741962.html>
- B. <http://hdfpga.blogspot.com/2011/05/install-cuda-40-on-ubuntu-1104.html>

GPGPU 개발환경 구축 : MPI 환경 설치

1 Home Page

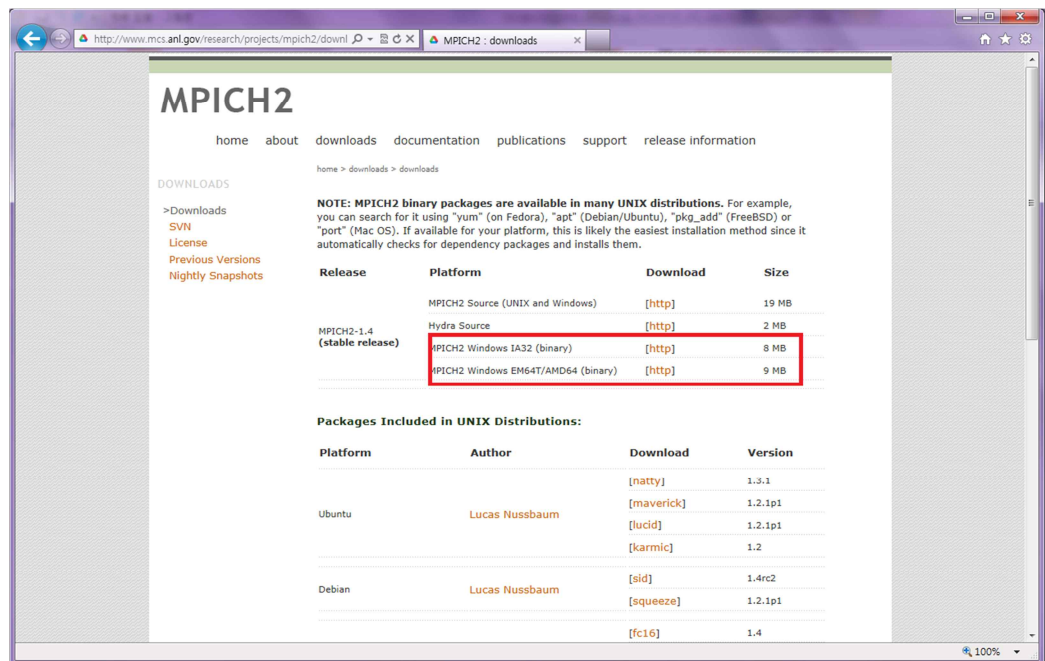
A. MPICH2 : <http://www.mcs.anl.gov/research/projects/mpich2/downloads/index.php?s=downloads>

