Enabling of Grid based Diffusion Tensor Imaging using a Workflow Implementation of FSL

Healthgrid 2009, Berlin 30.06.2009

Ralf Lützkendorf
Johannes Bernarding
Frank Hertel
Fred Viezens
Andreas Thiel
Dagmar Krefting
Diffusion Tensor Imaging

- is based on diffusion weighted MRI
- allows for tracking of nerve fibers

Main applications

- Fundamental research
  - Anatomy and formation of axonal tracts
  - Connectivity between functional areas
- Clinical research
  - Recognition of disease patterns
  - Planning support in brain tumor therapy
Image Processing

Different Software Tools available for DTI
FMRIB’s Diffusion Toolbox:
  • Graphical user interfaces
  • Command line tool

Image processing pipeline:
  • Preprocessing
    • Distortion correction
    • Segmentation
  • Local modeling of diffusion parameters (bedpostX)
    • Estimation of diffusion tensor for each voxel by Markov Chain Monte Carlo sampling
    • Computationally expensive
    • Parallelizable on slice level
  • Fiber tractography
Runtime

- Website information: 24 h
- Personal computer (512 Mb RAM, 3 GHz, 56 slices): 10 days!
- Runtime depends on slicenumber and content!

First approach:
- Local windows desktop grid of 10 computers:
  - using cygwin
  - Scales nearly linearly
  - Runtime ~ 24 hrs
  - Only temporarily available

Grid approach:
- Usage of powerful machines
- Parallel processing of MRI slices
Data Management

Requirements:
- Fixed number of dedicated files provided within a directory
- Generated by preprocessing
- Image Format: NIFTI

SDSC's Storage Resource broker:
- Linux-like file system structure
- Shared access using group rights

Stored data:
- Input data
- Intermediate results
- Final results
Process Management

Processing steps:
• Splitting the data into slices
• Run bedpostX
• Recombine data

Grid Workflow Execution Service:
• Orchestration of the processing steps
• Scheduling and resource brokering
• Error handling
• Reusable workflow templates
Workflow Layout

Monolithic implementation

Parallelized implementation
System architecture

Main components used for application

- All middleware components are free software for academic use
## Results

Runtime measurement:

<table>
<thead>
<tr>
<th></th>
<th># Slices</th>
<th>Monolithic hrs</th>
<th>Parallelized hrs</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grid</strong></td>
<td>11</td>
<td>6.5±0.7 (0.7)</td>
<td>3.5±2.5 (0.3)</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>14.3±0.8 (0.2)</td>
<td>3.5±1.7 (0.25)</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>8.9±3.8 (0.4)</td>
<td>3.6±1.2 (0.18)</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>22.0±8 (0.4)</td>
<td>3.8±2.2 (0.06)</td>
<td>5.6</td>
</tr>
<tr>
<td><strong>Local Cluster</strong></td>
<td>56</td>
<td>240.0 (4.2)</td>
<td>23 (0.4)</td>
<td>10.4</td>
</tr>
</tbody>
</table>
Results

Reliability:
- Complete monolithic grid implementation
- Single slice monolithic grid implementation
- Parallelized grid implementation
- Five subsequent monolithic grid implementation, tested on different grid sites
- Single slice monolithic local implementation on 64bit machine
- Single slice monolithic local implementation on 64bit machine
Results

Differences between 64bit and 32bit machine
Results

Accessibility
• Webbased up- and download to SRB
• Webbased upload, start and control of the workflow
• Data and application has been shared between Magdeburg and Berlin

Usability:
• BIRN’s SRB portlet
• GWES workflow portlets
• Workflow template (xml-file)
• No application specific gui
Discussion and Outlook

- Speedup by more powerful computing resources
- Speedup by parallelization
- Fault tolerance essential (poor SRB scalability, network)
- Speedup drastically reduced by queuing times
- Reliability given for current grid infrastructure
- Accessibility and usability sufficient for researcher

Further developments
- Completion of the processing on the grid
- Provision of a portlet
- Usage of more scalable GDMS (iRods ?)
- Process optimization (dynamic bundling of slices)
System architecture

User: medical doctor or scientist in medical sciences

- Clinical environment: strict data and network protection
- Not necessarily a computer expert
System architecture

Grid portal: gridsphere based web based access

- Accounts for strict firewall settings (plain http/httpss access)
- Grid access from “everywhere” (with internet connection…)
System architecture

Data storage: Storage Resource Broker

- Unix-like access rights (basic data protection)
- Metadata Management
System architecture

Grid nodes: Globus Toolkit 4

- Credential based security
- WS-GRAM generic webservice for program execution
Workflow Manager: Grid Workflow Execution Service (GWES)

