



# Accelerating 3D Elastic Wave Simulation using PGI Accelerator Directives

## **Proof-of-Concept Guide**

# Table of Contents

## Contents

<b>Proof-of-Concept Guide.....</b>	<b>1</b>
<b>Step-by-Step Instructions.....</b>	<b>3</b>
Retrieve Software.....	3
Install Software.....	3
Setup User Environment .....	4
Build Executables .....	5
Measure Performance .....	5
<b>References.....</b>	<b>7</b>



# Step-by-Step Instructions

This section describes how to configure the 3D elastic simulation proof-of-concept and demonstrate GPU acceleration using PGI directives. A package has been developed that contains the original FORTRAN source module, the augmented source module with accelerator directives, and a makefile. These procedures were tested on a system running RedHat Enterprise Linux 64-bit operating environment, with a dual socket Westmere x5670 (2.93 GHz, 12-core) CPUs, and dual Fermi M2090 GPUs inserted in PCIe x16-Gen2 slots.

---

## Retrieve Software

A compressed tar file named *3D-Elastic-POC.tar.gz* is available from NVIDIA or PGI that contains the necessary software for the 3D elastic forward wave modeling test. The only additional software required for this proof-of-concept is the Portland Group PGI CDK Accelerator FORTRAN/C/C++ compilers (V11.8 or later). To obtain a free 30-day trial version of the PGI CDK Accelerator kit, please contact:

Tom Toy  
Sales Manager,  
The Portland Group  
[thomas.toy@pgroup.com](mailto:thomas.toy@pgroup.com)  
Direct: +1 (503) 431-7143  
Mobile: +1 (503) 919-9853

and provide the name and email address of the individual in your organization who will be investigating the 3D Elastic Proof-of-Concept using PGI Accelerator tools.

---

## Install Software

Follow the [instructions to install the PGI software](#) on your system, and since the 3D elastic simulation uses MPI, be sure to install the PGI version of MPI (MPICH2) during this process. You will need cluster root privileges during the installation, and can install on any node, or multiple nodes in the cluster. For support during this process, please reference the [PGI installation support FAQ site](#), or contact Tom Toy.

---

## Setup User Environment

The next step is to adjust the user environment variables to properly reference the PGI tools. An example for Bourne-shell, or c-shell is included, and assumes the PGI tools were installed in the */opt/pgi*.

For bash environment add the following to  $\$(HOME)/.bashrc$

```
export PATH=/opt/pgi/linux86-64/11.8/bin:$PATH
export MANPATH=/opt/pgi/linux86-64/11.8/man:$MANPATH
export LM_LICENSE_FILE=$LM_LICENSE_FILE:/opt/pgi/license.dat
export PGI=/opt/pgi
export PGRSHA=ssh
```

For csh environment add the following to  $\$(HOME)/.cshrc$

```
set path=(/opt/pgi/linux86-64/11.8/bin $path)
setenv MANPATH /opt/pgi/linux86-64/11.8/man:$MANPATH
setenv LM_LICENSE_FILE $LM_LICENSE_FILE:/opt/pgi/license.dat
setenv PGI /opt/pgi
setenv PGRSHA ssh
```

---

## Build Executables

- `tar xvzf /tmp/3D-Elastic-POC.tar.gz`  
3D-Elastic-POC/  
3D-Elastic-POC/makefile  
3D-Elastic-POC/run\_cpu.sh  
3D-Elastic-POC/seismic\_CPML\_3D\_isotropic\_MPI\_OpenMP.F90  
3D-Elastic-POC/run\_gpu.sh  
3D-Elastic-POC/seismic\_CPML\_3D\_isotropic\_MPI\_ACC.F90  
3D-Elastic-POC/gold/  
3D-Elastic-POC/gold/energy.dat
- `cd 3D-Elastic-POC/`
- `make`  
  
`mpif90 -V11.8 -fast -mp -Minfo -mcmodel=medium -DNXSIZE=101 -`  
`DNYSIZE=641 -DNZSLICE=64 -DNUM_PROCS=2`  
`seismic_CPML_3D_isotropic_MPI_OpenMP.F90 -o cpu.out.2.64 >&`  
`buildcpu.log`  
  
`mpif90 -V11.8 -fast -ta=nvidia -Minfo=accel -DNXSIZE=101 -DNYSIZE=641`  
`-DNZSLICE=64 -DNUM_PROCS=2`  
`seismic_CPML_3D_isotropic_MPI_ACC.F90 -o gpu.out.2.64 >&`  
`buildgpu.log`

---

## Measure Performance

Since this test combines MPI and OpenMP, multiple tests can be performed to measure scalability across CPU cores and GPUs within a single system or across a cluster of nodes. It is possible to adjust the domain decomposition in the z-dimension of the seismic volume to be processed and measure scalability. The makefile is setup to adjust volume sizes when the executable is built, and a corresponding rule to properly invoke “*mpirun*” for the CPU and GPU experiments. The number of OpenMP threads can be adjusted for scalability tests by modifying the *run\_cpu.sh* script and changing the *OMP\_NUM\_THREADS* environment variable.

```
run_cpu:
    /usr/bin/time mpirun -np $(NP) run_cpu.sh cpu.out.$(NP).$(NZSLICE)

run_gpu:
    /usr/bin/time mpirun -np $(NP) gpu.out.$(NP).$(NZSLICE)
```

The contents of the run\_cpu.sh script:

```
#!/bin/bash

export OMP_NUM_THREADS=6
$1
```

The package contains a run script designed to compare 2 MPI rank using 6 OMP threads on the host server, against 2 MPI rank using 2 Tesla GPUs.

```
#!/bin/bash

make -f makefile run_cpu
make -f makefile run_gpu
```

This framework can be extended to compare scalability when the size of the volume is adjusted by modifying the variables NP and NZSLICE (by powers of two) in the makefile.



## References

- **“Seismic Convolutional Perfect Matched Layer (CPML)”, Dimitri Komatitsch and Roland Martin**
  - [http://web.univ-pau.fr/~dkomati1/README\\_seismic\\_cpml.html](http://web.univ-pau.fr/~dkomati1/README_seismic_cpml.html)

This software is governed by the [CeCILL license \(version 2\)](http://www.cecill.info) (a French license very similar to GNU GPL version 2) under French law and abiding by the rules of distribution of free software. You can use, modify and/or redistribute the software under the terms of the CeCILL license as circulated by CEA, CNRS and INRIA at the following URL "<http://www.cecill.info>".

## **Notice**

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## **Trademarks**

NVIDIA, the NVIDIA logo, Tesla and CUDA are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## **Macrovision Compliance Statement**

NVIDIA Products that are Macrovision enabled can only be sold or distributed to buyers with a valid and existing authorization from Macrovision to purchase and incorporate the device into buyer's products.

Macrovision copy protection technology is protected by U.S. patent numbers 5,583,936; 6,516,132; 6,836,549; and 7,050,698 and other intellectual property rights. The use of Macrovision's copy protection technology in the device must be authorized by Macrovision and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Macrovision. Reverse engineering or disassembly is prohibited.

## **Copyright**

© 2011 NVIDIA Corporation. All rights reserved.