Scalable Parallel Delaunay Image-to-Mesh Conversion for Shared and Distributed Memory Architectures

Daming Feng
Advisors: Dr. Nikos Chrisochoides
Dr. Andrey Chernikov
Department of Computer Science, Old Dominion University
Center for Real-time Computing (https://crtc.cs.odu.edu/)

This work in part is funded by NSF grant no. CCF-1439079, NASA grant no. NNX15AU39A and DoD’s PETTT Special Project PP-CFD-KY07-007. In addition, it utilized resources from the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. Finally, it is partially supported by the Richard T. Cheng Endowment.
Contents

• Motivations
• Contributions
• Parallel Delaunay Mesh Generation
• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)
• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchanges (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)
• Following Work, Time Table and Conclusion
Motivation

Application:

• Finite element modeling
  • Image guided therapy
  • Patient-specific interactive surgery simulation

Image-to-mesh conversion requirement

• Quality: radius-to-edge ratio, dihedral angles, size
• Fidelity: mesh surface should be a correct geometric representation of object boundary
• Scalability: the ability of a parallel algorithm to achieve a speedup proportional to the number of cores.

Midwest Medico-Legal Associates.

http://ireteth.certh.gr/mechatronics/biomechanics/
Three Parallel Meshing Algorithms

Proposed Parallel Delaunay Image-to-Mesh Algorithms:

• LAPD:
  • Employs a data locality-aware parallel mesh refinement procedure to reduce the communication overhead caused by the remote memory accesses

• PDR.PODM:
  • Combines the best features of two previous algorithms

• Hybrid MPI+Threads Meshing:
  • Inter-node Communication and Data Exchange (MPI)
  • Intra-node Parallel Mesh Refinement (Pthread)
Contents

• Motivations
• Contributions

• Parallel Delaunay Mesh Generation

• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)

• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchangen (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)

• Following Work, Time Table and Conclusion
Delaunay Mesh Generation

- Delaunay Mesh Generation creates mesh conforming to Delaunay property, i.e., in a Delaunay mesh, no point (vertex) in the mesh is inside the circumcircle (circumsphere) of any triangle (tetrahedron).

https://www.cs.cmu.edu/~quake/triangle.html

PDR.PODM Delaunay Mesh Example
(Parallel) Delaunay Mesh Refinement

point is inserted

concurrent insertions
Contents

• Motivations
• Contributions
• Parallel Delaunay Mesh Generation
• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)
• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchange (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)
• Following Work, Time Table and Conclusion
Remote Memory Access

Distributed Shared Memory:

- Distributed Shared Memory (DSM) is a form of memory architecture where the (physically separate) memories can be addressed as one (logically shared) address space.
- Local Memory Access vs Remote Memory Access

PODM:

- Performance deterioration
- Reasons: remote memory accesses and network congestion
Locality Aware Parallel Delaunay Mesh Generation Algorithm (LAPD)
Distributed Shared Memory Architecture

A reservation of 64 cores (4 blades).
Load Balancing

• Over decomposed Block Partition

Not a perfect scheme, but solve the problem.
LAPD Speedup and Overhead
Example Meshes of LAPD

CT Head-neck atlas
Input: Courtesy of SPL

Complex geometries

Big Brain
Input: Courtesy of BigBrain project
Contents

• Motivations
• Contributions

• Parallel Delaunay Mesh Generation

• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)

• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchange (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)

• Following Work, Time Table and Conclusion
Parallel Delaunay Refinement (PDR)

Concurrent points insertion between two regions

Delaunay-independency of two inserted points

Parallel points insertion results in non-conformal mesh

PDR.PODM Algorithm

(a) Parallel coarse mesh construction

(b) Coarse mesh distribution

(c) PDR.PODM parallel refinement

(d) Final Mesh
Speedup and Meshing Time Comparison

Speedup Comparison

Meshing time of PODM and PDR.PODM
Example Meshes of PDR.PODM
Contents

• Motivations
• Contributions
• Parallel Delaunay Mesh Generation
• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)
• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchanges (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)
• Following Work, Time Table and Conclusion
Hybrid MPI+Threads Meshing Algorithm

• Hybrid Parallel I2M Conversion Algorithm
  • Hybrid MPI+Threads I2M conversion algorithm which takes 3D multi-labeled images as input.
  • The algorithm exploits two levels of parallelization: process level parallelization (which is mapped to a node with multiple cores) and thread level parallelization (each thread is mapped to a single core in a node).
  • It scales well on distributed memory clusters.