- Data driven workflow modeling as high level Petri net
- Resource matching, basic brokering, error handling

Clinic: User

Workflow Execution Service
Univ. of Leipzig

Web-Portal
Uni Leipzig

www.medigrid.de

Grid-Node
GWDG

Grid-Node
Univ. of Leipzig

Data Storage and Metadata
Berlin, Dresden, Göttingen

Grid-Node
Zuse Inst. Berlin

Grid-Node
Univ. of Dresden

System architecture
Main tasks for the developer:

- Register the software components to the WF-DB
- Provide a workflow description (upload or template)
Storage of the SIESTA data in the SRB
- Preservation of the dedicated folder structure
- Group account for shared access
- Relevant metadata is stored as user defined Metadata (*Sufmeta*)

Integration of the processing tools
- Plain command line tools
- Execution by invocation of WS-GRAM automatically called by the GWES

Data transfer
- Srb-transfer protocol between SRB and gridnodes
- reliable file transfer (RFT) between nodes

<table>
<thead>
<tr>
<th>ECG</th>
<th>ECG presence</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODI</td>
<td>Oxygen Desaturation Index</td>
</tr>
<tr>
<td>TST</td>
<td>Total Sleep Time [min]</td>
</tr>
<tr>
<td>AI</td>
<td>Apnea Index [/min]</td>
</tr>
<tr>
<td>AHI</td>
<td>Apnea Hypopnea Index [/min]</td>
</tr>
<tr>
<td>Age</td>
<td>Age [years]</td>
</tr>
<tr>
<td>Sex</td>
<td>Sex [male=1,female=2]</td>
</tr>
<tr>
<td>Status</td>
<td>Health Status [tag]</td>
</tr>
<tr>
<td>Height</td>
<td>Height [cm]</td>
</tr>
<tr>
<td>Weight</td>
<td>Weight [kg]</td>
</tr>
<tr>
<td>Pulse</td>
<td>Pulse rate [/min]</td>
</tr>
<tr>
<td>BPsys</td>
<td>Blood press. syst. [mmHg]</td>
</tr>
<tr>
<td>BPdia</td>
<td>Blood press. Diast. [mmHg]</td>
</tr>
</tbody>
</table>
Sleep-phase resolved heart frequency for study samples, determined by different indicators (age, health status,..)

1. Record selection: Metadata query
2. Record analysis
   • Heart frequency (HF) analysis on the selected samples
   • Matching of HF and consensed sleep states
   • Statistical analysis of individual HF patterns
3. Collection analysis
   • Statistical analysis on sample set
Petri net modeling

Petri net, modelling the application:

Places & Transition Tokens

Query 1
Query 2
Query 3
Query 4
Query 5

Record Selection

T1

Record 1
Record 2
Record n

T2

Result 1
Result 2
Result n

T2

Record Analysis

T3

Collection Analysis

Final Result
Problem:
Special characters or blanks in token values
-> don’t pass complete srb queries
Portal integration

Integration of a workflow template into an applicationspecific portlet:
Portlet development

MediGRID Portal

Sleep Medicine: Analysis of Polysomnographies

Interdisciplinary Sleep Disorders Centre Hessen

Grid Integration: Charité Universitätsmedizin Berlin

<table>
<thead>
<tr>
<th>Description</th>
<th>Status</th>
<th>Begin</th>
<th>End</th>
<th>Level</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinical Research</td>
<td>COMPLETED</td>
<td>May 16 10:48:39</td>
<td>May 16 10:48:39</td>
<td>MEMORY</td>
<td>Show Results</td>
</tr>
<tr>
<td>Clinical Research</td>
<td>COMPLETED</td>
<td>May 16 10:48:39</td>
<td>May 16 10:48:39</td>
<td>MEMORY</td>
<td>Show Results</td>
</tr>
<tr>
<td>Clinical Research</td>
<td>COMPLETED</td>
<td>May 16 10:44:53</td>
<td>May 16 10:44:53</td>
<td>MEMORY</td>
<td>Show Results</td>
</tr>
</tbody>
</table>
Portlet development

Gridsphere Framework
- Gridportlets
- Predefined Components
- JSR168 compliance

MediGRID Portal
- SRB Portlet (from BIRN)
- Workflow GUI Portlets

Application specific Portlet:
- Workflow list and status
- Parameter setting
- Workflow initialization
- Result visualization
Developer usecase

Application of the tools on heterogeneous longterm measurements reveals significant flaws in the heart-beat detection

Healthy subjects

Apnea patients

sqrs

sqrs + correction

shortterm-FFT

Current usage of the grid: Algorithm testing and development

-> parameter scans and evaluation on the datasets
The application runs on a production grid infrastructure and is usable for all MediGRID participants.