B. Microsoft HPC SDK Pack : <http://www.microsoft.com/download/en/details.aspx?id=10505>

2 설치 및 환경 구성 방법

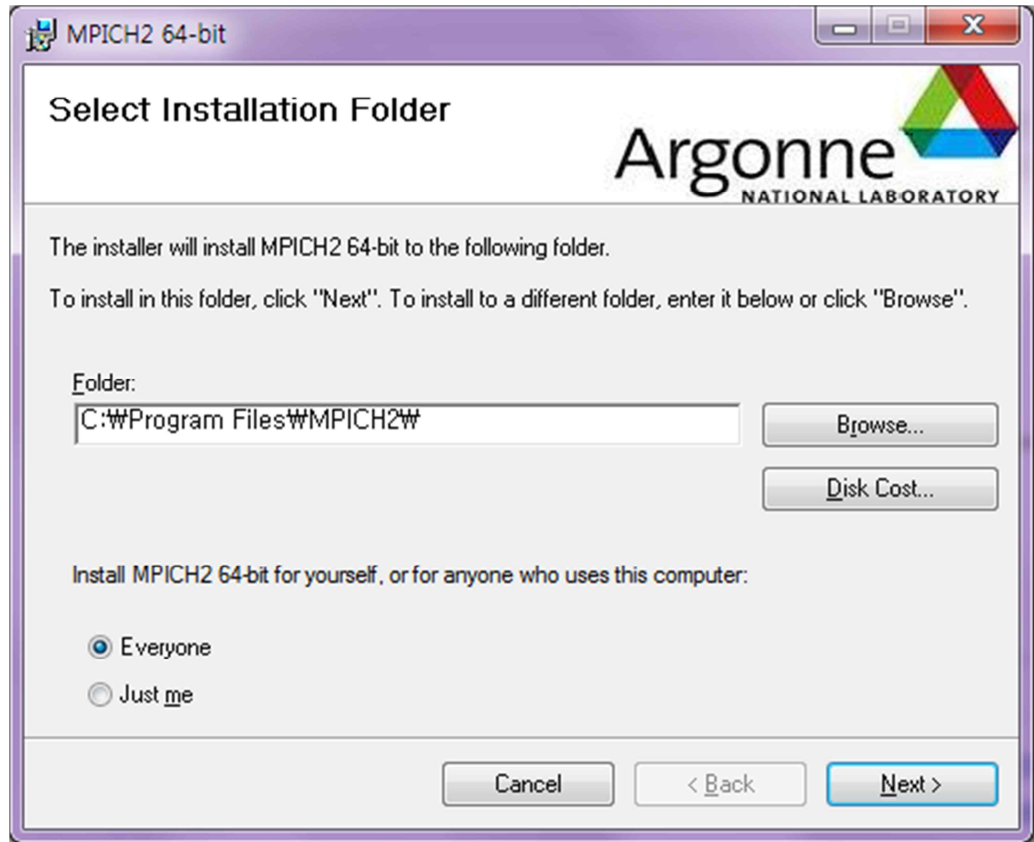
A. MPICH2

i. 해당 사이트에서 자신의 컴퓨터에 맞는 프로그램을 다운을 받는다.



개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

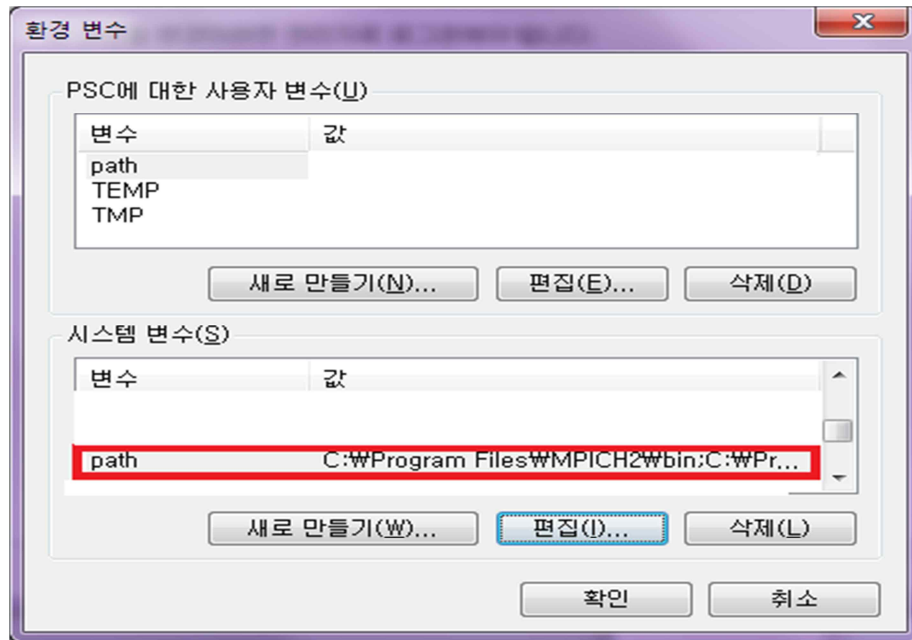
ii. 다운 받은 파일을 실행 및 설치



- ① 설치폴더는 기본으로 하되 그림과 같이 Everyone 으로 하는 것이 좋음

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- iii. 설치를 마친 뒤에는 환경변수 path 에 C:\Program Files\WMPICH2\Wbin 를 추가



- iv. 환경변수 등록을 마쳤으면 프로그램이 돌아가는지 확인

- ① 실행창에서 cmd 를 실행시킨 뒤에 mpiexec 를 타이핑

```
C:\Windows\system32\cmd.exe
C:\Users\WPSC>mpiexec

Usage:
mpiexec -n <maxprocs> [options] executable [args ...]
mpiexec [optional] executable [args ...] : [options] exe [args] : ...
mpiexec -configfile <configfile>

options:
standard:
-n <maxprocs>
-wdir <working directory>
-configfile <filename> -
    each line contains a complete set of mpiexec options
    including the executable and arguments
-host <hostname>
-path <search path for executable, ; separated>

extensions:
-env <variable value>
-hosts <n host1 host2 ... hostn>
-hosts <n host1 m1 host2 m2 ... hostn mn>
-machinefile <filename> - one host per line, #commented
-locallonly <numprocs>
-exitcodes - print the exit codes of processes as they exit
-genvlist <list of env var names a,b,c,...> - pass current values of these vars
-g<local arg name> - global version of local options
    genv, gudir, ghost, gpath, gmap
-file <filename> - old mpich1 job configuration file

examples:
mpiexec -n 4 cpi
mpiexec -n 1 -host foo master : -n 8 worker

For a list of all mpiexec options, execute 'mpiexec -help2'

C:\Users\WPSC>
```