• Nested Master-Worker Model:
  • separates and overlaps communication with computation.
MPI+Threads Parallel I2M Conversion

- Nested master-worker model and two level parallelization
  - The master process running on a node creates the coarse mesh, manages and schedules the tasks (subregions) and the worker processes running on other nodes communicate with each other for task request and data migration.
  - Inside each node, the master thread creates multiple worker threads and each of the worker thread runs on one core.

- Overlapping communication with computation
  - The master thread initializes the MPI environment and communicates with the master thread of other processes (communication).
  - The worker threads of each process do not make MPI calls and are only responsible for the local mesh refinement work in the shared memory of each node (computation).
Two-Level Buffer Zones

Two-level buffer zones to avoid the synchronization

Two-level neighbors

- First level buffer:
  - The submeshes inside first level neighbors need to be moved to the local memory of working process.

- Second level buffer:
  - Select and schedule independent subregions to multiple processes to avoid synchronization during refinement.

- Integer Flag
  - The process rank (node ID): where the submesh inside the subregion is stored.

Synchronization is needed in the overlap buffer zones
Inter-node Communication and Data Exchange (MPI)

Task: subregion
Data: submesh

Master thread: communication
Worker threads: computation

Worker Process $P_i$

Master Process

feedback
A task (neighbors)

Task request

Data request

Data

Worker Processes
$P_j, P_k, P_L$
Inter-node Communication and Data Exchange (MPI)

Task: subregion
data: submesh

Worker Process $P_i$

Master Process

A task (neighbors)

feedback

Data request

Data

Worker Processes

$P_j, P_k, P_L$

Pseudo-code
<table>
<thead>
<tr>
<th># Cores</th>
<th># Elements (million)</th>
<th>Running Time (s)</th>
<th>Elements/s (m/s)</th>
<th>Speedup</th>
<th>Efficiency</th>
<th>depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.63</td>
<td>65.13</td>
<td>0.10</td>
<td>1.0</td>
<td>100.00%</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>133.87</td>
<td>76.33</td>
<td>1.75</td>
<td>17.21</td>
<td>86.03%</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>261.56</td>
<td>103.64</td>
<td>2.52</td>
<td>24.76</td>
<td>61.90%</td>
<td>3</td>
</tr>
<tr>
<td>80</td>
<td>520.30</td>
<td>119.88</td>
<td>4.34</td>
<td>42.58</td>
<td>53.22%</td>
<td>3</td>
</tr>
<tr>
<td>120</td>
<td>780.99</td>
<td>133.72</td>
<td>5.84</td>
<td>57.30</td>
<td>47.75%</td>
<td>4</td>
</tr>
<tr>
<td>160</td>
<td>1034.82</td>
<td>126.44</td>
<td>8.18</td>
<td>80.29</td>
<td>50.18%</td>
<td>4</td>
</tr>
<tr>
<td>200</td>
<td>1292.20</td>
<td>119.05</td>
<td>10.85</td>
<td>106.49</td>
<td>53.24%</td>
<td>4</td>
</tr>
<tr>
<td>300</td>
<td>1937.04</td>
<td>111.57</td>
<td>17.36</td>
<td>170.33</td>
<td>56.78%</td>
<td>4</td>
</tr>
<tr>
<td>400</td>
<td>2577.94</td>
<td>107.19</td>
<td>24.05</td>
<td>235.95</td>
<td>58.99%</td>
<td>4</td>
</tr>
<tr>
<td>500</td>
<td>3217.94</td>
<td>109.02</td>
<td>29.51</td>
<td>289.57</td>
<td>57.91%</td>
<td>4</td>
</tr>
<tr>
<td>600</td>
<td>3858.97</td>
<td>109.22</td>
<td>35.33</td>
<td>346.61</td>
<td>57.77%</td>
<td>4</td>
</tr>
<tr>
<td>700</td>
<td>4501.39</td>
<td>111.97</td>
<td>40.19</td>
<td>394.97</td>
<td>56.34%</td>
<td>4</td>
</tr>
<tr>
<td>800</td>
<td>5143.09</td>
<td>118.77</td>
<td>43.30</td>
<td>424.82</td>
<td>53.10%</td>
<td>4</td>
</tr>
<tr>
<td>900</td>
<td>5776.34</td>
<td>127.66</td>
<td><strong>45.25</strong></td>
<td>445.89</td>
<td>50.10%</td>
<td>4</td>
</tr>
</tbody>
</table>

