GRIDIFICATION OF EARTHQUAKE LOCATION FINDING AND REGIONAL FAULT PLANE SOLUTION APPLICATIONS

by

Bilal Bektaş

B.S, Mathematics Engineering, Istanbul Technical University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computational Science and Engineering

Boğaziçi University

2013

GRIDIFICATION OF EARTHQUAKE LOCATION FINDING
AND
REGIONAL FAULT PLANE SOLUTION APPLICATIONS

APPROVED BY:

Prof. Can Özturan (Thesis Supervisor)	.....
Prof. Oğuz Tosun	.....
Assoc. Prof. Ali Ecder	.....

DATE OF APPROVAL: 18.02.2013

ACKNOWLEDGEMENTS

This research is supported by the FP7-SEEGRID-SCI project under the grant number 211338. I thank TUBITAK-ULAKBIM personnel for their help and for making available the TR-GRID infrastructure.

I owe the greatest gratitude to Prof. Can Özturan who has helped me prepare my master thesis and supported me throughout my graduate life and Dr. Mehmet Yılmazer who has helped me in this research. In addition, I am eternally grateful for my family who support me all the time.

ABSTRACT

GRIDIFICATION OF EARTHQUAKE LOCATION FINDING AND REGIONAL FAULT PLANE SOLUTION APPLICATIONS

In this thesis, two studies have been conducted. One of them is Gridification of Earthquake Location Finding (ELF) Application and the other one is Gridification of Regional Fault Plane Solution (FPS) Application. Earthquake Location Finding application finds the hypocenter of an earthquake by using the seismic waveform data generated by seismic stations. ELF is data intensive since it needs to process waveform files from several stations. The application is parallelized by decomposing the problem space and expressing it as a parallel JDL workflow. To implement spatial decompositions, stations are divided among the worker nodes. A node is assigned a task which is responsible for accessing a specific station's waveform file and computing picks. The results from all worker nodes are then sent to a collector node which runs the HYPO71 program to locate hypocenter of the earthquake. Timing results indeed show that parallelization of file accesses improves performance. The main goal of Regional Fault Plain Solution (FPS) application is the determination of the faulting mechanism that characterizes earthquake parameters (dip,strike, rake) by utilizing regional Moment Tensor Inversion method. The most time-consuming part of the FPS application involves the calculation of Green's functions. Therefore, calculation of the Green's functions has been parallelized. A Job Description Language workflow is automatically created based on the initial parameters of individual crustal models that are necessary for producing Green functions. Timing results obtained indicate that the FPS application did achieve speed-up as a result of parallelization.

ÖZET

DEPREM YERİNİ BULMA VE BÖLGESEL FAY DÜZLEMİ ÇÖZÜMÜ UYGULAMALARININ GRİDLEŞTİRİLMESİ

Bu tez kapsamında iki adet çalışma yapılmıştır. Bunlardan birisi, deprem yerini bulma (ELF) uygulamasının gridleştirilmesi; diğeri ise bölgesel fay düzlemi çözümü (FPS) uygulamasının gridleştirilmesidir. Deprem Yerini Bulma uygulaması (ELF), sismik istasyonlar tarafından üretilen sismik dalga verilerini kullanarak depremin merkezini bulur. ELF veri yoğunluklu bir uygulamadır. Çünkü birçok istasyon tarafından üretilmiş olan dalga formundaki veri dosyalarını işlemesi gerekmektedir. Problem uzayının ayrıştırılması ile ve bunu da paralel JDL iş akışı şeklinde ifade edilmesi ile uygulama paralelleştirilmiştir. Konumsal ayrışmaları uygulayabilmek için işçi düğümler arasında istasyonlar paylaştırılır. Belirli bir istasyonun dalga formundaki dosyasına ulaşma ve buradaki pik değerlerinin hesaplanması görevi bir düğüme verilir. İşçi düğümdeki tüm sonuçlar, HYPO71 programını kullanarak deprem merkezini bulan toplayıcı düğüme gönderilir. Dosya erişiminin paralelleştirilmesi yaklaşımının performans artışı sağladığı test sonuçlarında görüldü. Bölgesel Fay Düzlemi Çözümü (FPS) uygulamasının temel amacı deprem parametrelerini (eğim, doğrultu, kayma) karakterize eden fay mekanizmasının moment tensör ters çözümü (MTI) yönteminin kullanılarak belirlenmesidir. FPS uygulamasının en çok zaman alan kısmı Green fonksiyonlarının hesaplanmasıdır. Bu nedenle, Green fonksiyonlarının hesaplanması paralelleştirilmiştir. Green fonksiyonlarını üretmek için gerekli olan bireysel kabuk modellerinin başlangıç parametrelerine göre iş akışı (JDL workflow) otomatik olarak oluşturulur. Zaman sonuçları, paralelleştirme sayesinde FPS uygulamasında hız artışı olduğunu göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . vii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
2. RELATED WORKS . . . 5
3. GRIDIFICATION OF EARTHQUAKE LOCATION FINDING APPLICATION 8
3.1. Methodology for Finding the Location of Earthquake Center . . . 8
3.2. Methods of Parallelization . . . 9
3.3. Development Details . . . 11
3.4. Automatic JDL Generation . . . 13
4. GRIDIFICATION OF REGIONAL FAULT PLANE SOLUTION APPLICATION . . . 18
4.1. Moment Tensor Inversion (MTI) . . . 18
4.2. Input Parameters and Data Preparation . . . 20
4.3. Parallelizing Synthetic Seismogram Calculation . . . 22
5. TEST AND RESULTS . . . 30
5.1. Gridification of Earthquake Location Finding Application . . . 30
5.2. Gridification of Regional Fault Plane Solution Application . . . 32
6. CONCLUSIONS . . . 34
APPENDIX A: SEISMIC DATA STORAGE AND ORGANIZATION . . . 35
APPENDIX B: USAGE OF ELF APPLICATION . . . 39
B.1. Example usage of ELF application . . . 39
APPENDIX C: USAGE OF FPS APPLICATION . . . 42
C.1. Example usage of FPS application . . . 42
REFERENCES . . . 45

LIST OF FIGURES

Figure 1.1. Architecture of SEE-GRID-SCI 2

Figure 3.1. Computation of earthquake centers from waves arriving at the stations. 9

Figure 3.2. Parallelization by dividing the stations. 10

Figure 3.3. Parallelization by dividing the time. 11

Figure 3.4. Workflow used in the station (spatial) division method. 12

Figure 3.5. Workflow used in the time division method. 13

Figure 3.6. JDL workflow for serial ELF program. 14

Figure 3.7. Parallel workflow for the station (spatial) partition strategy (part-1). 15

Figure 3.8. Parallel workflow for the station (spatial) partition strategy (part-2). 16

Figure 4.1. Basic schema of FPS workflow on one worker node. 21

Figure 4.2. Example of Model.ini file. 22

Figure 4.3. Example file of Crustal Model for synthetic seismogram calculation. 23

Figure 4.4. FPS workflow on multiple worker nodes. 25

Figure 4.5. Example of automatically generated JDL workflow file (part-1). . 26

Figure 4.6. Example of automatically generated JDL workflow file (part-2). . 27

Figure 4.7. Jobs distributions and workflow of FPS on P-GRADE Grid portal. 29

Figure 5.1. KML file generated by SDSAS application service. 31

Figure A.1. Logical file hierarchy. 36

Figure A.2. The name format of the file containing seismic waveform data. . . 37

Figure A.3. The names of the example files containing seismic waveform data. 37

Figure A.4. Proxy command. 37

Figure A.5. An example usage of insert_waveformdata.sh. 38

Figure A.6. .mdclient.config file. 38

Figure B.1. Usage type of the ELF application. 39

Figure B.2. Proxy command. 40

Figure B.3. Example usage of the ELF application to generate JDL file for station divided parallel jobs. 40

Figure B.4. Job submission to the Grid. 40

Figure B.5. Controlling job status. 41

Figure B.6. Downloading output files of the job. 41

Figure C.1. Usage type of the FPS application. 42

Figure C.2. Proxy command. 43

Figure C.3. Example usage of the FPS application to generate JDL file. 43

Figure C.4. Job submission to the Grid. 43

Figure C.5. Controlling job status. 44

Figure C.6. Downloading output files of the job. 44

LIST OF TABLES

Table 5.1. Various tests performed and timings obtained. 30

Table 5.2. Test results for FPS application in different conditions. 32

LIST OF SYMBOLS

A_{ij}	Green's Functions or components of synthetic seismograms
d	Density of layer
h	Depth of layer
M_0	Seismic Moment
M_{ij}	Moment Tensor Components
M_L	Local Magnitude
Q_p	Q factor for P wave respectively
Q_s	Q factor for S wave respectively
S_j	Waveform Seismic Data
V_p	P wave velocity of layer
VR	Variance Reduction
δ	The angle between the fault plane and the Earth's surface
λ	The angle between slip direction and the fault line
ϕ	The angle between fault direction and geographic north

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
3D	Three Dimensional
AMGA	The ARDA Metadata Grid Application
DAG	Direct Acyclic Graph
DMT_INV	Deviatoric Moment Tensor Inversion
EGEE	Enabling Grid for E-science
ELF	Earthquake Location Finding
EW	East and West
FPS	Fault Plane Solution
HYPO71	Earthquake location program
JDL	Job Description Language
LFC	LCG File Catalog
MTI	Moment Tensor Inversion
NS	North and South
P-GRADE	Parallel Grid Run-time and Application Development Environment
PREPDATA	A program which was developed converting waveform synthetic data into SAC format, the format used by the TDMT_INVC program
SDSAS	Seismic Data Server Application Service
SEE-GRID	South Eastern European Grid-enabled eInfrastructure Development
SEE-GRID-SCI	South Eastern European Grid e-Infrastructure for Regional e-Science
SZTAKI	Computer and Automation Research Institute (SZTAKI) of the Hungarian Academy of Sciences
TDMT_INVC	Time Domain Moment Tensor Inverse Code
TSBW_INV	Teleseismic Body-Wave Inversion
WMS	Workload Management System

1. INTRODUCTION

In the last century, there have been various technological and scientific developments. These developments have enabled us to share resources among distributed systems. Hence, E-Science began to develop on several platforms. One of them is grid platform in which computational resources, data, devices and applications in the so-called virtual organizations (VOs) can be shared. Scientists can access and benefit from these resources by joining virtual organizations associated with their research area.

