Two Fast Approaches for 3D Thermal Simulation of Integrated Circuits

Wenjian Yu
Dept. Computer Science and Technology
Tsinghua Univ., Beijing, China
2014.10.29

Thanks to Tao Zhang, Yuan Liang of Tsinghua, and Dr. Haifeng Qian of IBM Watson
Outline

- Background
- 3D FVM for IC Thermal Simulation
- Domain Decomposition Techniques
- A Hybrid Random Walk Method
- Conclusions
Background

- Motivation of IC thermal analysis
  - The devices in an IC continuously increase
  - Heat dissipation and thermal management become problems threatening circuit reliability and performance
- Chip-level thermal analysis (simulation): sign-off stage, also for design-time circuit optimizations
- 3D IC is a trend: reduce delay, enable heterogeneous integration
- Severe heat dissipation problem
- Importance of accurate thermal simulation during design
Background

- Chip-level thermal simulation
  - Should consider heat sink components
  - Simulating the whole IC thermal model (w/ irregular geometry) brings *computational challenges*

- Existing works
  1. Consider simplified rectangular domain
     - [Li, ICCAD’04]: Geometric multigrid iterative
  2. Consider realistic pyramid-geometry domain
     - [Zhan, TCAD’07]: Green’s function based
     - [Heriz, Thermal’07]: Convolution based, only for low-resolution
     - [Qian, ICCAD’10-TODAES’12]: Fast Poisson solver (FPS)+PCG, with increased unknowns and geometry-dependent convergence

cause > 10°C error!
3D FVM for Thermal Simulation

Problem formulation

3D steady-state heat equation

\[ k \cdot \left( \frac{\partial^2 T(x, y, z)}{\partial x^2} + \frac{\partial^2 T(x, y, z)}{\partial y^2} + \frac{\partial^2 T(x, y, z)}{\partial z^2} \right) = -p(x, y, z) \]

Boundary conditions: Neumann (adiabatic) condition, convective condition

\[ k \frac{\partial T}{\partial \vec{n}} + h(T - T_{amb}) = 0 \]

Finite difference (volume) discretization

Thermal resistor!
3D FVM for Thermal Simulation

- Practical considerations
  - Inhomogeneous material in IC region
  - Thermal resistors across various material interfaces
  - To simplify, approximate the interconnect layer with a homogeneous layer
    \[ k_{\text{eff}} = r_{\text{metal}} \cdot k_{\text{cooper}} + (1 - r_{\text{metal}}) \cdot k_{\text{oxide}} \]
  - Power source resembles current source; solve equivalent circuit equation: \[ AT = f \] Huge dimension!

- Two observations
  - Subdomain geometry regularity
  - Concern only the die region
Domain Decomposition Technique

- A general method for simulating complex domain
- Nonoverlapping DDM from FVM-circuit viewpoint
  - The solution of $\Omega_2$ provides Dirichlet boundary condition for $\Omega_1$
  - The heat flow at the bottom of $\Omega_1$ provides Neumann condition for $\Omega_2$
- Different iteration schemes
  - Top-to-bottom order
  - Bottom-to-top order
  - Middle-to-end order
  - End-to-middle order
  - Nested two-subdomain order (more reliable but costly)
  - Check convergence w/ the interfacial quantities
  - Relaxed iterative scheme:
    \[
    T_{V1}^{(i+1)} = T_{V1}^{(i)} + \omega(T_{V1}^{(i+1)} - T_{V1}^{(i)})
    \]
Domain Decomposition Technique

- Nonconformal discretization
  - Solve subdomains separately
  - Much coarser discretization used for heat spreader/sink
  - Linear interpolation converting quantities across interface; less affects the temperature in IC subdomain

- Exploit the regularity of subdomain
  - With conductivity homogenization, most subdomains are rectangular ones with simple configurations and conditions
  - FPS [Qian, ICCAD’10] with $O(n \log n)$ time-complexity and $O(n)$ space-complexity used for solving subdomains
Domain Decomposition Technique

Test cases
- Case 1: A 2D chip imitating the Power6 (175W in 1.6x2 die)
- Case 2: A 4-core 2D chip (176W in 1x1 die)
- Case 3: A 3D chip with Case 1+ 2 SRAM dies