개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

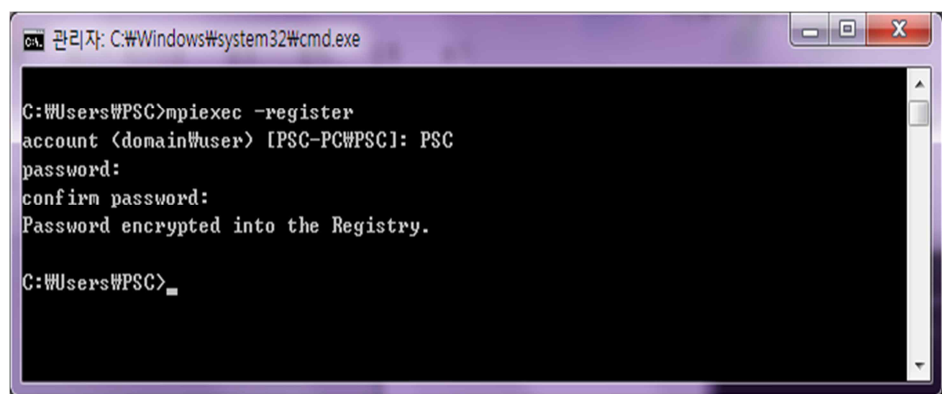
② 이러한 내용이 나왔다면 성공

v. 그 후 계정을 등록

① 이어서 `mpiexec -register` 를 타이핑

② 계정은 지금 현재 로그인 되어있는 윈도우 계정을 등록하는데 비밀번호 역시 같게 함.

③ 비밀번호가 없을 시에는 등록해야 함

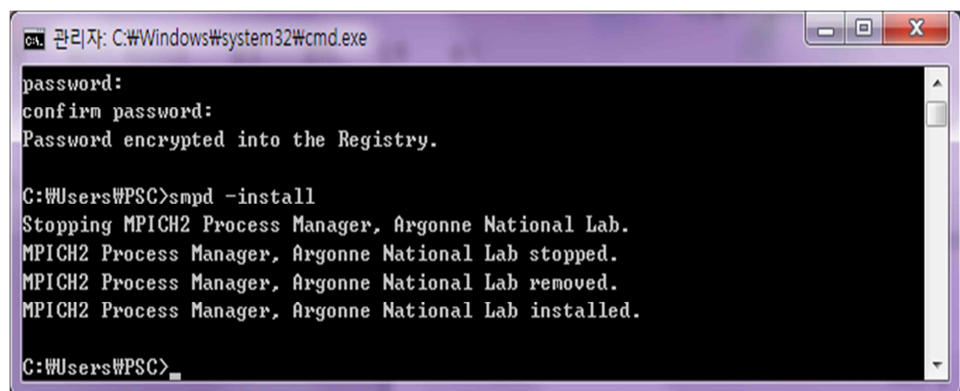


```
C:\Users\WPC>mpiexec -register
account (domain\User) [PSC-PC\WPC]: PSC
password:
confirm password:
Password encrypted into the Registry.

C:\Users\WPC>
```

vi. 등록을 마치면 `smpd` 를 설치하고 작동상태를 확인

① 명령창에 `smpd -install` 이라고 치면 설치가 됨



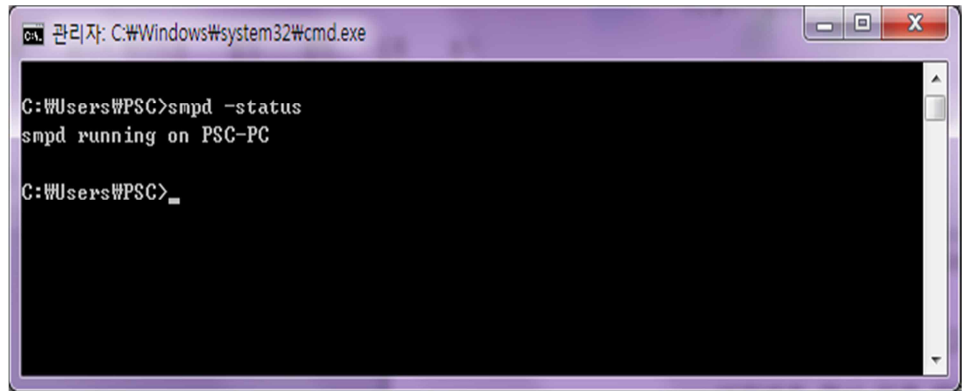
```
password:
confirm password:
Password encrypted into the Registry.

C:\Users\WPC>smpd -install
Stopping MPICH2 Process Manager, Argonne National Lab.
MPICH2 Process Manager, Argonne National Lab stopped.
MPICH2 Process Manager, Argonne National Lab removed.
MPICH2 Process Manager, Argonne National Lab installed.

C:\Users\WPC>
```