South Eastern European Grid e-Infrastructure for Regional e-Science (SEE-GRID-SCI) (SEE-GRID eInfrastructure for regional eScience) [**?**] co-funded by the European Commission has started on May 1st 2008 after Southeastern Europe Grid enabled e-Infrastructure Development (SEE-GRID1 and SEE-GRID2) were concluded. National Grid Initiatives of SEE-GRID-SCI are Albania, Bosnia-Herzegovina, Bulgaria, Croatia, FYR of Macedonia, Greece, Hungary, Serbia - Montenegro, Romania and Turkey. Applications which are categorized in three virtual organizations (VOs) and application services have been developed in SEE-GRID-SCI whose core purpose is to engage user communities from different countries in close collaboration so as to explore application similarities and extract out common elements with respect to the approaches to actual Grid usage [1]. Architecture of SEE-GRID-SCI can be seen in Figure 1.1 and name of these applications and services can be found as follows [2] :

- Seismology VO :
  - (i) ELF: Earthquake Location Finding
  - (ii) FPS: Fault Plane Solution
  - (iii) MDSSP-WA: Massive Digital Seismological Signal Processing with the Wavelet Analysis
  - (iv) NMMC3D: Numerical Modeling of Mantle Convection in 3D
  - (v) SDS: Seismic Data Server
  - (vi) SRA: Seismic Risk Assessment

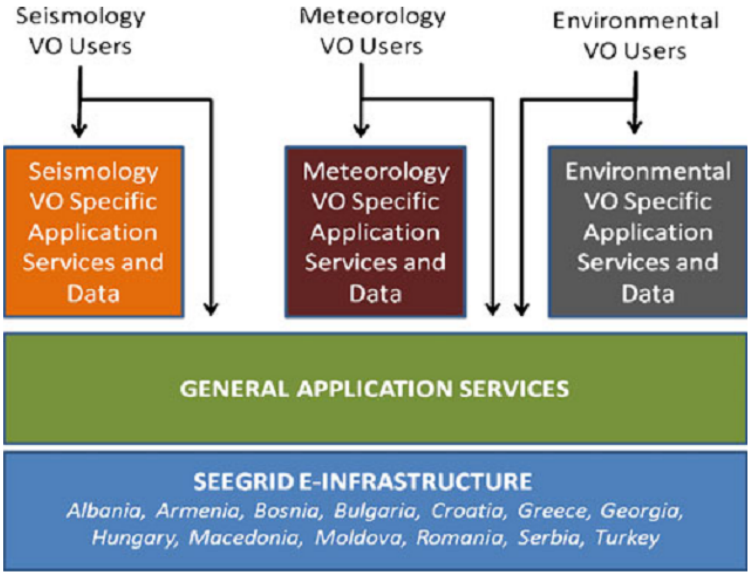


Figure 1.1. Architecture of SEE-GRID-SCI [3].

NMMC3D and SRA will be discussed in Related Works chapter.

- Meteorology VO :
  (i) REFS: Regional Scale Multi-Model, Multi-Analysis Ensemble Forecasting System
  (ii) WRF-ARW: Weather Research and Forecasting – Advanced Research WRF
- Environmental VO :
  (i) CHERS: Study of CHanges of Environment with Remote Sensing
  (ii) GreenView: Refinement of surface and vegetation parameters in SEE region based on satellite images
  (iii) Lizza-PAKP: Pseudo-acronym of the application for modeling groundwater transfer through porous medium
  (iv) MCSAES: Monte Carlo Sensitivity Analysis for Environmental Systems
  (v) MSACM: Multi-Scale Atmospheric Composition Modeling
  (vi) MSERRHSA: Modeling System for Emergency Response to the Release of Harmful Substances in the Atmosphere
- Application Services :
  (i) DM-WEB: Data Management Web Portal
  (ii) FM-J-API: SEE-GRID File Management Java API

(iii) SDSAS: Seismic Data Server Application Service
(iv) AWT: Advanced-workflow development tool and orchestration service
(v) ULMON: User Level Monitoring Tool
(vi) WORK BINDER
(vii) RAS: Rendering Application Service
(viii) ESIP PLATFORM: Environment oriented Satellite Data Processing Platform
(ix) CWRE: Common Workflow Repository Extension of P-GRADE Portal
(x) EVENT LOGGER
(xi) MEWS: Mathematical Expressions Web Service

In the SEE-GRID-SCI project massive waveform data is being collected from different countries. Currently, there are 3.4 terabytes of data in the archives of SEE-GRID-SCI Seismology Virtual Organization [4]. Waveform data are stored mostly in the form of 1-hour files. Currently, there are over 7 million waveform files in the archive.

In this thesis, gridification of earthquake location finding (ELF) application was done as part of the SEE-GRID-SCI project work. Earthquake location finding application can also be used with archived data in order to find historical earthquakes. For this reason Kandilli Observatory and Research Institute uses a HYPO71 [5] based application. ELF program was gridified and parallelized by using JDL workflows.

Secondly, gridification of regional fault plane solution (FPS) application was also part of the SEE-GRID-SCI project. The most time-consuming part of the FPS application involves the calculation of Green's functions. In order to speed up the application, this calculation is parallelized by using job description language workflows.

In the following chapters, related works about earthquake location finding (ELF) and regional fault plane solution (FPS) are given firstly. Afterwards, detailed information about gridification of ELF application will be explained. Chapter 3 contains 4 sections which are (i) methodology for the location of earthquake center, (ii) methods of parallelization, (iii) development details, and (iv) automatic JDL generation.

Thirdly, gridification of regional FPS application will be explained by giving information about moment tensor inversion method, input parameters, data preparation and parallelization of synthetic seismogram calculations. Later, test results of parallel ELF and FPS applications will be compared with their sequential versions. Subsequently, conclusion and future works will be given. In Appendix A, information about seismic data storage and organization will be given and how the new waveform data files can be entered into grid environment will be explained. In Appendix B and C, the usage of ELF and FPS applications will be described with the aid of examples.

2. RELATED WORKS

Applications that find earthquake locations by processing waveform data collected from seismology stations are quite important since they are used daily by seismologists. Most of the other applications in seismology are also based on location finding. For instance, Doser has estimated magnitude and location of Alaskan earthquakes by analyzing historical earthquakes occurring between 1899 and 1965 in the south-central, interior, and southeast of Alaska [6]. In addition, Bakun has researched implications of 1855 Ansei Earthquake, great 1923 Kanto earthquake and the 1855 and 1923 events to Philippine Sea–Eurasian plate interface by using Japan Meteorological Agency (JMA) intensity assignments [7]. In another research, after Bakun et al. have developed a Modified Mercalli intensity (MMI) model by analyzing 28 calibration events from Texas to the Grand Banks, Newfoundland. They have estimated source parameters of historical earthquakes in eastern North America by using this model [8]. Therefore, finding earthquake locations is crucially important. In this thesis, an application which can find earthquake location with historical waveform data by running parallel on grid platform has been developed.

In addition to gridification of earthquake location finding application, gridification of regional fault plane solution application has been developed because one of the most significant problems of seismology is the definition of the fault mechanism parameters (strike, dip and rake). Especially after the recent developments on broad-band seismometers, MTI method became the most popular solution for the determination of the Fault plane parameters. For example, Ichinose *et al.* have estimated for eastern California and western Nevada earthquakes from regionally recorded long-period seismograms using the moment-tensor inversion method and they have validated the moment tensor solutions by a comparison with Harvard centroid moment tensor (CMT) and P-wave first-motion focal mechanism solutions [9]. In another example, Stich *et al.* have presented and discussed a set of 77 moment tensor solutions for earthquakes in the Iberia–Maghreb region from mid-2005 to the end of 2008, with moment magnitudes ranging from 3.2 to 6.0 [10].

After 1990, a lot of analysis software was developed for MTI with many of them being often based on three main software packages: One of them is DMT_INV (Deviatoric Moment Tensor Inversion) developed by Ammon and Randall [11] that focus on regional scale solutions. The second one is TSBW_INV (Teleseismic Body-Wave Inversion) which is a software package developed by Kikuchi and Kanomori [12] for global scale solutions of fault mechanisms. The third one is TDMT_INV (Time Domain Moment Tensor Inversion) package developed by Dreger [13] for regional solutions. All these software packages consist of two stages: the first step is calculation of synthetics seismograms and the second stage is application of MTI procedure.

When literature review is done, it can be seen that lots of regional fault plane solutions have been done. One of them is study about seismicity, fault plane solutions, depth of faulting, and active tectonics of The Andes of Peru, Ecuador, and Southern Colombia [14]. Suárez *et al.* scrutinize the interaction of these structures. In addition, Forsyth has researched focal mechanisms which are presented for 46 earthquakes that occurred in the South Atlantic Ocean, in the Scotia Sea, and in southern Chile during the period 1963–1973 [15]. Moreover, Papazachos *et al.* have researched fault plane solutions for 127 earthquakes, of magnitude $M >= 5.5$ and shallow focal depth ($h < 60km$), which occurred in the Aegean Sea and surrounding area (34°N-43°N, 18°E-30°E) during the period 1953-1995 [16]. Furthermore, Hinzen has studied stress field in the Northern Rhine area, Central Europe, and determined fault plane solutions (FPS) from 110 earthquakes in the northern Rhine area with local magnitudes, ranging from 1.0 to 6.1, and occurring between 1976 and 2002 [17]. While these studies focus on solutions of specific earthquakes, FPS focuses on providing parallel solutions of earthquakes.

In addition, the most recent application that is integrated with EGEE (Enabling Grid for E-science) is developed by Clévédé *et al.* [18] for fault plane solution. The main difference between this work and our FPS application is that while EGEE application is using teleseismic data, FPS is using regional broad-band records.

In addition to gridification of earthquake location finding and regional fault plane

solution applications, there are two applications which use grid infrastructure in SEE-GRID-SCI Seismology Virtual Organization. One of them is Seismic Risk Assessment (SRA) application developed by Şener *et al.* [19]. For public safety and hazards mitigation, SRA application produces seismic hazard maps for South-East European (SEE) region by using a uniform earthquake catalog available on the grid. Secondly, the Numerical Modeling of Mantle Convection (NMMC3D) application is developed by Miklos *et al.* [20]. To study the structure and the surface manifestation of the mantle plumes, NMMC3D application calculates mantle convection models in 3D Cartesian domain. It uses SEE-GRID-SCI grid infrastructure in order to submit the developed workflow model.

3. GRIDIFICATION OF EARTHQUAKE LOCATION FINDING APPLICATION

The main aim of parallelization is to speed up Earthquake Location Finding Application. In order to achieve this, different parallelization strategies are investigated. These are: (i) the use of MPI message passing libraries, (ii) the use of OpenMP multithreading directives and (iii) expressing the parallel application as a workflow using Job Description Language (JDL). Since the application is based on legacy HYPO71 program written in Fortran, we try to minimize the changes that have to be performed on the code. Such an approach will also help us to quickly parallelize and gridify our application. For this reason, parallelization is achieved without changing the original code and by just dividing the input files among multiple processors. The tasks that result from this division and the dependencies among the tasks are then expressed as a JDL workflow.

In the following sections, the details of how earthquake centers are found are given firstly. Then, the details of parallelization and gridification using workflows are given.

