Energy Auto-tuning using Polyhedral Approach

Wei Wang[1], John Cavazos[1], Allan Porterfield[2]

[1] Dept. of Computer and Info. Sciences, University of Delaware

[2] RENCI, University of North Carolina-Chapel Hill

Introduction

- Application Energy Consumption
  - Optimizing for lower energy has become critical when we approach Exascale.
- Tuning for faster execution vs. tuning for lower Energy?
  - Knowledge of the relationship between the two will guide auto-tuning process.
- Energy Impact of Polyhedral Optimization
  - Never studied.
  - Polyhedral optimizations barely studied on non-trivial/realistic applications.

Energy Measurement using RCRTool

- Architecture Tested
  - Sandy Bridge, Ivy Bridge
  - Shared memory stores MSR counters.
  - Update frequency: > 1000/s.
  - Supported Language: OpenMP, MPI.
- MIC
  - Shared memory stores energy obtained from PAPI and Intel MicAccessSDK.
  - Update frequency: about 20/s.
  - Host version and MIC-native version.
  - Supported Language: OpenMP (offload and native), OpenCL (host).

RCRTool Exposed APIs

- energyDaemonInit()
- energyDaemonEnter(): Start/Resume measurement when entering a region.
- energyDaemonExit(file, line_no): Stop/Pause measurement upon exiting the region
- energyDaemonTerm()
- energyDaemonTEStart(): Start measuring Time and Energy
- energyDaemonTEStop(): Stop measuring Time and Energy

RCRTool Exposed APIs - Example

```
1  int main() {
2
3    initialize();
4
5
6    while  {
7
8      #pragma omp parallel for
9      compute_region_1();
10
11
12     serial_code();
13
14
15     #pragma omp parallel
16     #pragma for
17     compute_region_2();
18
19
20   }
21
22
23   finalize();
24 }
```

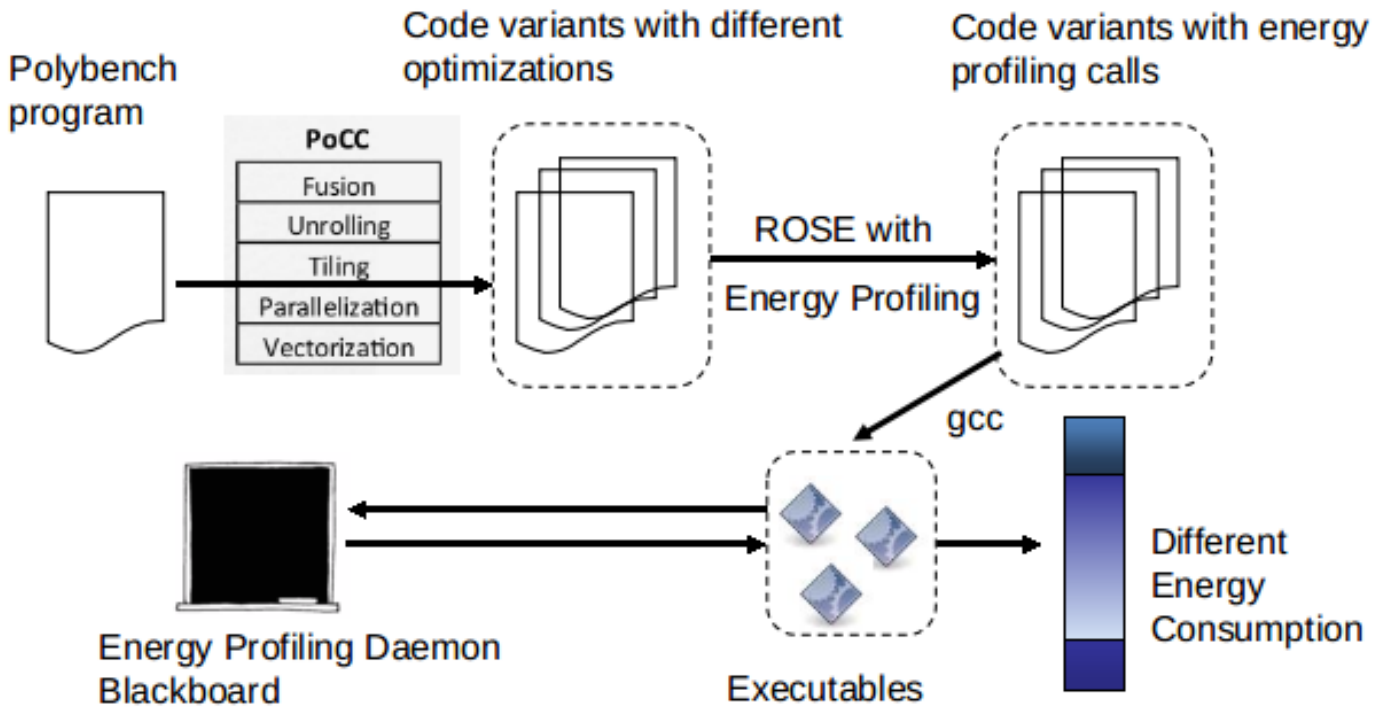
```
1  int main() {
2    energyDaemonInit();
3    initialize();
4
5    energyDaemonTEStart();
6    while  {
7      energyDaemonEnter();
8      #pragma omp parallel for
9      compute_region_1();
10     energyDaemonExit("thefile", 10);
11
12     serial_code();
13
14     energyDaemonEnter();
15     #pragma omp parallel
16     #pragma for
17     compute_region_2();
18     energyDaemonExit("the file", 18);
19
20   }
21   energyDaemonTEStop();
22
23   finalize();
24   energyDameonTerm();
25 }
```