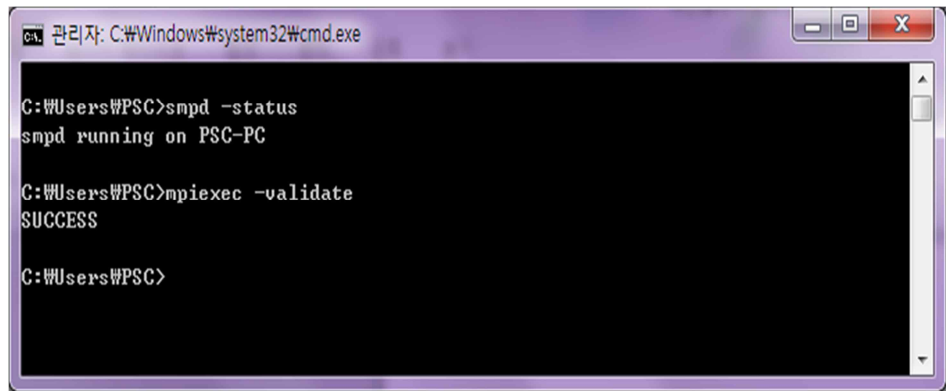
개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- ② 그 후 `smpd -status` 를 쳐서 작동이 되는지 확인



```
관리자: C:\Windows\system32\cmd.exe
C:\Users\PSC>smpd -status
smpd running on PSC-PC
C:\Users\PSC>
```

- ③ 그 다음 `mpiexec -validate` 를 쳐서 mpi 가 작동되는지 확인



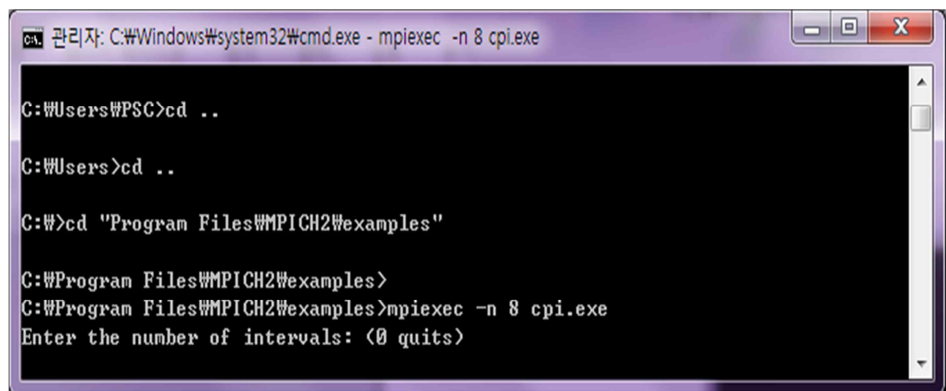
```
관리자: C:\Windows\system32\cmd.exe
C:\Users\PSC>smpd -status
smpd running on PSC-PC
C:\Users\PSC>mpiexec -validate
SUCCESS
C:\Users\PSC>
```

vii. 작동이 정상적인지 확인

- ① Example 이 있는 폴더로 이동해서 작동

`mpiexec -n x cpi.exe`

(x : CPU 개수)



```
관리자: C:\Windows\system32\cmd.exe - mpiexec -n 8 cpi.exe
C:\Users\PSC>cd ..
C:\Users>cd ..
C:\>cd "Program Files\MPICH2\examples"
C:\Program Files\MPICH2\examples>
C:\Program Files\MPICH2\examples>mpiexec -n 8 cpi.exe
Enter the number of intervals: <0 quits>
```

viii. 다른 컴퓨터와 연결 하여 MPI 를 작동시킬 때

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

① 같은 WORKGRUOP 에 있어야 할 것

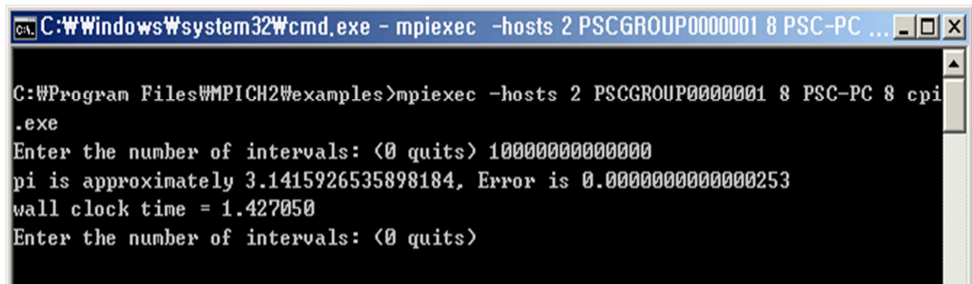
② 같은 계정과 비밀번호를 써야 할 것

③ 그 후 다음과 같은 코드를 씬.

```
mpiexec -hosts n xxx m yyy l cpi.exe
```

(n : 컴퓨터 개수 xxx, yyy :컴퓨터 명 m, l : CPU 개수)