It creates **45.25 million** elements per second when runs with 900 cores.
Performance Comparison
Quality

Dihedral angle range abdominal: Hybrid Method: \((4.7^\circ, 172.1^\circ)\)

Dihedral angle range knee: Hybrid Method: \((4.8^\circ, 171.7^\circ)\)
Contents

• Motivations
• Contributions
• Parallel Delaunay Mesh Generation
• Proposed Parallel Delaunay I2M Algorithms for Shared Memory Architectures
  • Locality-Aware Parallel Delaunay Image-to-Mesh (I2M) Conversion Algorithm (LAPD)
  • Parallel Delaunay I2M Conversion Algorithm for Distributed Shared Memory Architectures (PDR.PODM)
• Proposed Parallel Delaunay I2M Algorithm for Distributed Memory Clusters
  • Hybrid MPI+Threads Parallel Delaunay I2M Conversion Algorithm for Distributed Memory Clusters (Hybrid Meshing)
    • Inter-node Communication and Data Exchanges (MPI)
    • Intra-node Parallel Mesh Refinement (Pthread)
• Following Work, Time Table and Conclusion
Conclusion

Conclusion:

• LAPD:
  • Employs a data locality-aware parallel mesh refinement procedure to reduce the communication overhead caused by the remote memory accesses.

• PDR.PODM:
  • Combines the best features of two previous algorithms.

• Hybrid MPI+Threads
  • Explores two levels of parallelization: process level parallelization among nodes and thread level parallelization inside node.
  • Separate the MPI communication and local shared memory mesh refinement.

Future Work:

• Optimize current implementation and improve the scalability.
  • Utilize the current mesh generation software (like MOAB and MeshKit).
  • Explore the Index-based customized layer to avoid the explicit mesh format translation.

• Explore the hybrid MPI+Thread implementation of parallel mesh generation algorithm on heterogeneous node cluster.
Publications

Journal Papers:


Wenlu Zhang, Rongjian Li, Daming Feng, Andrey Chernikov, Nikos Chrisochoides, Christopher Osgood and Shuiwang Ji, "Evolutionary soft clustering: formulations, algorithms, and applications", Data Mining and Knowledge Discovery, August 2014.

Conference Papers and Posters:

Daming Feng, Christos Tsolakis, Andrey Chernikov and Nikos Chrisochoides, "Scalable 3D Hybrid Parallel Delaunay Image-to-Mesh Conversion Algorithm for Distributed Shared Memory Architectures", 2015 International Meshing Roundtable (IMR24), Austin, TX, October 11-14, 2015.


Kevin Garner, Daming Feng, Fotios Drakopoulos, Yixun Liu and Nikos Chrisochoides, "Image-to-Mesh Conversion Tool", 2015 International Meshing Roundtable (IMR24), Austin, TX, October 11-14, 2015
Thank you!