Polyhedral Compilers

Generate code variants of a program containing Static Control Parts (SCoP) using PoCC (Polyhedral Compiler Collection).

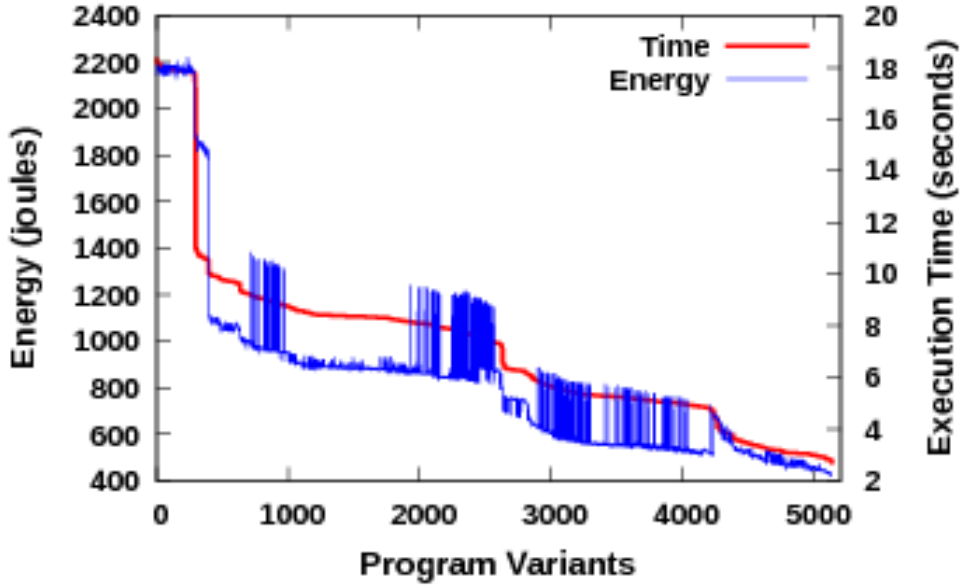
- Loop Transformations
- Auto Parallelization (PLUTO)
- Applications
  - Existing: Polybench, SWIM
  - New: 2D Cardiac Wave Propagation Simulation, LULESH (C/C++)