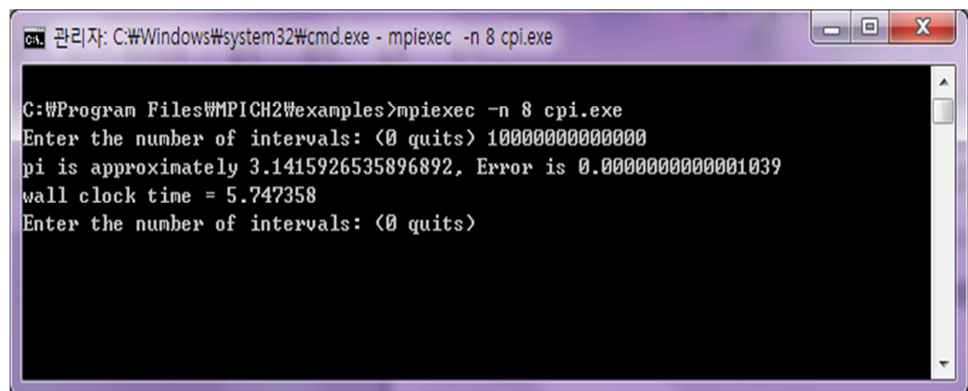
④ 속도 비교 => 1.427050 : 5.747358 => 약 4 배 이상 성능 향상



```
C:\Windows\system32\cmd.exe - mpiexec -hosts 2 PSCGROUP0000001 8 PSC-PC ...
C:\Program Files\MPICH2\examples>mpiexec -hosts 2 PSCGROUP0000001 8 PSC-PC 8 cpi.exe
Enter the number of intervals: (<0 quits>) 10000000000000
pi is approximately 3.1415926535898184, Error is 0.000000000000253
wall clock time = 1.427050
Enter the number of intervals: (<0 quits>)
```

< Intel I7 2630QN(Quad cores x 2 hyper-threading)

+ Intel Xeon E5620(Quad cores x 2 hyper-threading)>



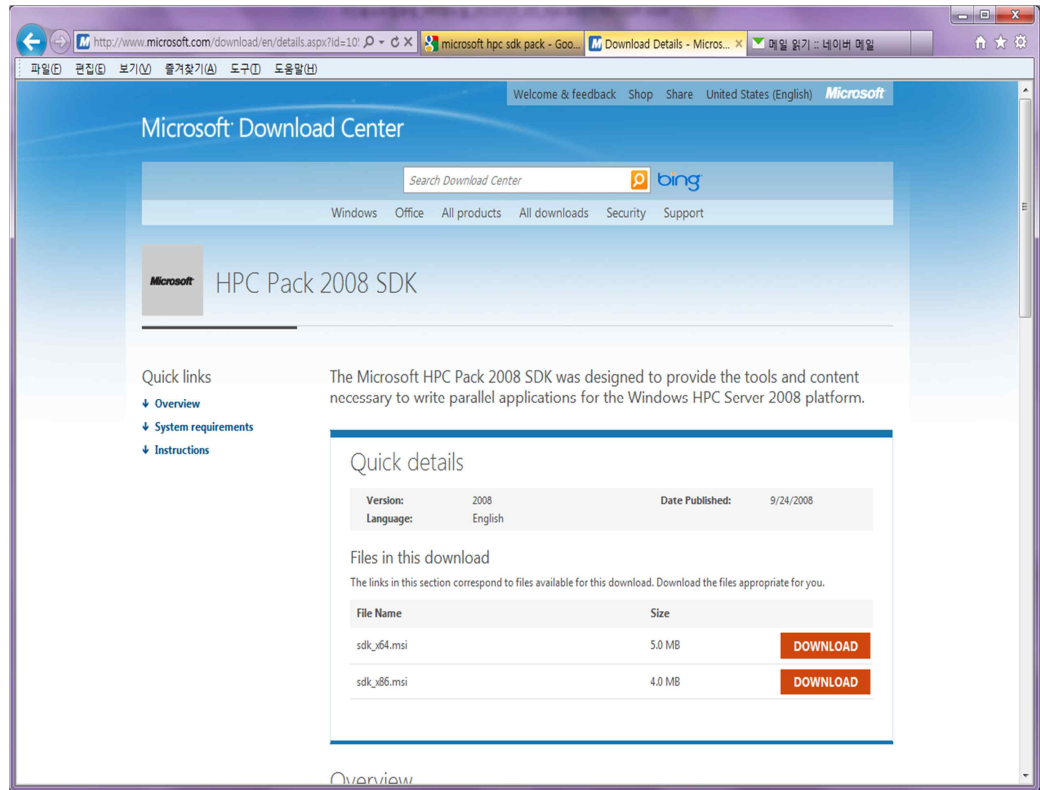
```
관리자: C:\Windows\system32\cmd.exe - mpiexec -n 8 cpi.exe
C:\Program Files\MPICH2\examples>mpiexec -n 8 cpi.exe
Enter the number of intervals: (<0 quits>) 10000000000000
pi is approximately 3.1415926535896892, Error is 0.0000000000001039
wall clock time = 5.747358
Enter the number of intervals: (<0 quits>)
```