3.1. Methodology for Finding the Location of Earthquake Center

The process of finding the earthquake center involves the following stages:

(i) Computation of the arrival times of the earthquake waves to the stations by comparing magnitude changes [21]. This computation is implemented by scanning waveform files of all stations corresponding to a given interval.

(ii) Passing of the output of stage 1 to HYPO71 program and computation of the hypocenter of the earthquake.

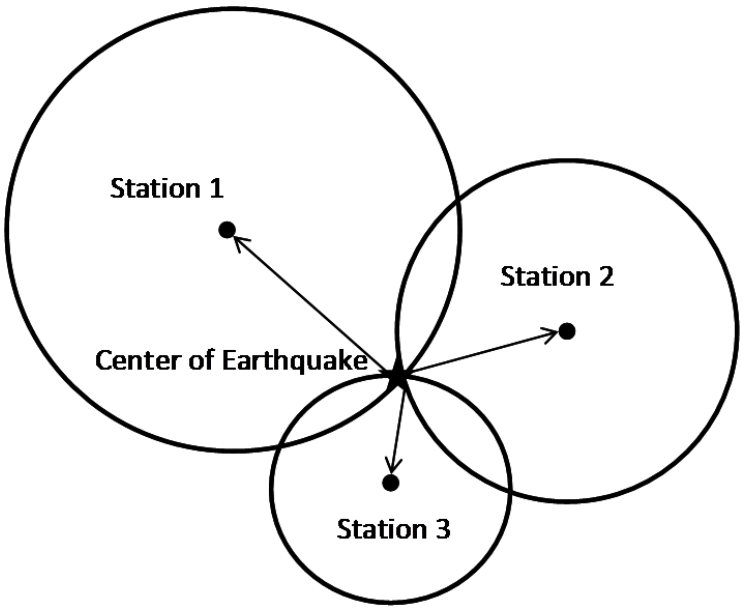


Figure 3.1. Computation of earthquake centers from waves arriving at the stations.

In the first stage, all waveform data are scanned. In a sliding time window, amplitude changes of waveform data recorded by stations are looked for. External events, like passing of a truck near a station, can also lead to the changes in the amplitudes of signals recorded by that station. In order to remove such noises, amplitude changes in at least 4 stations in a time window of 10 seconds are sought. If this condition is satisfied, then it is concluded that an earthquake happened. Otherwise, it is taken as noise.

Figure 3.1 shows how the earthquake location finding algorithm works. Distance of an earthquake location from a station can be computed by using the arrival time of the earthquake wave to the station and the speed of earthquake wave in various earth layers. Earthquake location can be found by intersecting equidistant locations (which form circles) around stations. A minimum of 3 stations are required for this purpose.

3.2. Methods of Parallelization

In order to parallelize earthquake location finding, problem space can be divided spatially or temporally. This gives us two approaches for parallelization.

The first approach is illustrated in Figure 3.2. In this approach, data of each station are processed independently and the access time of an earthquake wave to the station is computed separately.

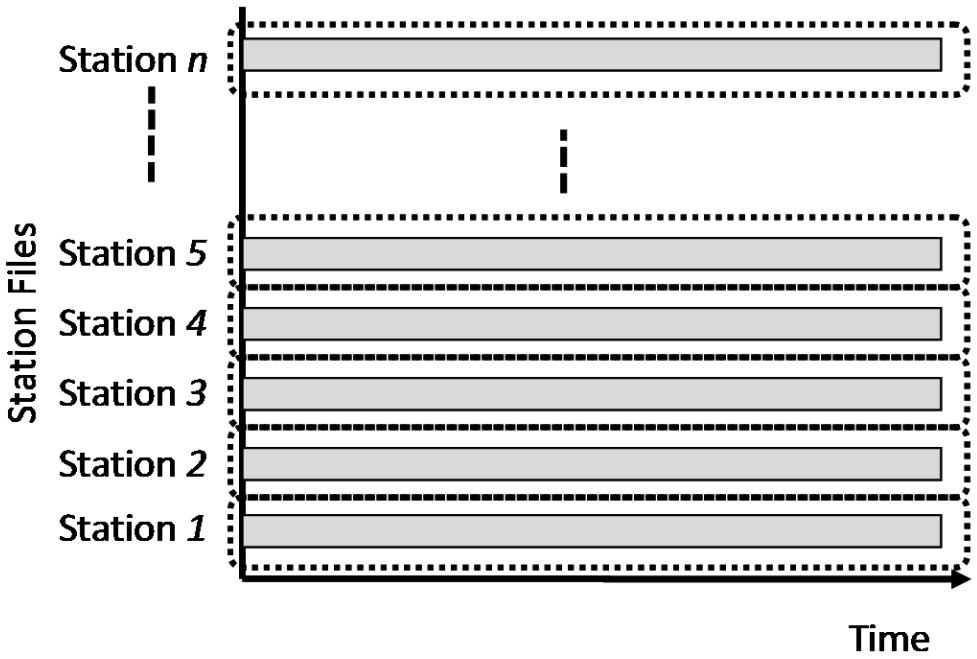


Figure 3.2. Parallelization by dividing the stations.

In the second approach, time range is divided into small sub-intervals and these sub-intervals are processed in a parallel manner. In each sub-interval, data from all the stations are processed. The data are scanned in order to determine earthquake waves and their arrival times to the stations. These are then passed to the HYPO1 program. This processing is carried out on each sub-interval. Figure 3.3 shows how a time interval is divided into four sub-intervals. It should be noted here that one should be careful in order to allow an overlap time that is at least as big as the sliding window. This will make sure that wave magnitude changes on the subinterval borders are not missed.

The second approach is not implemented because waveform data files of all stations in specific time interval are required for each node that is assigned to be responsible in this time interval. In grid environment, net communication cost is expensive.

In addition, it is costlier to transfer small files than a single large file.

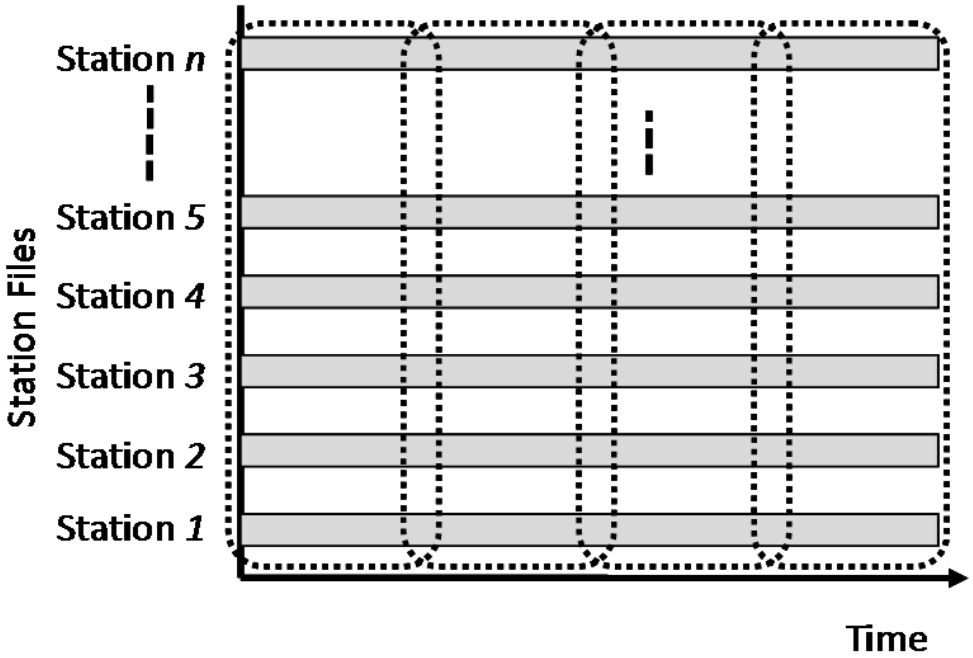


Figure 3.3. Parallelization by dividing the time.

3.3. Development Details

In [22], the serial version of earthquake location finding program is gridified. In [22], however, no parallelization is done. The parallelization strategies mentioned in Section 3.2 can easily be implemented by expressing them as a JDL workflow. In order to carry out parallelization in this manner, the only thing that needs to be done is the partitioning of the application's input.

In the spatial division method, data files corresponding to the stations and residing on storage elements (SE) are distributed to the worker nodes (WN) that work in parallel. Each worker node scans the station data file it is assigned to in order to determine the waves and their arrivals (picks). When all these picks have been collected, earthquake location finding program HYPO71 is executed on a collection node and the hypocenter is found. The workflow that corresponds to this strategy is displayed in Figure 3.4.

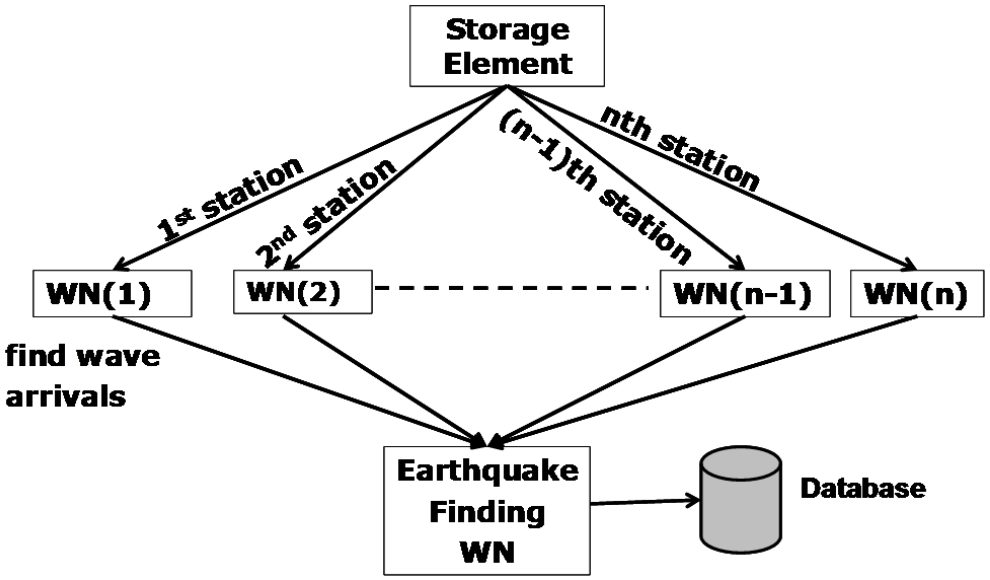


Figure 3.4. Workflow used in the station (spatial) division method.

Figure 3.5 demonstrates the workflow that is used in the time division method. Here, firstly, files are partitioned into smaller files corresponding to each sub-interval. Later, each worker node is sent all the files corresponding to the sub-interval it is assigned to. All the stages (finding of the picks and the invocation of HYPO71 to find the hypocenter) are carried out on each worker node.