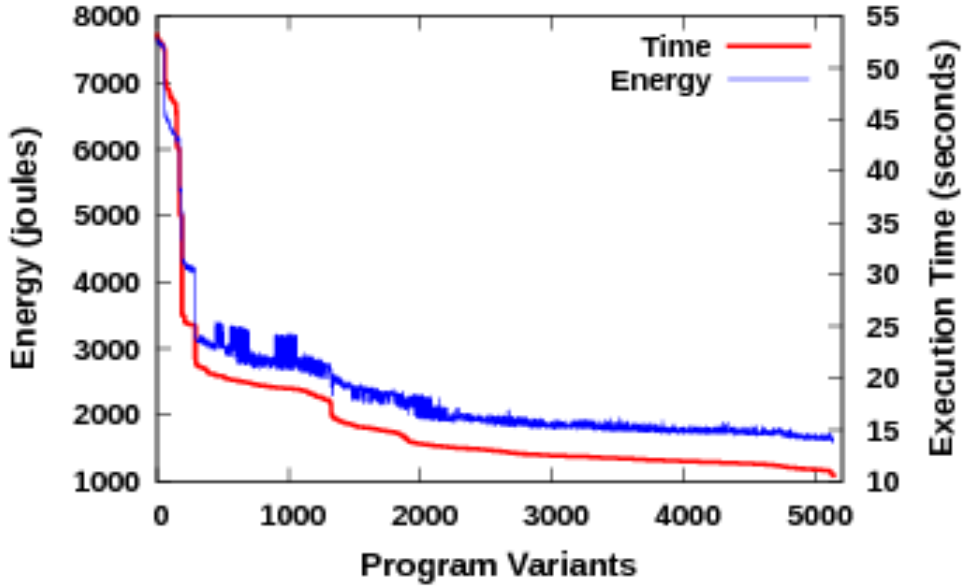
Energy Profiling of Different Program Optimizations



Workflow of energy-aware polyhedral framework

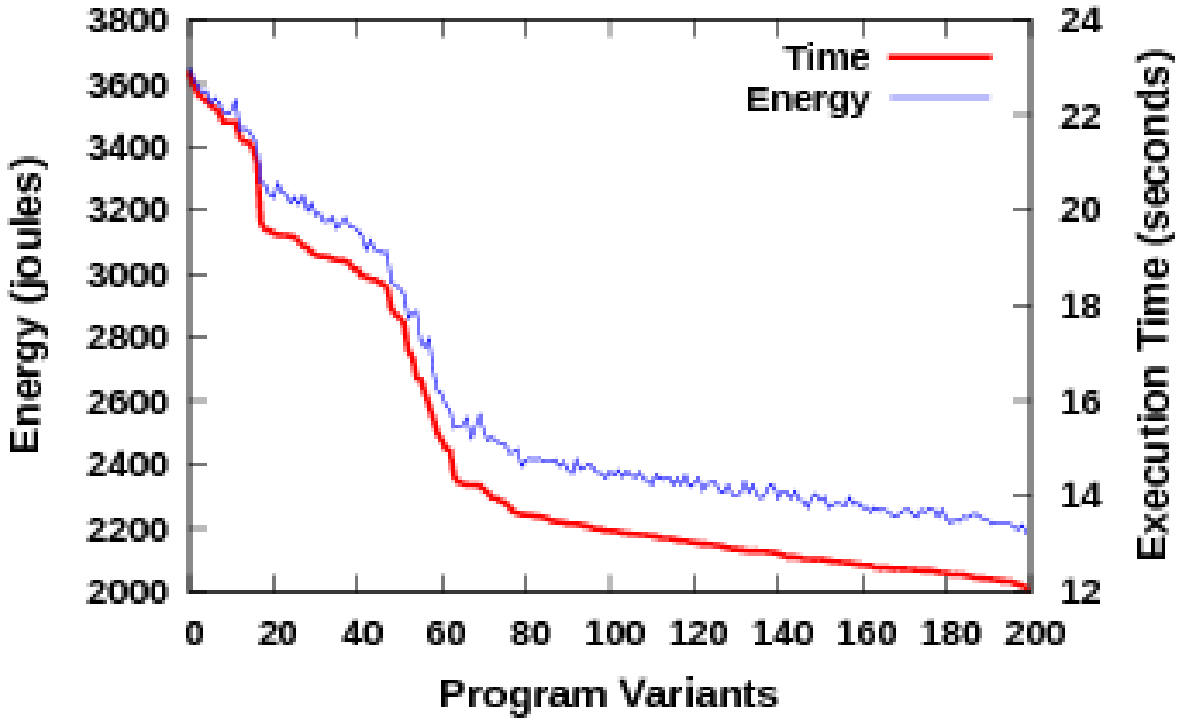
Energy Consumption and Execution Time Correlation (Polybench)





Loop fusion (maxfuse) reduce execution time but increases energy consumption (spikes and the tail in Covariance benchmark). Bad tiling configuration increases energy consumption (spikes in 2mm benchmark). Best optimizations for time are best for energy savings for these two polybench application.