<Intel I7 2630QN(Quad cores x 2 hyper-threading)>

B. Microsoft HPC SDK Pack

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- i. 자신의 윈도우 버전에 맞게 프로그램을 다운로드 받는다

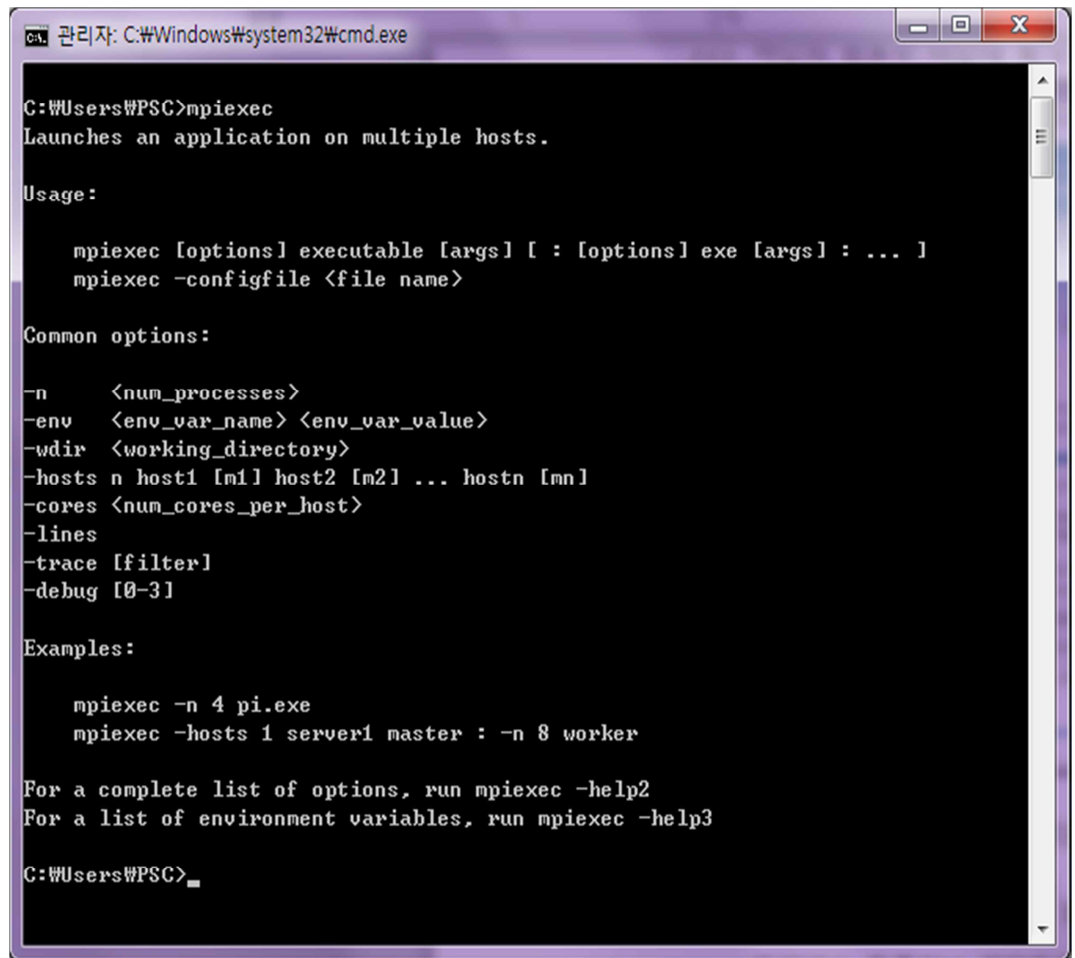


- ii. 다운로드 받은 파일을 설치를 한다

- iii. HPC Pack 은 MPICH2 와 다르게 자동으로 셋팅을 다 해줘서 따로 환경변수 등록이나 계정등록을 할 필요가 없다

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

iv. 명령창에서 작동이 잘 되었는지 확인



```
C:\> 관리자: C:\Windows\system32\cmd.exe

C:\Users\WPSC>mpiexec
Launches an application on multiple hosts.

Usage:

    mpiexec [options] executable [args] [ : [options] exe [args] : ... ]
    mpiexec -configfile <file name>

Common options:

-n      <num_processes>
-env    <env_var_name> <env_var_value>
-wdir   <working_directory>
-hosts  n host1 [m1] host2 [m2] ... hostn [mn]
-cores  <num_cores_per_host>
-lines
-trace  [filter]
-debug  [0-3]

Examples:

    mpiexec -n 4 pi.exe
    mpiexec -hosts 1 server1 master : -n 8 worker

For a complete list of options, run mpiexec -help2
For a list of environment variables, run mpiexec -help3

C:\Users\WPSC>
```