In addition, some developments have been done when earthquake location finding program is gridified. One of them is that the algorithm in [23] is used in order to find magnitude and depth. Secondly, MSEED2SAC [24] converter is integrated to the ELF application. This change provides the ELF application with the opportunity to be run with seismic data in Mini-SEED format. Thus, the application can use seismic data in other formats. Thirdly, the waveform file handles are obtained by using the SDSAS application service [25] which is integrated into ELF application. Hence, waveform file queries can become very fast. Moreover, the new version of ELF application gets time range to analyze waveform data and hence speeds up the process.

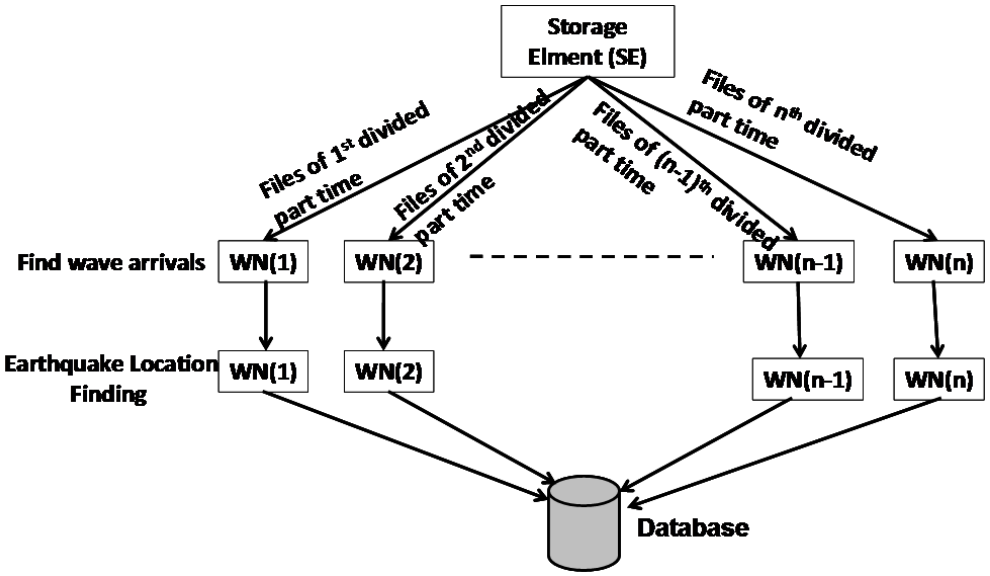


Figure 3.5. Workflow used in the time division method.

3.4. Automatic JDL Generation

Manual preparation of JDLs can be time-consuming and error prone. Therefore, a more practical way of JDL preparation is needed. To accomplish this, a program is developed. Users can benefit from this program to automatically generate workflows and to submit the workflow job to the grid.

Figure 3.6 shows a workflow for a serial ELF application. In the JDL file, RUN-SINGLEHOUR script file is performing some preprocessing activities related to LFC catalog, storage element and the AMGA catalogue. After the preprocessing is performed, the input parameters are passed to the elf.exe file for execution.

The template.txt file in the serial JDL file contains information about the latitude, longitude and depth information about the stations. Katsay.tmp file holds coefficient information of stations. Response.zip file includes the so called Pole Zero files which contain information about station sensors. Files.zip file contains mdclient.config for AMGA, libMDClient.so.0 library file for ELF application running on grid sites, and files of mseed2sac converter.

JDL file uses the arguments such as the year, month, day, and hour so as to determine which station files will be processed by the elf.exe program. After carrying out the computations about the earthquake finding, it records the found earthquakes to the AMGA database [26]. AMGA mdclient.config file is used during the saving of earthquake results.

JDL workflow generated for the station partitioning strategy is presented in Figure 3.7 and Figure 3.8. The JDL file uses DAG type workflow and the seismo.see-grid-sci.eu virtual organization. The list files that will be processed in each "part" are presented in the "arguments" section. In the workflow given, processes numbered 0 through 69 are carried out in parallel and each process records the analysis results to the storage element. After all the part0 through part69 processes are finished, part70 (HYPO71) process is started to find the earthquake location from the picks and record them to the AMGA database. Dependencies among the processes are specified in the

```
Type = "Job";
JobType = "Normal";
Executable = "ELF";
StdOutput = "out.txt";
StdError = "err.txt";
InputSandbox = {
    "ELF","ELF.exe","template.txt",
    "katsay.tmp","response.zip",
    "Files.zip"
};
OutputSandbox = {
                "out.txt","err.txt","1","2","3"
                       };
Arguments = "1 TR KAN 2007 01 01 01";
RetryCount = 3;
```

Figure 3.6. JDL workflow for serial ELF program.

```
type = "dag";
VirtualOrganisation = "seismo.see-grid-sci.eu";
Rank = other.GlueCEStateFreeCPUs;
max_nodes_running = 71;
InputSandbox = {
      "ELF","ELF.exe","template.txt",
      "katsay.tmp","response.zip",
      "Files.zip"
};
nodes = [
  part0 = [
     description = [
       Type="Job";
       JobType="Normal";
       Executable = "ELF";
       StdOutput = "out.txt";
       StdError = "err.txt";
       OutputSandbox = { "out.txt","err.txt","1","2","3" };
       Arguments = "2 TR KAN 2008 03 15 12
                    20080315T120000-20080315T130000-TR-KAN-ADVT-BHZ.zip
                    process-20080315T12-50.0 1";
       RetryCount = 3;
       ];
  ];

...
```

Figure 3.7. Parallel workflow for the station (spatial) partition strategy (part-1).

```
...
  part69 = [
     description = [
       Type="Job";
       JobType="Normal";
       Executable = "ELF";
       StdOutput = "out.txt";
       StdError = "err.txt";
       OutputSandbox = { "out.txt","err.txt","1","2","3" };
       Arguments = "2 TR KAN 2008 03 15 12
                    20080315T120000-20080315T130000-TR-KAN-ADVT-BHZ.zip
                    process-20080315T12-50.69 1";
       RetryCount = 3;  ];  ];
  part70 = [
     description = [
       Type="Job";
       JobType="Normal";
       Executable = "ELF";
       StdOutput = "out.txt";
       StdError = "err.txt";
       OutputSandbox = { "out.txt","err.txt","1","2","3" };
       Arguments = "2 TR KAN 2008 03 15 12 dummy
                     process-20080315T12-50.70 2";
       RetryCount = 3;
    ]; ]; ];
dependencies={{{part0, ... , part69 },part70}};
```

Figure 3.8. Parallel workflow for the station (spatial) partition strategy (part-2).

"dependencies" section at the end of the workflow. Here, it is stated that process part70 is dependent on the processes part0,..., part69 and it hence will be started after all these have been completed.

The workflow used for the time division strategy has a similar structure to that of the spatial division strategy. The only difference is that the dependencies section at the end is now empty. Thus, each process can compute earthquake locations in its sub-interval independently of other processes.

4. GRIDIFICATION OF REGIONAL FAULT PLANE SOLUTION APPLICATION

The major goal of this thesis is to accelerate fault plane solution (FPS). In order to fulfill this goal, parallelization is performed by generating artificial seismogram level which is computationally intensive. FPS is parallelized by creating a process for each crustal model file at one of the seven different depths. This parallelization is done using Job Description Language (JDL) workflows with the help of Kandilli Observatory and Research Institute. Moreover, no algorithm change is required in order to run FPS application in parallel on the grid because the process for each crustal model file is independent of each other and there is no intermediate communication or data transfer between these separate processes.

In the FPS, synthetic seismograms or Green's functions are calculated, based on Haskell's theory [27], using a frequency domain algorithm called FKRPROG that was developed for horizontally layered media by Saikia [28]. For MTI procedure FPS application uses Dreger's TDMT_INV [13, 29, 30] software on SAC (Seismic Analysis Code, [31]) formatted data.

In the following sections, the details of moment tensor inversion are described. Then, the details of input parameters and data preparation are presented. Finally, the details of parallelizing synthetic seismogram calculation are provided.

4.1. Moment Tensor Inversion (MTI)

Moment tensor, on a three-dimensional coordinate system, is the matrix form of a dipole force (seismic source) covering all nine possible force configurations. Here, the concept of dipoles force is two forces which are at the equal intensity and opposite direction. These dipole forces in the form of matrices can be demonstrated as follows [32, 33]:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix} \tag{4.1}$$

Depending on seismic moment (M_0) and geographic coordinates, elements of this equation can be written as:

$$\begin{aligned} M_{xx} &= -M_0[\sin(\delta)\cos(\lambda)\sin(2\phi) + \sin(2\delta)\sin(\lambda)\sin^2(\phi)] \\ M_{xy} &= M_0[\sin(\delta)\cos(\lambda)\cos(2\phi) + 0.5\sin(2\delta)\sin(\lambda)\sin(2\phi) \\ M_{xz} &= -M_0[\cos(\delta)\cos(\lambda)\cos(\phi) + \cos(2\delta)\sin(\lambda)\sin(\phi)] \\ M_{yy} &= M_0[\sin(\delta)\cos(\lambda)\sin(2\phi) - \sin(2\delta)\sin(\lambda)\cos^2(\phi)] \\ M_{yx} &= -M_0[\cos(\delta)\cos(\lambda)\sin(\phi) - \cos(2\delta)\sin(\lambda)\cos(\phi)] \end{aligned} \tag{4.2}$$

where δ is dip (the angle between the fault plane and the Earth's surface); ϕ is the strike (the angle between fault direction and geographic north); λ is rake (the angle between slip direction and the fault line).

Determination of the seismic moment tensor components (M_{ij}) will be sufficient to calculate fault parameters (dip, strike and slip). In terms of moment tensor components (M_{ij}), waveform seismic data (S_j), and Green's Functions or components of synthetic seismograms (A_{ij}) the problem can be defined with the following linear equation system:

$$[A_{ij}.S_j] = M_i \tag{4.3}$$

In this equation $i = 1, 2, 3, 4, 5$ and $j = 1, 2, ..., 3n - 1, 3n$ (n is total number of elements of all 3 component seismic data sequences). The solution of this equation system gives us the seismic moment tensor components (M_i) that are directly related to faulting parameters. Therefore, basic structure of FPS application is built on Equation 4.3.

4.2. Input Parameters and Data Preparation

Input data for FPS can be separated into two groups, namely, stored data and calculated data. First of all FPS requires 3D broad-band digital data, station (seismic recorder) information (latitude, longitude and elevation) and instrument response parameters. All these requirements are downloaded from storage server at the beginning of FPS process. Second group of information for the application contains both pre-calculated and manually defined parameters. User must supply previously calculated origin time, local magnitude (M_L) and location (latitude, longitude, depth) of the earthquake. Moreover, distances from the earthquake to stations and azimuth of stations parameters must be known. FPS application determines all these parameters using event location program HYPO71 [34] and some external tools developed as part of FPS application. Another important parameter is the initial crustal model file that contains a 1D velocity structure for working region.