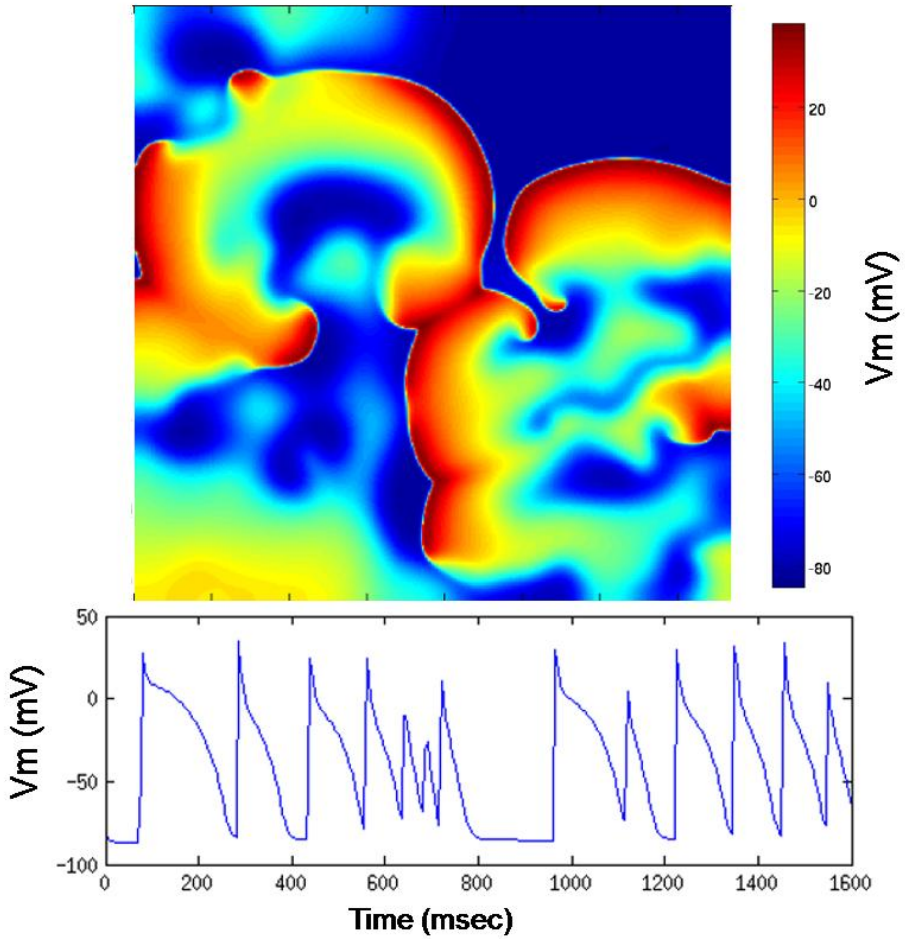
Energy Consumption and Execution Time Correlation (LULESH)