v. 이런식으로 나오면 작동이 잘 되는 것인데, MPICH2 와 별로 차이가 보이지 않아서 혼동될 경우 아랫줄에 help 를 보면 됨. 앞에서 MPICH2 는 help3 이 존재하지 않음

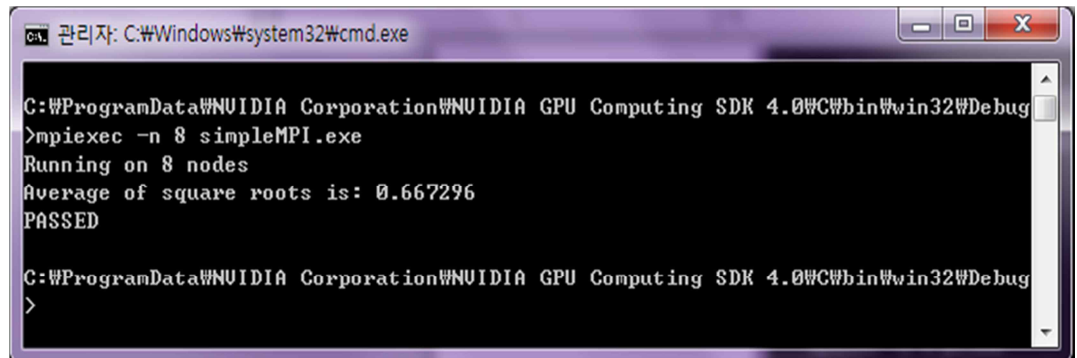
vi. 여기서 테스트 할 샘플은 CUDA 에 있는 simpleMPI 임. 그래서 앞의 문서를 보고 CUDA 를 설치해야 함.

vii. 설치를 마치고 cuda 예제가 있는 폴더로 이동한 뒤 작동 시도

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

viii. 프로그램 실행은 MPICH2 와 같음

```
mpiexec -n x simpleMPI.exe
```



```
C:\ProgramData\NVIDIA Corporation\NVIDIA GPU Computing SDK 4.0\bin\win32\Debug
>mpiexec -n 8 simpleMPI.exe
Running on 8 nodes
Average of square roots is: 0.667296
PASSED
C:\ProgramData\NVIDIA Corporation\NVIDIA GPU Computing SDK 4.0\bin\win32\Debug
>
```

GPGPU 개발환경 구축 : Hadoop-mapreduce 환경 설치(Linux)

1 Home Page

A. Hadoop : <http://hadoop.apache.org/>

B. Java : <http://www.java.com/ko/>

2 Process

A. Hadoop 을 다운받음.

i. 첫 홈페이지에서 Common 페이지에 들어가면 Getting Start 부분에 Download 가 있음

ii. 여기서 최신 버전을 받도록 함

iii. 그 후 다운받은 Hadoop 의 압축을 해제

① tar xzf hadoop-0.20.203.0.tar.gz

iv. Hadoop 을 사용하려면 java 의 환경변수를 등록해주어야 함.

① cd hadoop-0.20.203.0/conf

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

② nano hadoop-env.sh

③ 맨 아랫줄에 java 환경변수를 적어줌. 작성자는 이러함
export JAVA_HOME=/usr/local/jre1.6.0_26

v. Hadoop 은 3 가지의 실행모드가 있다.

① Standalone Operation

- mkdir input
- cp conf/*.xml input
- bin/hadoop jar hadoop-examples-0.20.203.0.jar grep input
output 'dfs[a-z.]+'
- cat output/*

② Pseudo-Distributed Operation

● Setting

1. conf/core-site.xml

```
<configuration>  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://localhost:9000</value>  
  </property>  
</configuration>
```

2. conf/hdfs-site.xml

```
<configuration>  
  <property>  
    <name>dfs.replication</name>
```

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

```
<value>1</value>
</property>
</configuration>
```

3. conf/mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

- HDFS 포맷

1. bin/hadoop namenode -format

- SSH 연결 설정

1. ssh-keygen -t rsa

2. cp /home/psc/.ssh/id_rsa.pub /home/psc/.ssh/authorized_keys

- 데몬 프로그램 실행

1. bin/start-all.sh

- Input 파일을 HDFS 에 복사

1. bin/hadoop fs -put conf input

- 실행

1. bin/hadoop jar hadoop-examples-0.20.203.0.jar grep input output 'dfs[a-z.]+'