After gathering all necessary information, process begins with waveform data preparation. A sample workflow of whole FPS application process is displayed in Figure 4.1. This pre-process contains three sub procedures:

(i) Removing instrument effects form waveform data using instrument response parameters,
(ii) Rotating EW and NS components of seismogram to radial and transverse components,
(iii) Converting velocity seismograms to displacement seismograms.

Second phase of FPS application involves calculation of synthetic seismograms.

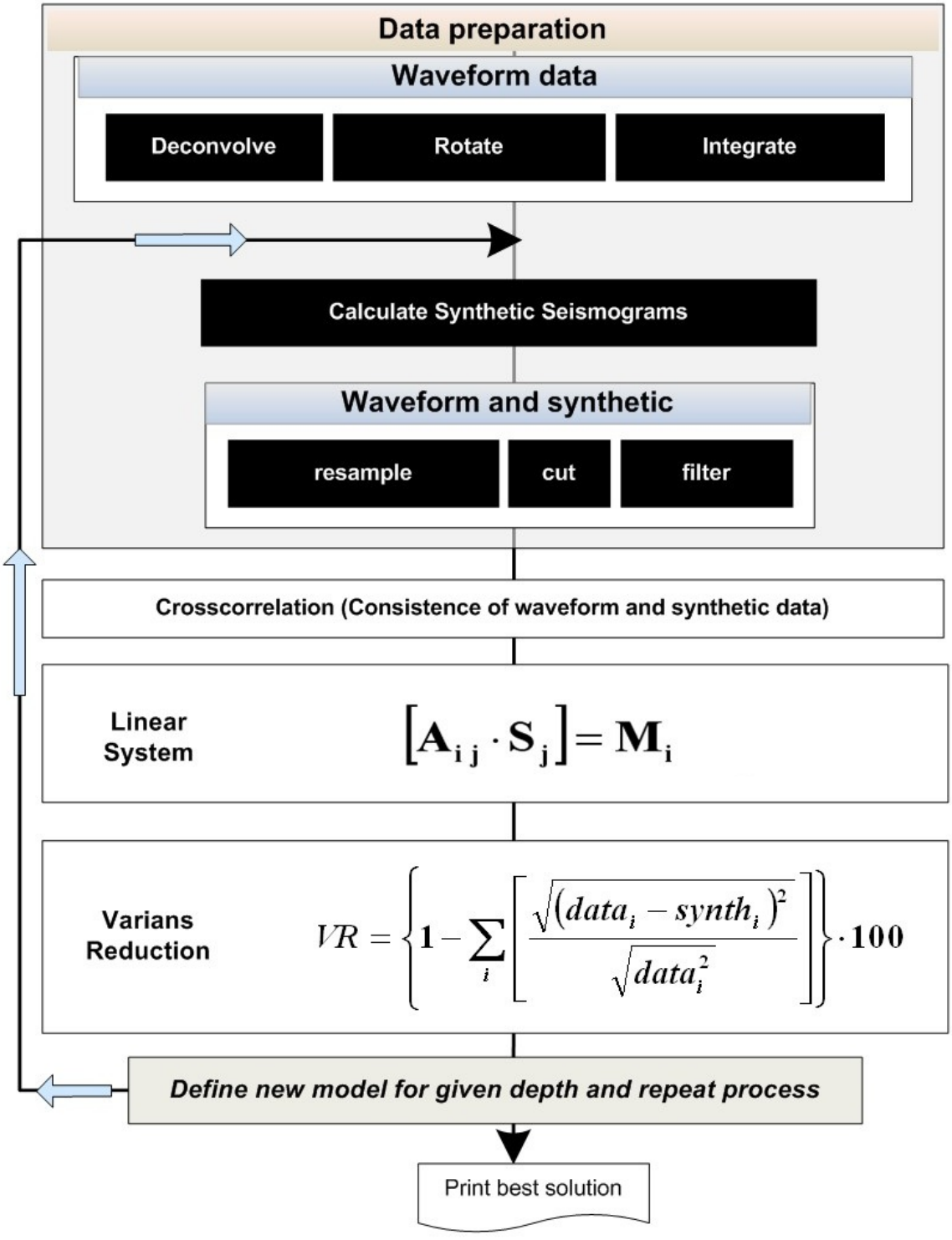


Figure 4.1. Basic schema of FPS workflow on one worker node [35].

As we know, in order to correlate waveform data and synthetic seismograms both waveform data and synthetic must have the same sampling rate and the same length. Also we have to apply the same filter (for regional earthquakes like ours 10-20s) to both data sets.

4.3. Parallelizing Synthetic Seismogram Calculation

FPS application generates crust models having 7 different depths by using the information provided in an initial model file called the Model.ini file and by taking the earthquake parameters determined by the HYPO71 program. Figure 4.2 shows an example of Model.ini file.

```
[MODEL]
nLayer=4
;h(km),Vp(km/s),Vs(km/s),d(g/cm3),Qp,Qs
Layer01=5.4,4.5,3.9,2.5,600,300
Layer02=26.2,5.9,4.3,2.6,1000,500
Layer03=56.6,7.8,5.7,3.0,1000,500
Layer04=400.0,8.3,6.1,3.4,1000,500
...
```

Figure 4.2. Example of Model.ini file (h: depth of layer, Vp: P wave velocity of layer, d: density of layer, Qp and Qs: Q factor for P and S wave respectively).

Kandilli Observatory and Research Institute used to compute Moment Tensor Inverse Solution by using crustal model files that were formed by taking station distances in increments of 5 km. Now, by using the output file of HYPO71 (that contains origin time, event location, station distances and azimuths), actual distances of stations are written to crustal model files. Since the regular 5 km increments have been canceled, run_filter and run_fksort (tools for FPS preprocess) programs that provide the initially prepared data for TDMT_INVC program were also updated for using real distances of stations. An example of crustal model file that is created using Model.ini file and real stations distances is shown in Figure 4.3.

```
.F.
    0   64
GREEN.2.1
    4.0      4.00       1  512 1024    0.500     5    1
    1   1   1   1   1   1   1   1   1   1   0
 0.4000E+01 ... 0.2500E+01   600.00   300.00
 0.1400E+01 ... 0.2500E+01   600.00   300.00
 0.2620E+02 ... 0.2600E+01  1000.00   500.00
 0.5660E+02 ... 0.3000E+01  1000.00   500.00
 0.4000E+03 ... 0.3400E+01  1000.00   500.00
    2
  0.4000000E+03  1.500000E+00          0
    4  10000.0     30.0      1.9       1.7
  269.45      0.0       8.0
  305.83      0.0       8.0
  405.93      0.0       8.0
  422.56      0.0       8.0
```

Figure 4.3. Example file of Crustal Model for synthetic seismogram calculation.

A program called PREPDATA was developed converting waveform synthetic data into SAC format, the format used by the TDMT_INVC program. A JDL workflow which is managed by the Workload Management System (WMS) components was developed for parallelization. Every crustal model file which is automatically generated by FPS and the input files associated with crustal model files are sent to different worker nodes. Therefore, every computationally intensive Green's functions calculation is performed on different worker node. FPS code consists of 7 worker nodes which are not dependent on each other and there is no intermediate communication or data transfer between these separate calculations. So, in order to run FPS application on a parallel grid, no algorithm change is required. Each worker node calculates faulting components by using the Green's functions generated by FPS. The faulting components that have been computed using the worker nodes are sent to a best solution finder worker

node. Solution finder node selects the best solution that yields the largest variance reduction, given in Equation 4.4, according to source depth and sends it to output unit:

$$VC = \left\{ 1 - \sum_{i} \left[\frac{\sqrt{(data_i - synth_i)^2}}{\sqrt{data_i^2}} \right] \right\} .100 \tag{4.4}$$

where VR is variance reduction; data, and synth are the data and Green's function time series, respectively, and the summation is performed for all stations and components.

In order to generate JDL workflows automatically, a program was developed and integrated into FPS. So, in this study, FPS application used to generate and execute the JDL tasks. Figure 4.4 illustrates an example workflow of multiple worker nodes that was designed for automatic JDL generation. A sample JDL workflow file that was generated automatically, based on Figure 4.4, is given in Figure 4.5 and 4.6. The file contains 8 nodes 7 of which are worker nodes and the last one is solution finder node. RunProcess.2.1.sh executed in JDL workflow is a bash script that is responsible for running the processes in order.

Example JDL file which is automatically generated by FPS application includes a group files and every file has its own task in JDL workflow. RunProcess.2.1.sh facilitates the execution of all the tasks in the JDL workflow. The file named 2 that is input for FPS is the output of the HYPO71 application, MODEL.2.1 is a crustal model file, b2s.par is the input of the run_filter script, and MODEL.2.1_mt_inv.in is the input file of TDMT_INVC program. Response_Coordinate.zip file holds coordinate information about the stations and SAC style instrument response (pole-zero) files for the seismic sensors at the stations.

File.zip file holds all executables in it, like TDMT_INVC program, PREPDATA, FKRPROG, WVINT9, run_filter, run_fkrsort, tdmt.config.linux and tdmt_redi_prepdata