Experiments
- Apply conformal discretization, and compare with Matlab "\"
  ICT-PCG, AMG-PCG (the fastest iterative PG solver), and FPS-PCG
- With the result of Matlab "\", check accuracy
- Apply nonconformal discretization, to show efficiency improvement
- Use the simulator in heat sink component design
Domain Decomposition Technique

- >10X memory save!
- Faster than iterative solvers for large case
- Converge in 8, 9 steps

<table>
<thead>
<tr>
<th>Matrix</th>
<th>n</th>
<th>&quot;&quot;&quot;</th>
<th>ICT-PCG</th>
<th>AMG-PCG</th>
<th>DDM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time(s)</td>
<td>Iter.</td>
<td>Time(s)</td>
<td>Iter.</td>
</tr>
<tr>
<td>M1-1</td>
<td>1.45e5</td>
<td>35.4</td>
<td>75</td>
<td>16.7</td>
<td>16</td>
</tr>
<tr>
<td>M1-2</td>
<td>1.51e5</td>
<td>38.9</td>
<td>75</td>
<td>16.8</td>
<td>13</td>
</tr>
<tr>
<td>M1-3</td>
<td>3.42e5</td>
<td>331.5</td>
<td>110</td>
<td>43.1</td>
<td>14</td>
</tr>
<tr>
<td>M1-4</td>
<td>5.47e6</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>15</td>
</tr>
<tr>
<td>M1-5</td>
<td>1.51e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>14</td>
</tr>
<tr>
<td>M1-6</td>
<td>3.42e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>M2-1</td>
<td>8.97e4</td>
<td>18.4</td>
<td>66</td>
<td>6.88</td>
<td>12</td>
</tr>
<tr>
<td>M2-2</td>
<td>1.40e5</td>
<td>43.4</td>
<td>78</td>
<td>16.7</td>
<td>12</td>
</tr>
<tr>
<td>M3-3</td>
<td>3.59e5</td>
<td>359.3</td>
<td>103</td>
<td>204.9</td>
<td>12</td>
</tr>
<tr>
<td>M2-4</td>
<td>3.51e6</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>12</td>
</tr>
<tr>
<td>M2-5</td>
<td>1.40e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>13</td>
</tr>
<tr>
<td>M2-6</td>
<td>3.31e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>M3-1</td>
<td>1.50e5</td>
<td>39.6</td>
<td>77</td>
<td>16.8</td>
<td>13</td>
</tr>
<tr>
<td>M3-2</td>
<td>1.55e5</td>
<td>40.8</td>
<td>77</td>
<td>16.9</td>
<td>13</td>
</tr>
<tr>
<td>M3-3</td>
<td>3.46e5</td>
<td>493.4</td>
<td>112</td>
<td>33.3</td>
<td>14</td>
</tr>
<tr>
<td>M3-4</td>
<td>5.54e6</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>14</td>
</tr>
<tr>
<td>M3-5</td>
<td>1.55e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>15</td>
</tr>
<tr>
<td>M3-6</td>
<td>3.46e7</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

- The temperature error in IC region is less than 0.01 °C

Graphs showing comparison of FPS-PCG, AMG-PCG, DDM.
Domain Decomposition Technique

- Nonconformal discretization
  - High-resolution simulation and design
    - $1.05 \times 10^7$ unknown (50μm discretization-step) in IC region, needs 72s simulation time
    - Study the temperature variations with the widths of heat spreader/sink changed
    - 24-configuration simulation costs 7 mins.