A developer can:
- test deployed algorithms for different groups/setups (portlet)
- perform Parameterscans on the deployed algorithms (workflow)
- test algorithms or modifications (deployment to gridnodes)

Main flaws:
- Metadata handling -> switching to iRODS
- no manual ECG annotation available for SIESTA data
  -> creation of ECG annotation
  -> integration of physiobank databases with annotation
Discussion and outlook

The application runs on a production grid infrastructure and is usable for all MediGRID participants

A clinical researcher can:
- perform statistical analysis on SIESTA data with deployed tools (portlet)
- upload and analyse further (pseudonymized) polysomnographies (portlet)

Main flaws:
- Portal stability -> switching to LiveRay
- robustness of the algorithms -> developer’s task
- Userfriendly pseudonymization and upload tool
Discussion and outlook

The application runs on a production grid infrastructure and is usable for all MediGRID participants.

A physician can:
- upload and analyse (pseudonymized) polysomnographies (portlet)

Main flaws:
- Approvement and Validation of the tools -> clinical researcher
- Grid access - > simplified user registration
- Validation of the infrastructure -> implementation of GCP-rules
- Pricing models and accounting -> grid business models
Thanks to all collaborators

Sebastian Canisius  
Univ. Marburg  
Sleep Center

Andreas Hoheisel  
Fraunhofer FIRST

Thomas Tolxdorff  
Charité  
Medical Inform.

Thomas Penzel  
Charité  
Sleep Center
Petri Nets
Petri nets

Mathematical modeling language for distributed systems, consisting of

- Transitions (squares)
- Places (circles), that may hold $n_p$ tokens (black dots)
- Flow relations (arrows between places and transitions)
  - Input place: arrow is pointing from place to transition
  - Output place: arrow is pointing from transition to place
- Marking: Distribution of tokens on places

![Petri net diagram with places, transitions, and tokens]
Petri nets

- Enabling of a transition:
  - All input places are occupied
  - All output places may receive further tokens
- Firing of a transition:
  - One token of each input place is consumed
  - One token is added to each output place
- Modeling of image processing workflows
  - Data -> token, executables -> transitions
  - Program execution -> firing
Implementation of command line tools
Implementation steps

Implementation of command-line tools to the grid

1. Deployment of the software to the gridnodes
2. Generation of a wrapper script
3. Registration of the software
4. Creation of a workflow description
5. Optional: Integration of the workflow into the user portal
Software has to be installed on the front-end of the sites
- Each application group has its own remote directory
  - Copy application from a local directory to the remote installation directory with gsiscp (script)
  - Access to the gridnodes via gsissh and svn update
A shell-script

• Sets environment (pathes, environment variables)
• Calls the program(s)
• Requirement: all parameters have to be passed as name/value pair
  • Program call:
    segmentation 51123_1100.png 51123_roi.mat
  • Script call:
    gwes-segmentation-simple.sh
    –input_image 51123_1100.png –roi 51123_roi.mat
Database-entry (exIST-database, dgrdl):
  • new software (path of the script)
  • gridnodes where the software is available
Workflow Description

Xml-based GWorkflowDL

gwes-segmentation-simple.sh

`-input_image 51123_1100.png -roi 51123_roi.mat`

```
  <description>Segmentation workflow</description>
  <property name="resource.repository.collection">
    <place id="input_image">
      <token data xmlns="" value xsi:type="xsd:string">51123_1100.png</value>
    </place>
    <place id="roiMat">
      <token data xmlns="" value xsi:type="xsd:string">51123_roi.mat</value>
    </place>
  </property>
  <transition id="segmentation">
    <description>biopsy needle segmentation</description>
    <input place id="input_image" edgeExpression="input_image"/>
    <input place id="roiMat" edgeExpression="roi"/>
    <output place id="output" edgeExpression="result"/>
  </transition>
</workflow>
```
Further Projects
Results

Currently implemented:
- 5 image- and signalprocessing applications
- With application specific portlets:
  - Functional MRI: simple workflow (needs matlab)
  - Virtual vascular surgery: basic interactive visualization
  - Ultrasound imaging: 4 different workflows
  - MRE brain segmentation (FSL sienax)
- Without portlets:
  - Analysis of polysomnographic signals from a clinical study
  - DTI analysis with FSL

- Prospected: dynamical lung imaging (PneumGRID)
Additional slides

MediGRID Portal
Requirements of Medical Grids
Medical Grids
demand special requirements with respect to mere computing Grids:

- High security and safety
  - Patient data, traceability of processing steps

- User friendliness
  - User accustomed to graphical user interfaces

- Virtualization of grid resources
  - Heterogeneous data and applications

Current research on modern Grids is working to overcome these barriers.