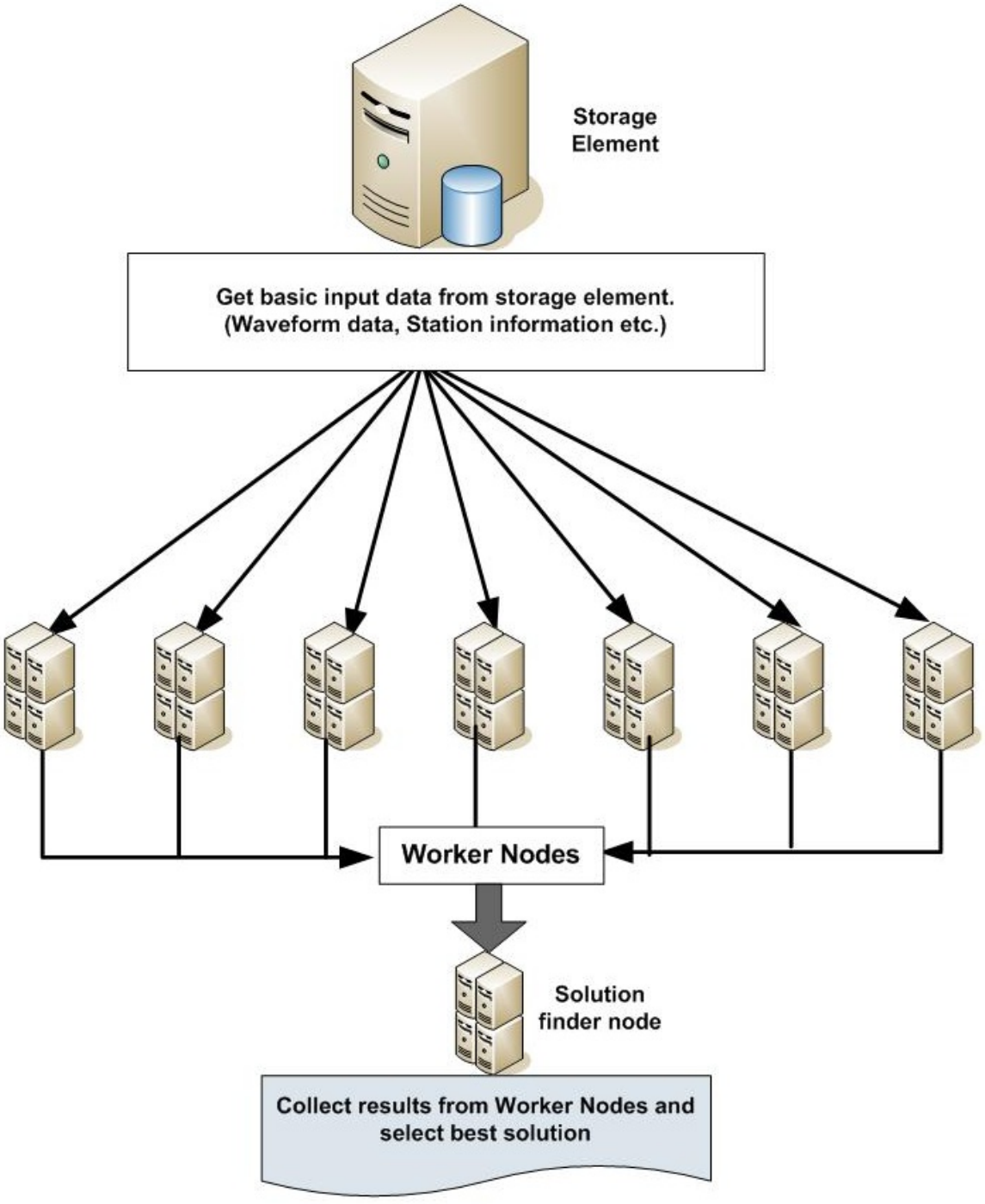


Figure 4.4. FPS workflow on multiple worker nodes.

```
type = "dag";
VirtualOrganisation = "seismo.see-grid-sci.eu";
Rank = other.GlueCEStateFreeCPUs;
max_nodes_running = 8;
InputSandbox = {"FPS.exe","FPS", "Files.zip"};
nodes = [
 part0 = [
  description = [
   Type="Job";
   JobType="Normal";
   Executable = "/bin/sh";
   Arguments = "RunProcess.2.1.sh";
   StdOutput = "out.txt";
   StdError = "err.txt";
   InputSandbox = { "2","MODEL.2.1","MODEL.2.1_mt_inv.in",
                    "RunProcess.2.1.sh","FPS.exe","b2s.par",
                    "Response_Coordinate.zip","Files.zip" };
   OutputSandbox = {"MODEL.2.1.zip","out.txt","err.txt" };
   RetryCount = 3;
   ];
 ];
...
```

Figure 4.5. Example of automatically generated JDL workflow file (part-1).

```
...
  part7 = [
  description = [
   Type="Job";
   JobType="Normal";
   Executable = "FPS";
   Arguments = "4 7 MODEL.2.1 MODEL.2.2 MODEL.2.3 MODEL.2.4
                MODEL.2.5 MODEL.2.6 MODEL.2.7 ";
   StdOutput = "out.txt";
   StdError = "err.txt";
   InputSandbox = { root.InputSandbox,
                    root.nodes.part0.description.OutputSandbox[0],
                    root.nodes.part1.description.OutputSandbox[0],
                    root.nodes.part2.description.OutputSandbox[0],
                    root.nodes.part3.description.OutputSandbox[0],
                    root.nodes.part4.description.OutputSandbox[0],
                    root.nodes.part5.description.OutputSandbox[0],
                    root.nodes.part6.description.OutputSandbox[0]  };
   OutputSandbox = { "BestSolution.zip",
                     "Others.zip","out.txt","err.txt"  };
   RetryCount = 3;
   ];  ]; ];
dependencies = { { {part0, ... , part6 },part7}};
```

Figure 4.6. Example of automatically generated JDL workflow file (part-2).

codes. tdmt.config.linux is responsible for the definition of TDMT_INVC and PREPDATA paths; in addition, it makes LFC catalog definition. FKRPROG program is the main time consuming part of the application that generates frequency domain Green's functions. run_fksort transforms Green's functions frequency domain to time domain using WVINT9 application and creates individual components of synthetic seismograms. PREPDATA is responsible for removing instrument effect, filtering, re-

sampling and rotating of horizontal component of waveform data so that both observed and synthetic are ready for the TDMT_INVC application.

JDL file shown in Figure 4.5 and 4.6 has 7+1=8 worker nodes. JDL file uses a Direct Acyclic Graph (DAG) type workflow and the seismo.see-grid-sci.eu virtual organization. Thanks to RunProcess.2.1.sh, every node knows which files it is going to process. Worker nodes from 0 to 6 independently compute faulting components without having to communicate between separate nodes and then send their results to part7 using feature of DAG. In DAG type workflow, outputs of any node can be input for others. So transferring data between different nodes can be performed easily and automatically. After all the part0 to part6 processes are completed, part7 process finds the best solution which has the biggest variance reduction and transfers the best solution results to the output environment. Dependencies among the processes are specified in the "dependencies" section at the end of the JDL workflow. Here, it is stated that process part7 is dependent on the processes part0,... part6, hence it will be started after all seven worker nodes (part0 to part6) have completed their process.

Additionally, FPS can run on a web based application called P-GRADE Grid portal from SZTAKI, Hungary. This type of running can be seen in Figure 4.7.

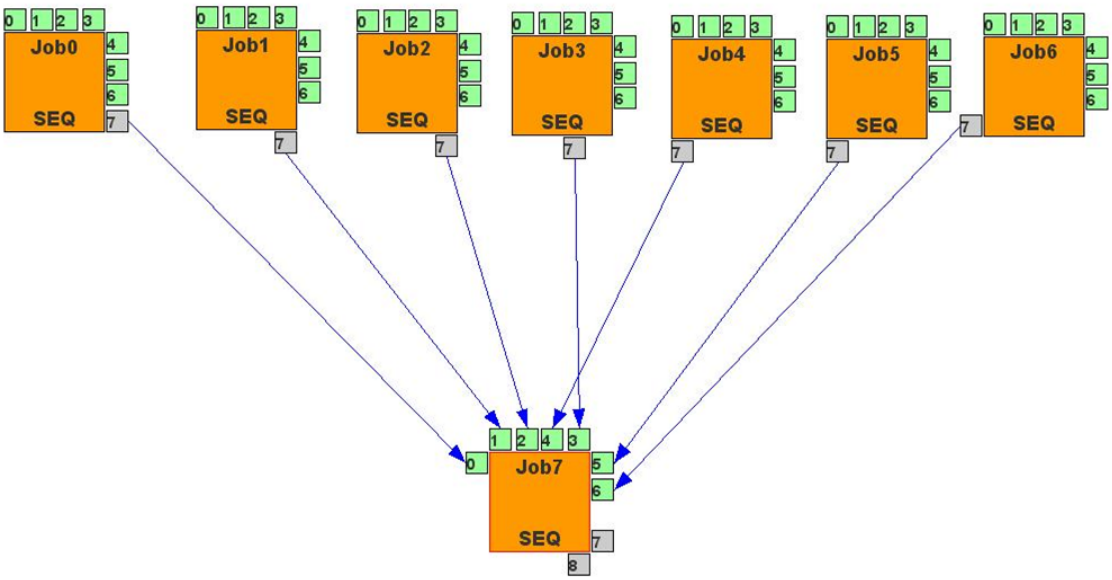


Figure 4.7. Jobs distributions and workflow of FPS on P-GRADE Grid portal.

5. TEST AND RESULTS

5.1. Gridification of Earthquake Location Finding Application

Output of the ELF application can be visualized by using SDSAS application service kml generation routines. These routines will directly read results stored on the AMGA table and generate the kml file. An example kml file is shown in Figure 5.1.

We have also carried out several timing tests to measure the performance of the ELF application. The results obtained are shown in Table 5.1.

Table 5.1. Various tests performed and timings obtained.

Tests	**Timings (secs) without magnitude finding**		**Timings (secs) with magnitude finding**	
	Serial	**Parallel Workflow**	**Serial**	**Parallel Workflow**
20080715T07	226	48	387	147
20080328T00	193	63	567	307
20080803T00	222	57	461	207
20080715T23	225	59	890	391
20080315T12	192	41	596	279
20051017T08	94	32	190	86
20051020T21	88	50	616	422
20051125T09	81	40	335	230
20051126T15	90	37	225	177

Each test is run so that earthquakes within a one-hour period are searched for. Therefore, for example 20080715T07 means the time period 6:00-7:00 on July 15th, 2008. As stated earlier in this thesis, our ELF application is not compute intensive. However, it is data intensive; it needs to read several waveform files. As a result, spatial

(station-wise) parallelization facilitates the reading of the files in parallel. This in turn speeds-up the performance of the program. When we look at speed-up values (i.e. serial time divided by parallel workflow time), we see values ranging from 1.76 to 4.7 in the version of ELF that does not invoke earthquake magnitude finding. Magnitude finding process takes place on one node similar to the node running the HYPO71 program. Therefore, it reduces parallel component of the workflow resulting in less speed-ups. In the table, we see that speed-ups in the range 1.3 to 2.6 are obtained with magnitude finding.