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- 출력

1. bin/hadoop fs -get output output
2. cat output/*

③ Fully-Distributed Operation

- Hadoop 설정파일에는 두종류가 있음

1. Read-only default Configuration

1. src/core/core-default.xml
2. src/hdfs/hdfs-default.xml
3. src/mapred/mapred-default.xml

2. Site-specific Configuration

1. conf/core-site.xml
2. conf/hdfs-site.xml
3. conf/mapred-site.xml

- Master Setting

1. conf/core-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://PSC:9000</value>
  </property>
</configuration>
```

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

2. conf/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name/>
    <value>/hdfs/data</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.block.size</name>
    <value>1048576</value>
  </property>
</configuration>
```

3. conf/mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>hdfs://PSC:9001</value>
  </property>
  <property>
    <name>mapred.system.dir</name/>
    <value>/hdfs/mapreduce/system</value>
  </property>
</configuration>
```

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

```
</property>
<property>
  <name>mapred.local.dir</name>
  <value>/hdfs/mapreduce/local</value>
</property>
</configuration>
```

- Slave setting

1. conf/core-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://PSC2:9000</value>
  </property>
</configuration>
```

2. conf/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name/>
    <value>/hdfs/data</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
```

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

```
</property>
<property>
    <name>dfs.block.size</name>
    <value>1048576</value>
</property>
</configuration>
```

3. conf/mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>hdfs://PSC2:9001</value>
  </property>
  <property>
    <name>mapred.system.dir</name/>
    <value>/hdfs/mapreduce/system</value>
  </property>
  <property>
    <name>mapred.local.dir</name>
    <value>/hdfs/mapreduce/local</value>
  </property>
</configuration>
```

- PSC 와 PSC2 를 신뢰관계로 등록

1. PSC 에서 ~/.ssh/authorized_keys 를 PSC2 의 id_dsa.pub 파일의 내용을 넣어주고
2. PSC2 에서 ~/.ssh/authorized_keys 를 PSC 의 id_dsa.pub 파일의 내용을 넣어줌

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

- 마스터에서 HDFS 포맷
 1. bin/hadoop namenode -format
- 데몬 프로그램 실행
 1. bin/start-all.sh

vi. Sample

① 소스는

http://hadoop.apache.org/mapreduce/docs/r0.21.0/mapred_tutorial.html

여기서 받아쓰기로 함

② 그 후 컴파일을 하도록 하는데 폴더를 생성해 주는것이 좋음

```
mkdir word
```

```
javac -classpath hadoop-core-0.20.203.0.jar -d word  
WordCount.java
```

③ 그리고 jar 파일로 만들어 줌

```
jar -cvf Word.jar -C word .
```

④ input 파일들을 만들어 주는데 사용자가 편한쪽으로 만듦

```
nano input/input1
```

```
//내용은 Hello World Bye World
```

```
nano input/input2
```

```
//내용은 Hello Hadoop Goodbye Hadoop
```

⑤ HDFS 에 input 파일을 등록

```
bin/hadoop fs -put input /user/psc/word/input
```

⑥ 등록됐는지 확인

```
bin/hadoop fs -ls /user/psc/word/input
```

개인용 슈퍼컴퓨팅 개발가이드(초안 v1.0 beta 6)

⑦ 실행

```
bin/hadoop fs jar Word.jar org.myorg.WordCount  
/user/psc/word/input /user/psc/word/output
```

⑧ 결과 확인

```
bin/hadoop fs -cat /user/psc/word/output/part-00000
```

GPGPU 개발환경 구축 : Hadoop 빌드 방법

1 Ant를 설치

A. `sudo apt-get install ant`

2 Ant가 잘 설치 되었는지 확인

A. `ant -version`

B. 만약 "Unable to locate tools.jar. Expected to find it in /usr/local/jre1.6.0_26/lib tools.jar" 라는 메시지가 나온다면, tools.jar 파일이 다른곳에 있을 가능성이 크므로 찾아서 옮김

C. `find -type f -name tools.jar`

D. 필자의 경우는 /usr/lib/jvm/java-6-openjdk/jre/lib 에 위치했음

E. `mv /usr/lib/jvm/java-6-openjdk/jre/lib /tools.jar $JAVA_HOME/lib`

3 빌드

A. w설치되어있는 hadoop 폴더에 build.xml파일을 빌드

B. `ant mvn-install`

4 Build Success라는 메시지가 나왔다면 build 폴더에 hadoop-0.20.203.1-SNAPSHOT이라는 폴더가 생성되었을 것이고, bin/hadoop 을 실행했을 때 돌아가는 것을 확인 하면 성공