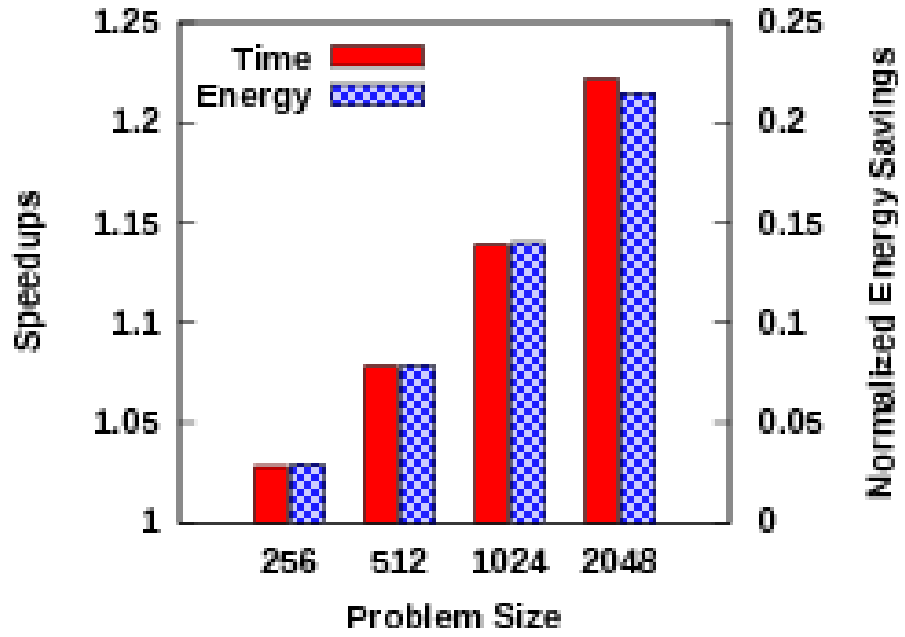
As a larger application, LULESH also displays the similar correlation between energy and time. The best optimized program for time is also for energy. (Note: the graph is from optimizing one loop nest).

Effectiveness of Polyhedral Optimizations on a Realistic Application

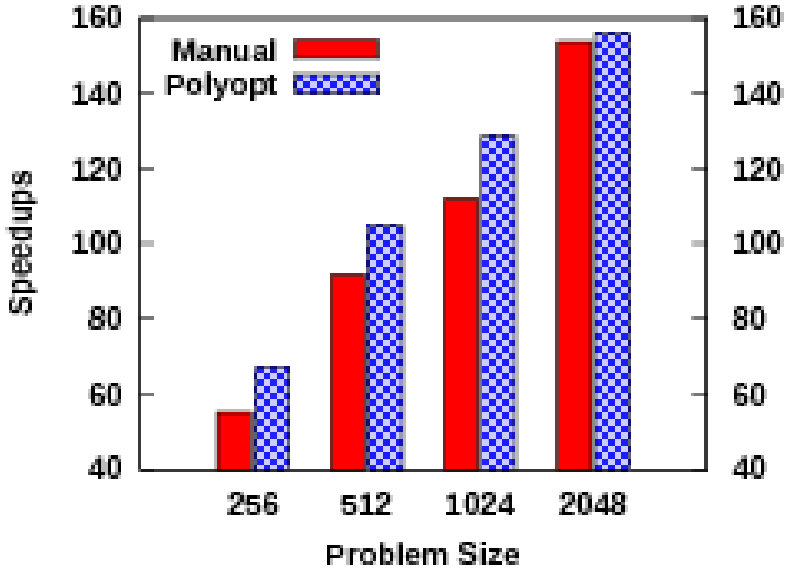
- 2D Cardiac Wave Propagation Simulation

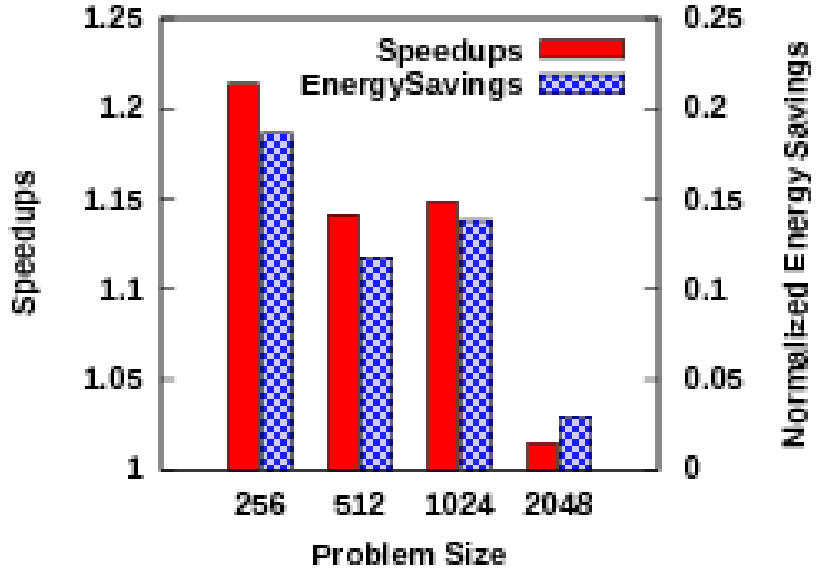


- Speedups obtained on a Sandy Bridge system



Preliminary Results on MIC for Cardiac Simulation





Left: The best optimized PolyOpt program variant vs manual OpenMP (over sequential baseline).