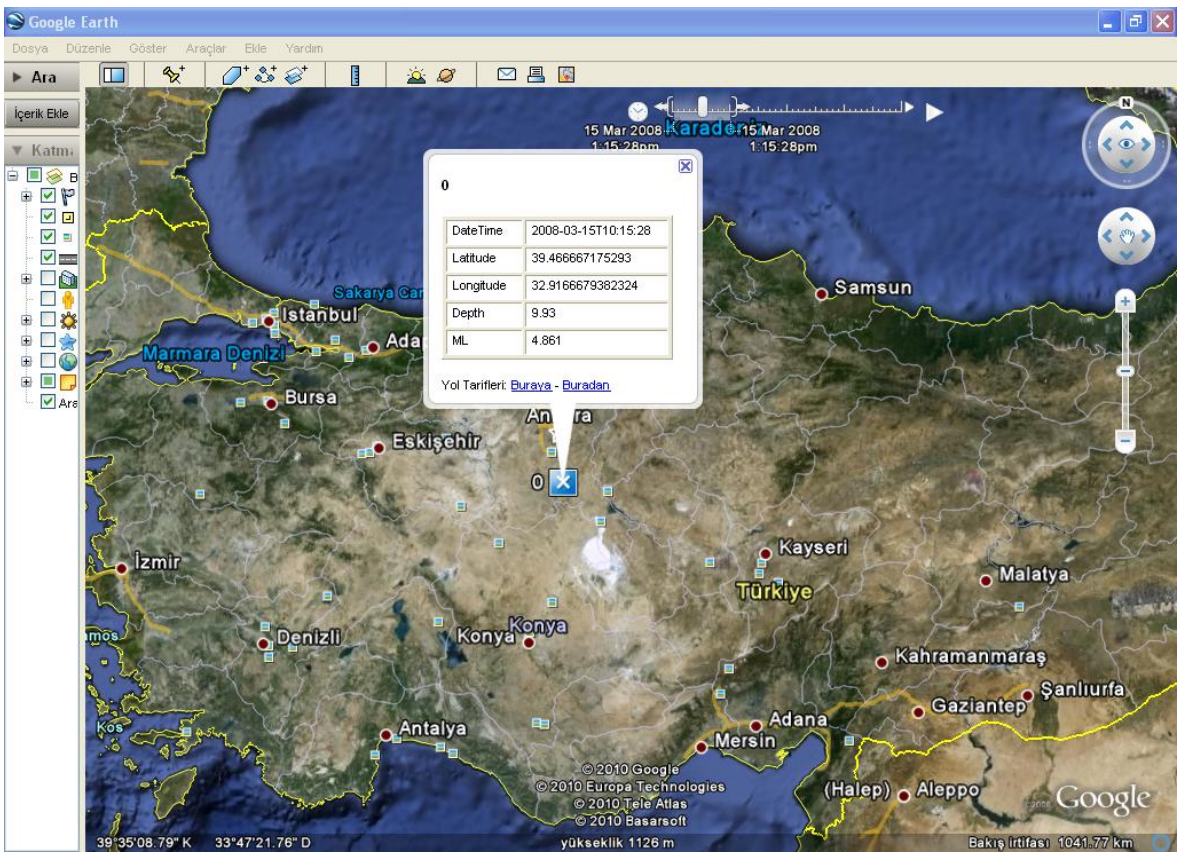


Figure 5.1. KML file generated by SDSAS application service.

5.2. Gridification of Regional Fault Plane Solution Application

In this thesis, we presented details of parallelization of FPS application using JDL workflows. FPS application has been tested on SEEGRID Infrastructure. We did not discuss quantitative results of the application since MTI is a widely used application for obtaining faulting parameters and our focus and contribution in this thesis is its JDL workflow based parallelization. In this work, the most important issue is runtime of the application. Table 5.2 shows the timings obtained from FPS on two different case studies (4 stations and 12 stations). The table shows various statistics such as the number of stations and the number of files used. It also shows the total time taken by the application and how this total time is distributed among individual components. The tests were performed by running the application sequentially and in parallel. The parallel execution was performed in two different platform settings. In the first platform setting (indicated by parallel1) the worker nodes were not physically close to the storage elements on the same site. In the second platform setting (i.e. parallel2), both the workers nodes as well as the storage element were physically close to each other on the same site.

Table 5.2. Test results for FPS application in different conditions (SQ: Sequential, P1: parallel1, P2: parallel2, NoS: number of stations, NoF: number of files).

Type	**NoS**	**NoF**	**AMGA Query (sec)**	**Download time (sec)**	**Computation time (sec)**	**Total Time (sec)**
SQ	4	12	5	165	110	281
P1	4	12	45	162	14	222
P2	4	12	45	91	14	151
SQ	12	36	5	485	215	706
P1	12	36	45	479	21	546
P2	12	36	45	279	21	346

The results show that FPS performance was indeed improved by parallelization. The waveform file handles are obtained by using the SDSAS [25] application service

which in turn queries the AMGA server for this purpose. We have noticed that AMGA performance degrades drastically, especially when multiple concurrent requests are sent to the AMGA server from multiple processes. If the AMGA server had better performance, then FPS application would be much faster.

6. CONCLUSIONS

Earthquake location finding and regional fault plane solution are parallelized by using Job Description Language (JDL) on the grid platform. As can be seen from the results of the test, parallelization has enabled the acceleration of the applications. In the future, applications can be developed on the cloud platform. In order to do that, an infrastructure on the cloud platform should be configured like a grid infrastructure. There should be a user interface through which a user submits his job to the infrastructure, workload management system which manages workflow, worker nodes which perform computation, a storage element which provides data for computation and a database system which contains information about of where data is stored. After this configuration, two parts of the applications should be re-developed according to new environment. These parts are responsible for generating JDL workflow file automatically and downloading correct waveform data files by using a database system in order to do computation. Furthermore, better performance can be obtained by using a different database system such as Oracle, IBM DB2 or Hadoop.

Support for waveform data which is in different format in other countries can be provided by adding the format converters such as from CSS to SAC or from-EVT-to-SAC. Thus, data heterogeneity can be supported for the applications. Also, waveform data from many countries or organizations can be collected in a single center by making agreements among countries or organizations. Therefore, earthquake analysis can be done more quickly and effectively.

APPENDIX A: SEISMIC DATA STORAGE AND ORGANIZATION

Seismic waveform data and station pole zero files supplied by data sources are needed to run ELF, FPS and other applications in the SEE-GRID-SCI Seismology VO. They are stored in storage elements and registered in logical file catalog (LFC). Ulakbim LFC catalog (lfc.ulakbim.gov.tr) is used among LFC catalogs. Seismic Data Server (SDS) uses the following logical file hierarchy shown in Figure A.1.

Earthquake and station data are stored in AMGA tables. Also, metadata about the seismic waveform files are also stored in AMGA tables. The seismic waveform AMGA table acts as an index which keeps track of every single seismic waveform stored in the seismology VO. As a result, the user does not have to go through the slow and cumbersome traversal of the LFC in order to see what files are available in the seismology VO. Queries to AMGA can give these faster. C++ iterators that are provided by SDSAS [25] can do this even at a higher level and can eliminate the need to learn AMGA queries.

In order to use new waveform data files on the applications in SEE-GRID-SCI Seismology VO, insert_waveformdata.sh can be used because this script is used to (i) save seismic waveform data file to the storage element, (ii) register the file in LFC, and (iii) add metadata about the file to an AMGA table. Moreover, this script can insert multiple files which are expected to be located in a directory. Country codes on file names can include two characters or three characters (TR or TUR).

The name of the file containing seismic waveform data must have the format shown in Figure A.2 and waveform data must not cross monthly boundaries.

```
/grid/grid/seismo.see-grid-sci.eu/data/
                                        albania/
                                                 Waveformdata_Years
                                                 Station_PoleZeroFiles/
                                        armenia/
                                                 Waveformdata_Years
                                                 Station_PoleZeroFiles/
                                        macedonia/
                                                 Waveformdata_Years
                                                 Station_PoleZeroFiles/
                                        moldova/
                                                 Waveformdata_Years
                                                 Station_PoleZeroFiles/
                                        turkey/
                                               kandilli/
                                                  Waveformdata_Years
                                                  Station_PoleZeroFiles/
                                    /tools/
                                           scripts/
                                                 insert_waveformdata.sh
                                                 insert_earthquake.sh
```

Figure A.1. Logical file hierarchy.

```
StartDateTime-EndDateT.-Country-Organization-Station-Sensor.FileFormat
```

Figure A.2. The name format of the file containing seismic waveform data.

Seismograph stations have three types of sensors. These are BHE, BHN, and BHZ. The names of the example files containing seismic waveform data can be seen in Figure A.3.

```
20080911T235010-20080912T213020-TR-KAN-IST-BHE.SAC
20090910T225010-20090912T215020-AL--IST-BHN.SAC
```

Figure A.3. The names of the example files containing seismic waveform data.

Before running insert_waveformdata.sh, it should be made sure that .mdclient.config file in Figure A.6 is located in $HOME directory and login name in .mdclient.config file is seismo. Then, you should get a seismo.see-grid-sci.eu proxy by running the command in Figure A.4.

```
$voms-proxy-init --voms seismo.see-grid-sci.eu
```

Figure A.4. Proxy command.

insert_waveformdata.sh command expects the waveform files directory and the storage element on which it will be stored. An example command is given in Figure A.5.

```
$./insert_waveformdata.sh /tmp/waveformdataDirectory/ se.ulakbim.gov.tr
```

Figure A.5. An example usage of insert_waveformdata.sh.

```
host = amga.ulakbim.gov.tr
Port = 8822

# User settings
Login = seismo
PermissionMask = rwx
GroupMask = r-x
Home = /

# Security options
UseSSL = require
AuthenticateWithCertificate = 1 # Use certificate to authenticate
CertFile=$HOME/.globus/usercert.pem
KeyFile=$HOME /.globus/userkey.pem
UseGridProxy = 1
IgnoreCertificateNameMismatch = 1

# If server certificates are verified, CA certificates need to be ...
TrustedCertDir = /etc/grid-security/certificates
```

Figure A.6. .mdclient.config file.

APPENDIX B: USAGE OF ELF APPLICATION

To run the ELF application, four different types of information are needed. These are operation mode, country, organization and date information. The ELF application runs sequentially or in parallel mode according to given operation mode and it uses country, organization and date information in order to send query to the AMGA database through SDSAS. Date information contains year, month, day and hour information and the ELF application gets time range to analyze waveform data. The ELF application is executed as shown in Figure B.1. The second main part of time information that gives time range is optional.

```
$./ELF run_type Country Organization yyyy mm dd hh
$./ELF run_type Country Organization yyyy mm dd hh  [ yyyy mm dd hh ]
```

Figure B.1. Usage type of the ELF application.

The ELF application provides three operating modes. (i) One of them is 1 which means the ELF application runs on sequential mode. (ii) The second one is 5 meaning the ELF application generates JDL file for sequential job on the grid. (iii) The third one is 6 which means the ELF application generates JDL file for station divided parallel jobs.

B.1. Example usage of ELF application

To run the ELF application in parallel mode and get results of the application, a five-step process is performed.

(i) Firstly, user gets a seismo.see-grid-sci.eu proxy by running the command as shown in Figure B.2.

```
$voms-proxy-init --voms seismo.see-grid-sci.eu
```

Figure B.2. Proxy command.

(ii) In order to create JDL file for station divided parallel jobs, the command shown in Figure B.3 is executed. In this command line, 1 is the operation mode of the application, TR is the country abbreviation of Turkey, KAN is the organization abbreviation of Kandilli Observatory and Research Institute, 2008 is the year, 03 is the month, 28 is the day and 01 is the hour.

```
$./ELF 6 TR KAN 2008 03 28 01
```

Figure B.3. Example usage of the ELF application to generate JDL file for station divided parallel jobs.

(iii) After "JDL.file.jdl" file is created by the previous command given in Figure B.3, it is submitted to grid environment by the command shown in Figure B.4. example.id contains submitted job id.

```
$ glite-wms-job-submit -o example.id -a JDL_File.jdl
```

Figure B.4. Job submission to the Grid.

(iv) Job status is controlled through the command shown in Figure B.5.

```
$ glite-wms-job-status -i example.id | more

====================== glite-wms-job-status Success ==================
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms-aegis.ipb.ac.rs:9000/AvDJ-4ZB0N1
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        dagman
...
```

Figure B.5. Controlling job status.