• Apply to larger case
• $< 0.05$ °C Error on hot spot
• $>10X$ memory save

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Conformal grid</th>
<th>Non-conformal grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td>Time(s)</td>
</tr>
<tr>
<td>M1-4</td>
<td>5.47e6</td>
<td>32.3</td>
</tr>
<tr>
<td>M1-5</td>
<td>1.51e7</td>
<td>90.7</td>
</tr>
<tr>
<td>M1-6</td>
<td>3.42e7</td>
<td>203.2</td>
</tr>
<tr>
<td>M1-10</td>
<td>5.41e7</td>
<td>310.1</td>
</tr>
</tbody>
</table>
A Hybrid Random Walk Method

- Thermal simulation for hot-spots at device layer
  - Entire temperature profile is often not required
- Random walk method for thermal analysis
  - Equivalent to the P/G analysis problem, w/ thermal resistors
  - P/G Random walk methods [Qian, TCAD’05][Miyakawa, GLSVLSI’11]
  - [Wong, DATE’06]: used for thermal via planning in 3D IC
- Drawbacks of existing works
  - Not consider characteristics of IC thermal problem (boundary condition, geometry features)
  - The computational speed is very slow
Main ideas

- Heat sink components (e.g., Pyramid-shape)
- Slow speed of RW (called GRW) is due to the long length of a walk path
- Another RW (FRW) is able to reduce the length of a walk path
- FRW encounters difficulty if there are the source item and Neumann, convective boundary condition

Can We Combine Them?
A Hybrid Random Walk Method

- **GRW+FRW**
  - Perform FRW in simple regions
  - **Cuboid** transition domains
    - I (homogeneous)
    - II (half & half homogeneous)
  - FRW transition
    \[
    T_c = \int_S G_s(r)T(r)dr
    \]
  - Pre-characterize the transition domains with a hop-target table
  - **Neumann boundary**: path reflection / special transition domain
  - **Convective boundary**: Large $R_{amb}$ barriers GRW hop; convective-specific transition domain
A Hybrid Random Walk Method

- **Test cases**
  - Case 1: A 4-core 2D chip (176W in 1x1 die)
  - Case 2: A 2D chip imitating the Power6 (175W in 1.6x2 die)

- **Experimental results**
  - **Hybrid0**: GRW+FRW; **Hybrid1**: Neumann boundary treatment;
  - **Hybrid2**: convective boundary treatment (1% 1-σ error)

  Average runtime for calculating the temperature of a node (s)

<table>
<thead>
<tr>
<th>Test case</th>
<th>#node</th>
<th>GRW</th>
<th>Hybrid random walk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>time</td>
<td>#walk</td>
</tr>
<tr>
<td>1-1</td>
<td>5.24e5</td>
<td>49.8</td>
<td>5471</td>
</tr>
<tr>
<td>1-2</td>
<td>4.19e6</td>
<td>199</td>
<td>5522</td>
</tr>
<tr>
<td>1-3</td>
<td>6.55e7</td>
<td>949</td>
<td>6409</td>
</tr>
<tr>
<td>2-1</td>
<td>5.33e5</td>
<td>35.5</td>
<td>3576</td>
</tr>
<tr>
<td>2-2</td>
<td>4.26e6</td>
<td>143</td>
<td>3709</td>
</tr>
<tr>
<td>2-3</td>
<td>6.66e7</td>
<td>762</td>
<td>3281</td>
</tr>
</tbody>
</table>
A Hybrid Random Walk Method

- Experimental results
  - Memory overhead: ~81MB for transition-domain characterization
  - The pre-characterization runs only once for same chip structure or chips manufactured by same materials
  - Accuracy validated by Matlab “\”
  - How the aspect ratio of the cuboid transition domain in FRW affects the efficiency of the proposed hybrid method?

Average Hybrid1’s runtime for calculating a node’s temperature (s)

<table>
<thead>
<tr>
<th>Aspect ratio</th>
<th>1</th>
<th>5</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time/node</td>
<td>170</td>
<td>43.3</td>
<td>41.9</td>
<td>41.3</td>
<td>40.9</td>
<td>41.8</td>
<td>42.5</td>
</tr>
</tbody>
</table>

- Choosing cuboid transition domain (a.r.=10) brings 4.1X speedup
Conclusions

- Domain decomposition method
  - Make the fast thermal solvers workable while appreciating the effect of heat sink components
  - Can beat iterative equation solver for large case
  - Nonconformal discretization grid reduces the computation with negligible loss of accuracy
- Hybrid random walk method
  - Combining GRW and FRW brings one or two orders of magnitude speedup, with some memory overhead
  - Suitable for the scenarios where only the temperature of some local hot-spots is needed
Thanks!