Right: Speedups and energy savings comparing the manual OpenMP with the best PolyOpt program variant.

Conclusion: Polyhedral Approach is effective in optimizing the 2D Cardiac Wave Propagation Simulation.

Challenges/Limitations using Polyhedral Compilers

- Exposing SCoPs of the application
  - LULESH contains six large regions that are potential SCoPs.
- Temporary (array/scalar) variables
  - Large number of dependences between statements in a SCoP.
  - In LULESH, a human-readable SCoP can easily contain thousands of dependences.
- Temporary variables elimination
  - Resulting code is not human-readable and may reduce optimization effectiveness.

Polyhedral Transformable LULESH Code

```
  dxx(WW) = (1.0)/( (8.) * ( (.125 * ( ((m_x[i+1][j+1][k+1] - (0.5)*dt * m_xd[i+1][j+1][k+1])-(m_x[i][j][k] - (0.5)*dt * m_xd[i][j][k])) - ((m_x
[i+1][j][k+1] - (0.5)*dt * m_xd[i+1][j][k+1])-(m_x[i][j+1][k] - (0.5)*dt * m_xd[i][j+1][k])) + ((m_x[i+1][j+1][k] - (0.5)*dt * m_xd[i+1][j+1][k]
)-(m_x[i][j][k+1] - (0.5)*dt * m_xd[i][j][k+1])) - ((m_x[i+1][j][k] - (0.5)*dt * m_xd[i+1][j][k])-(m_x[i][j+1][k+1] - (0.5)*dt * m_xd[i][j+1][k+
1])) ) ) * (- ((.125 * ( ((m_y[i+1][j+1][k+1] - (0.5)*dt * m_yd[i+1][j+1][k+1])-(m_y[i][j][k] - (0.5)*dt * m_yd[i][j][k])) + ((m_y[i+1][j][k+1]
 - (0.5)*dt * m_yd[i+1][j][k+1])-(m_y[i][j+1][k] - (0.5)*dt * m_yd[i][j+1][k])) - ((m_y[i+1][j+1][k] - (0.5)*dt * m_yd[i+1][j+1][k])-(m_y[i][j][k
+1] - (0.5)*dt * m_yd[i][j][k+1])) - ((m_y[i+1][j][k] - (0.5)*dt * m_yd[i+1][j][k])-(m_y[i][j+1][k+1] - (0.5)*dt * m_yd[i][j+1][k+1])) ) ) * (.1
25 * ( ((m_z[i+1][j+1][k+1] - (0.5)*dt * m_zd[i+1][j+1][k+1])-(m_z[i][j][k] - (0.5)*dt * m_zd[i][j][k])) + ((m_z[i+1][j][k+1] - (0.5)*dt * m_zd[
i+1][j][k+1])-(m_z[i][j+1][k] - (0.5)*dt * m_zd[i][j+1][k])) + ((m_z[i+1][j+1][k] - (0.5)*dt * m_zd[i+1][j+1][k])-(m_z[i][j][k+1] - (0.5)*dt * m
_zd[i][j][k+1])) + ((m_z[i+1][j][k] - (0.5)*dt * m_zd[i+1][j][k])-(m_z[i][j+1][k+1] - (0.5)*dt * m_zd[i][j+1][k+1])) ) )) + ((.125 * ( ((m_z[i+1
][j+1][k+1] - (0.5)*dt * m_zd[i+1][j+1][k+1])-(m_z[i][j][k] - (0.5)*dt * m_zd[i][j][k])) + ((m_z[i+1][j][k+1] - (0.5)*dt * m_zd[i+1][j][k+1])-(m
_z[i][j+1][k] - (0.5)*dt * m_zd[i][j+1][k])) - ((m_z[i+1][j+1][k] - (0.5)*dt * m_zd[i+1][j+1][k])-(m_z[i][j][k+1] - (0.5)*dt * m_zd[i][j][k+1]))