(v) After "current status" of the job is changed to "Done (Success)", the command in Figure B.6 is executed to download output files of the job.

```
$ glite-wms-job-output --dir ./output/ -i example.id
```

Figure B.6. Downloading output files of the job.

The resulting files of the ELF application are located in the last part of output files of the workflow. out.txt contains the location and the magnitude information about earthquakes. In addition, output of the ELF application can be visualized by using SDSAS kml generation routines.

APPENDIX C: USAGE OF FPS APPLICATION

To run FPS application, six different types of information are needed. These are operation mode, initial file name, low frequency, high frequency, second and Model.ini file. The FPS application runs sequentially or in parallel mode according to specified operation mode. Low frequency, high frequency and second are used to filter Green's functions. By using initial file, output file of the HYPO71 application, the FPS application also gets information about an earthquake time and stations which detect the earthquake. By using these information and crustal information in Model.ini file, the FPS application automatically generates crustal model files, necessary for TDMT_INV, and shell script files which will manage the sequence of operations on worker nodes. The FPS application is executed as shown in Figure C.1.

```
$./FPS <RunType> <Initial File Name> <Low Freq.> <High Freq.> <Second>
```

Figure C.1. Usage type of the FPS application.

The FPS application provides two operating modes:

(i) The first one is 1. The FPS application runs on sequential mode.
(ii) The second one is 2. The FPS application generates JDL file and other files necessary for computations.

C.1. Example usage of FPS application

To run FPS application in parallel mode and get results of the application, a five-step process is performed.

(i) Firstly, user gets a seismo.see-grid-sci.eu proxy by running the command in Figure C.2.

```
$voms-proxy-init --voms seismo.see-grid-sci.eu
```

Figure C.2. Proxy command.

(ii) In order to create JDL file, the command in Figure C.3 is executed. In this command line, the first 2 is the operation mode of the application, the second 2 is the output file of HYPO71 application, 0.02 is the low frequency, 0.05 is the high frequency and 120 is the second.

```
$./FPS 2 2 0.02 0.05 120
```

Figure C.3. Example usage of the FPS application to generate JDL file.

(iii) After "JDL.file.jdl" file is created by the previous command in Figure C.3. It is submitted to grid environment by the command in Figure C.4. example.id contains submitted job id.

```
$ glite-wms-job-submit -o example.id -a JDL_File.jdl
```

Figure C.4. Job submission to the Grid.

(iv) Job status is controlled through the command given in Figure C.5.

```
$ glite-wms-job-status -i example.id | more

====================== glite-wms-job-status Success ==================
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms-aegis.ipb.ac.rs:9000/ByEK-1CE3B6
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        dagman
...
```

Figure C.5. Controlling job status.

(v) After "current status" of the job is changed to "Done (Success)", the command in Figure C.6 is executed to download output files of the job.

```
$ glite-wms-job-output --dir ./output/ -i example.id
```

Figure C.6. Downloading output files of the job.

The resulting files of the FPS application are located in the output directory after the command in Figure C.6. BestSolution.zip contains the best solution among fault plane solutions and Others.zip contains the other solutions.

REFERENCES

1. *South Eastern European Grid e-Infrastructure for Regional e-Science (SEE-GRID-SCI)*, `http://www.see-grid-sci.eu`, accessed at May 2012.

2. SEE-GRID-SCI consortium, *Deliverable JRA1.4 - Application Services Implementation and Deployment Report*, pp. 23–73, 2010.

3. Özturan, C., V. Kotroni and E. Atanassov, "Development of Virtual Organizations, Applications and Services for Earth Science on Grid E-Infrastructures", *Earth Science Informatics*, Vol. 3, No. 4, pp. 197–198, 2010.

4. *SEE-GRID-SCI Seismology VO*, `http://wiki.egee-see.org/index.php/SG_Seismology_VO`, accessed at May 2012.

5. Lee, W. H. K. and J. C. Lahr, "HYPO71: A Computer Program for Determining Hypocenter, Magnitude, and First Motion Pattern of Local Earthquakes", *Open File Report, U. S. Geological Survey*, 1972.

6. Doser, D. I., "Estimating Magnitude and Location of Alaskan Earthquakes Using Intensity Data", *Bulletin of the Seismological Society of America*, Vol. 99, No. 6, pp. 3430–3453, 2009.

7. Bakun, W. H., "Magnitude and Location of Historical Earthquakes in Japan and Implications for The 1855 Ansei Edo Earthquake", *Journal of Geophysical Research*, Vol. 110, No. B02304, p. 22, 2005.

8. Bakun, W. H., A. C. Johnston and M. G. Hopper, "Estimating Locations and Magnitudes of Earthquakes in Eastern North America from Modified Mercalli Intensities", *Bulletin of the Seismological Society of America*, Vol. 93, pp. 190–202, 2003.

9. Ichinose, G. A., J. G. Anderson, K. D. Smith and Y. Zeng, "Source Parameters of Eastern California and Western Nevada Earthquakes from Regional Moment Tensor Inversion", *Bulletin of the Seismological Society of America*, Vol. 93, No. 1, pp. 61–84, 2003.

10. Stich, D., J. G. Anderson, R. Martín and J. Morales, "Moment Tensor Inversion for Iberia–Maghreb Earthquakes 2005–2008", *Tectonophysics*, Vol. 483, pp. 390–398, 2010.

11. Randall, G., "Efficient Calculation of Complete Differential Seismograms for Laterally Homogeneous Earth Models", *Geophysical Journal International*, Vol. 118, pp. 245–254, 1994.

12. Kikuchi, M. and H. Kanamori, *Note on Teleseismic Body-Wave Inversion Program*, 2003, `http://www.eri.u-tokyo.ac.jp/ETAL/KIKUCH/manual.doc`, accessed at June 2012.

13. Dreger, D., *Time-Domain Moment Ternsor Inverse Code (TDMT_INVC) Release 1.1*, 2002, `http://www.seismo.berkeley.edu/~dreger`, accessed at June 2012.

14. Suárez, G., P. Molnar and B. C. Burchfiel, "Seismicity, Fault Plane Solutions, Depth of Faulting, and Active Tectonics of The Andes of Peru, Ecuador, and Southern Colombia", *Journal Of Geophysical Research*, Vol. 88, No. B12, pp. 10403–10428, 1983.

15. Forsyth, D. W., "Fault Plane Solutions and Tectonics of the South Atlantic and Scotia Sea", *Journal Of Geophysical Research*, Vol. 80, No. 11, pp. 1429–1443, 1975.

16. Papazachos, B. C., E. E. Papadimitriou, A. A. Kiratzi, C. B. Papazachos and E. K. Louvari, "Fault Plane Solutions in the Aegean Sea and the Surrounding Area and Their Tectonic Implication", *Bollettino di Geofisica Teorica ed Applicata*, Vol. 39, No. 3, pp. 199–218, 1998.

17. Hinzen, K.-G., "Stress Field in the Northern Rhine Area, Central Europe, from Earthquake Fault Plane Solutions", *Tectonophysics*, Vol. 377, pp. 325–356, 2003.

18. Clévédé, E., D. Weissenbach and B. Gotab, "Distributed Jobs on EGEE Grid Infrastructure for an Earth Science Application: Moment Tensor Computation at the Centroid of an Earthquake", *Earth Sci Inform 2*, pp. 97–106, 2009.

19. Şener, C., A. Akkaya, C. Kocair and E. Ülgen, "Seismic Risk Assessment", *SEE-GRID-SCI User Forum 2009*, pp. 29–34, 2009.

20. Kozlovszky, M., Ákos Balaskó and B. Süle, "Numerical Modeling of Mantle Convection in 3D on the SEE-GRID-SCI Infrastructure", *Earth Science Informatics*, Vol. 3, No. 4, pp. 229–137, 2010.

21. Allen, R., "Automatic Phase Pickers: Their Present Use and Future Prospects", *Bulletin of the Seismological Society of America*, Vol. 72, No. 6b, pp. 225–242, 1982.

22. Şenay, E. and A. S. Balkır, *Kandilli Earthquake Seismic Data Server and Analysis*, Undergraduate Thesis Report, Boğaziçi University, 2007.

23. Arıkan, R. and M. S. Geden, *Kandilli Earthquake Seismic Data Server and Analysis*, Undergraduate Thesis Report, Boğaziçi University, 2008.

24. MSEED2SAC, *Incorporated Research Institutions for Seismology*, `http://www.iris.edu/software/downloads/conversion/`, accessed at August 2012.

25. Özturan, C., B. Bektas and M. Yilmazer, "Seismic Data Server Application Service for SEEGRID Seismology Virtual Organization", *Earth Science Informatics*, Vol. 3, No. 4, pp. 219–228, 2010.

26. *AMGA: Arda Metadata Catalogue Project*, `http://amga.web.cern.ch/amga`, ac-

cessed at June 2012.

27. Haskell, N., "Radiation Pattern of Surface Waves From Point Sources in a Multi-Layered Medium", *Bulletin of the Seismological Society of America*, Vol. 54, No. 1, pp. 377–393, 1964.

28. Saikia, C., "Modified Frequency-Wavenumber Algorithm for Regional Seismograms Using Filon's Quadrature; Modeling of Lg Waves in Eastern North America", *Geophysical Journal International*, Vol. 118, pp. 142–158, 1994.

29. Dreger, D. and B. Woods, "Regional Distance Seismic Moment Tensors of Nuclear Explosions", *Tectonophysics*, Vol. 356, pp. 139–156, 2002.

30. Dreger, D., B. Romanowic, G. Clitheroe, P. Hellweg and J. Stevens, "Development of Automated Moment Tensor Software at the Prototype International Data Center", *23rd Seismic Research Review: Worldwide Monitoring of Nuclear Explosions*, 2001.

31. Goldstein, P., *Seismic Analysis Code SAC2000*, 2008, `http://www-ep.es.llnl.gov/www-ep/esd/seismic/sac.html`, accessed at Aug 2012.

32. Herrmann, R. and C. Wang, "A Comparison of Synthetic Seismograms", *Bulletin of the Seismological Society of America*, Vol. 75, No. 1, pp. 41–56, 1985.

33. Jost, M. and R. Herrmann, "A Student's Guide to and Review of Moment Tensors", *Seismological Research Letters*, Vol. 60, pp. 37–57, 1989.

34. Lee, W. and J. Lahr, "HYPO71: A Computer Program for Determining Hypocenter, Magnitude, and First Motion Pattern of Local Earthquakes", *Open File Report, U. S. Geological Survey*, p. 100, 1972.

35. Yılmazer, M., *Determination of Faulting Mechanism of Earthquakes using 3 Com-*

ponent Strong Motion and Broadband Records and their Seismotectonic Implication, Ph.D. Thesis, İstanbul University, 2009.