 - ((m_z[i+1][j][k] - (0.5)*dt * m_zd[i+1][j][k])-(m_z[i][j+1][k+1] - (0.5)*dt * m_zd[i][j+1][k+1])) ) ) * (.125 * ( ((m_y[i+1][j+1][k+1] - (0.5
)*dt * m_yd[i+1][j+1][k+1])-(m_y[i][j][k] - (0.5)*dt * m_yd[i][j][k])) + ((m_y[i+1][j][k+1] - (0.5)*dt * m_yd[i+1][j][k+1])-(m_y[i][j+1][k] - (0
.5)*dt * m_yd[i][j+1][k])) + ((m_y[i+1][j+1][k] - (0.5)*dt * m_yd[i+1][j+1][k])-(m_y[i][j][k+1] - (0.5)*dt * m_yd[i][j][k+1])) + ((m_y[i+1][j][k
] - (0.5)*dt * m_yd[i+1][j][k])-(m_y[i][j+1][k+1] - (0.5)*dt * m_yd[i][j+1][k+1])) ) )) ) + (.125 * ( ((m_y[i+1][j+1][k+1] - (0.5)*dt * m_yd[i+1
][j+1][k+1])-(m_y[i][j][k] - (0.5)*dt * m_yd[i][j][k])) - ((m_y[i+1][j][k+1] - (0.5)*dt * m_yd[i+1][j][k+1])-(m_y[i][j+1][k] - (0.5)*dt * m_yd[i
][j+1][k])) + ((m_y[i+1][j+1][k] - (0.5)*dt * m_yd[i+1][j+1][k])-(m_y[i][j][k+1] - (0.5)*dt * m_yd[i][j][k+1])) - ((m_y[i+1][j][k] - (0.5)*dt *
 m_yd[i+1][j][k])-(m_y[i][j+1][k+1] - (0.5)*dt * m_yd[i][j+1][k+1])) ) ) * (  ((.125 * ( ((m_x[i+1][j+1][k+1] - (0.5)*dt * m_xd[i+1][j+1][k+1])-(
m_x[i][j][k] - (0.5)*dt * m_xd[i][j][k])) + ((m_x[i+1][j][k+1] - (0.5)*dt * m_xd[i+1][j][k+1])-(m_x[i][j+1][k] - (0.5)*dt * m_xd[i][j+1][k])) -
 ((m_x[i+1][j+1][k] - (0.5)*dt * m_xd[i+1][j+1][k])-(m_x[i][j][k+1] - (0.5)*dt * m_xd[i][j][k+1])) - ((m_x[i+1][j][k] - (0.5)*dt * m_xd[i+1][j][k
])-(m_x[i][j+1][k+1] - (0.5)*dt * m_xd[i][j+1][k+1])) ) ) * (.125 * ( ((m_z[i+1][j+1][k+1] - (0.5)*dt * m_zd[i+1][j+1][k+1])-(m_z[i][j][k] - (0.
5)*dt * m_zd[i][j][k])) + ((m_z[i+1][j][k+1] - (0.5)*dt * m_zd[i+1][j][k+1])-(m_z[i][j+1][k] - (0.5)*dt * m_zd[i][j+1][k])) + ((m_z[i+1][j+1][k]
 - (0.5)*dt * m_zd[i+1][j+1][k])-(m_z[i][j][k+1] - (0.5)*dt * m_zd[i][j][k+1])) + ((m_z[i+1][j][k] - (0.5)*dt * m_zd[i+1][j][k])-(m_z[i][j+1][k+
1] - (0.5)*dt * m_zd[i][j+1][k+1])) ) )) - ((.125 * ( ((m_z[i+1][j+1][k+1] - (0.5)*dt * m_zd[i+1][j+1][k+1])-(m_z[i][j][k] - (0.5)*dt * m_zd[i][
j][k])) + ((m_z[i+1][j][k+1] - (0.5)*dt * m_zd[i+1][j][k+1])-(m_z[i][j+1][k] - (0.5)*dt * m_zd[i][j+1][k])) - ((m_z[i+1][j+1][k] - (0.5)*dt * m_
zd[i+1][j+1][k])-(m_z[i][j][k+1] - (0.5)*dt * m_zd[i][j][k+1])) - ((m_z[i+1][j][k] - (0.5)*dt * m_zd[i+1][j][k])-(m_z[i][j+1][k+1] - (0.5)*dt *
 m_zd[i][j+1][k+1])) ) ) * (.125 * ( ((m_x[i+1][j+1][k+1] - (0.5)*dt * m_xd[i+1][j+1][k+1])-(m_x[i][j][k] - (0.5)*dt * m_xd[i][j][k])) + ((m_x[i+
1][j][k+1] - (0.5)*dt * m_xd[i+1][j][k+1])-(m_x[i][j+1][k] - (0.5)*dt * m_xd[i][j+1][k])) + ((m_x[i+1][j+1][k] - (0.5)*dt * m_xd[i+1][j+1][k])-(
m_x[i][j][k+1] - (0.5)*dt * m_xd[i][j][k+1])) + ((m_x[i+1][j][k] - (0.5)*dt * m_xd[i+1][j][k])-(m_x[i][j+1][k+1] - (0.5)*dt * m_xd[i][j+1][k+1])
) ) )) ) + (.125 * ( ((m_z[i+1][j+1][k+1] - (0.5)*dt * m_zd[i+1][j+1][k+1])-(m_z[i][j][k] - (0.5)*dt * m_zd[i][j][k])) - ((m_z[i+1][j][k+1] - (0
.5)*dt * m_zd[i+1][j][k+1])-(m_z[i][j+1][k] - (0.5)*dt * m_zd[i][j+1][k])) + ((m_z[i+1][j+1][k] - (0.5)*dt * m_zd[i+1][j+1][k])-(m_z[i][j][k+1]
 - (0.5)*dt * m_zd[i][j][k+1])) - ((m_z[i+1][j][k] - (0.5)*dt * m_zd[i+1][j][k])-(m_z[i][j+1][k+1] - (0.5)*dt * m_zd[i][j+1][k+1])) ) ) * (- ((.1
25 * ( ((m_x[i+1][j+1][k+1] - (0.5)*dt * m_xd[i+1][j+1][k+1])-(m_x[i][j][k] - (0.5)*dt * m_xd[i][j][k])) + ((m_x[i+1][j][k+1] - (0.5)*dt * m_xd[
i+1][j][k+1])-(m_x[i][j+1][k] - (0.5)*dt * m_xd[i][j+1][k])) - ((m_x[i+1][j+1][k] - (0.5)*dt * m_xd[i+1][j+1][k])-(m_x[i][j][k+1] - (0.5)*dt * m
_xd[i][j][k+1])) - ((m_x[i+1][j][k] - (0.5)*dt * m_xd[i+1][j][k])-(m_x[i][j+1][k+1] - (0.5)*dt * m_xd[i][j+1][k+1])) ) ) * (.125 * ( ((m_y[i+1][
j+1][k+1] - (0.5)*dt * m_yd[i+1][j+1][k+1])-(m_y[i][j][k] - (0.5)*dt * m_yd[i][j][k])) + ((m_y[i+1][j+1][k+1] - (0.5)*dt * m_yd[i+1][j+1][k+1])-(m_y
[i][j+1][k] - (0.5)*dt * m_yd[i][j+1][k])) + ((m_y[i+1][j+1][k] - (0.5)*dt * m_yd[i+1][j+1][k])-(m_y[i][j][k+1] - (0.5)*dt * m_yd[i][j][k+1])) +
 ((m_y[i+1][j][k] - (0.5)*dt * m_yd[i+1][j][k])-(m_y[i][j+1][k+1] - (0.5)*dt * m_yd[i][j+1][k+1])) ) )) + ((.125 * ( ((m_y[i+1][j+1][k+1] - (0.5
```

Eliminating dependences resulting in bad-looking code

Conclusion

- Collect power info from OpenMP programs using RCRTool on SandyBridge (IvyBridge) and on MIC.
- Power/time correlation observed for small polybench programs and larger LULESH program.
- Speedups and energy savings obtained from a medium-size realistic (2D cardiac wave propagation) application on SandyBridge and on MIC.
- Problems/limitations exist when applied to larger-size LULESH program using polyhedral techniques.