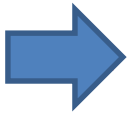
Multigrain Parallelization for MATLAB/Simulink Using the OSCAR Compiler

Ryo Koyama・Yuta Tsumura・Dan Umeda
・Keiji Kimura・Hironori Kasahara

Waseda University

Our motivation

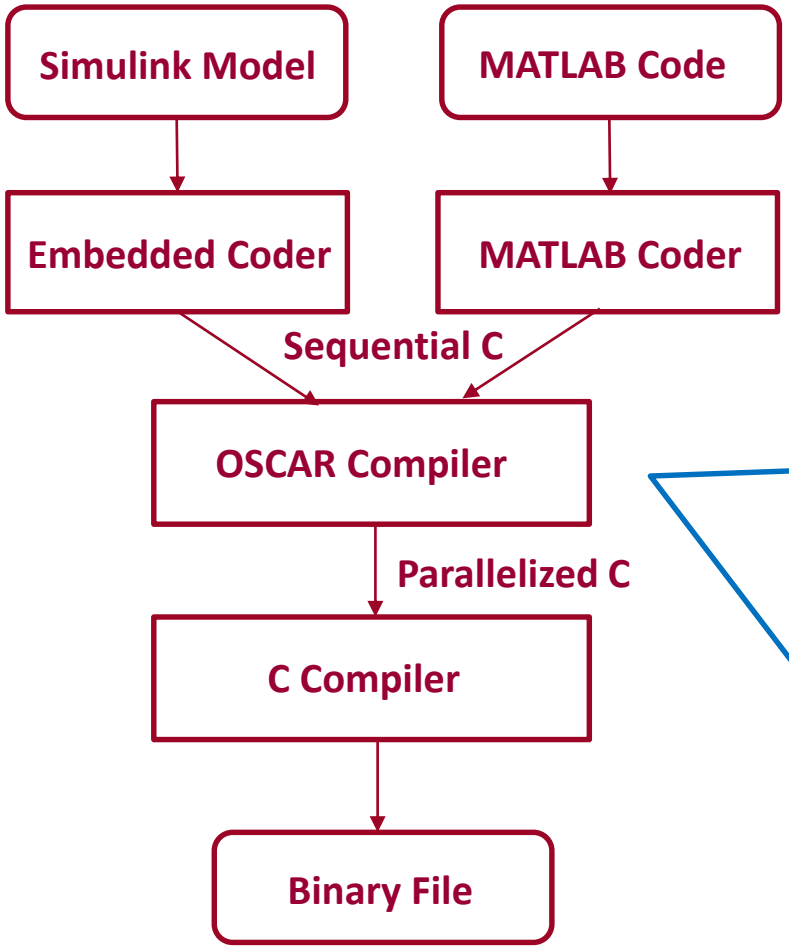
- Model-based developments become very popular.
  - MATLAB/Simulink for embedded systems
- MATLAB/Simulink models need more speed.
  - Fast program execution requires parallelization.
  - Parallelizing huge applications manually is tough work.



OSCAR Automatically Parallelizing Compiler

- Powerful program analysis and parallelization
- Taking C programs generated by MATLAB/Simulink
- Generating parallelized programs
  - To deploy them to embedded multicore systems, as well as servers

Work Flow



Multigrain Parallelization

coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism

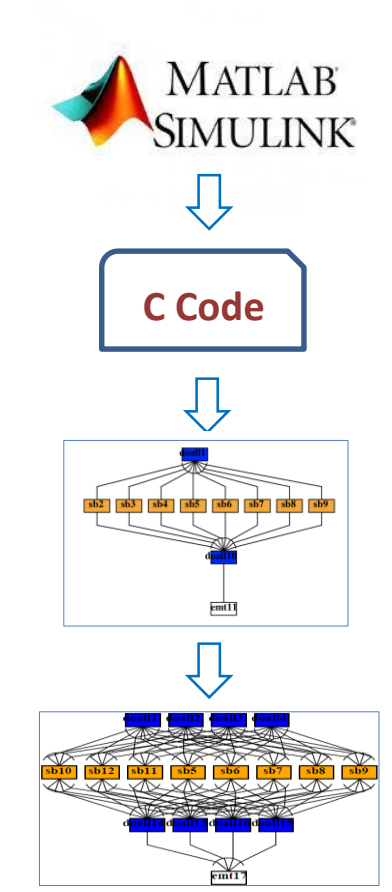
Data Localization

Automatic data management for distributed shared memory, cache and local memory

Power Reduction

Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.

Parallelism Analysis



Macro-tasks graph (MTG)

MATLAB/Simulink generates C code

1. MATLAB Coder
2. Embedded Coder

OSCAR generates Macro-tasks from C code and exploits parallelism

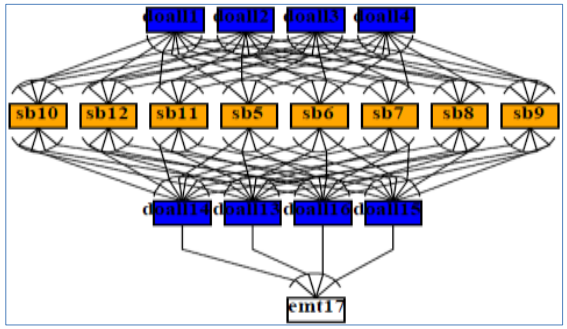
1. Basic Block
2. Subroutine Block
3. Repetition Block (Loop Block)

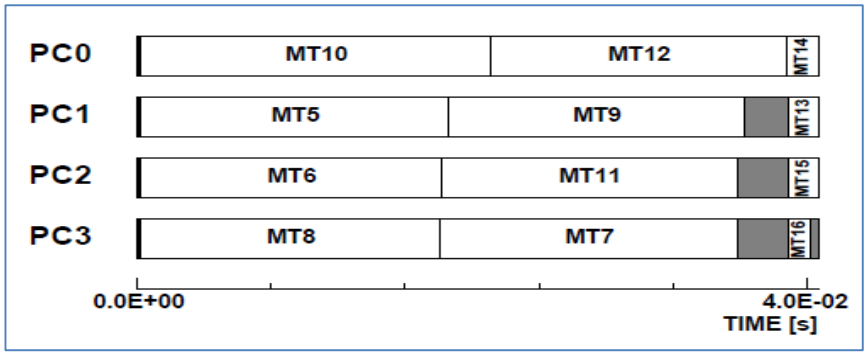
Exploiting parallelism among them(Macro Task Graph: MTG)

OSCAR restructures program for more parallelism

1. Macro task fusion
2. Branch decomposition

Scheduling onto Multicores





Static Scheduling

Assign macro-tasks statically to reduce runtime overhead

Macro-task fusion

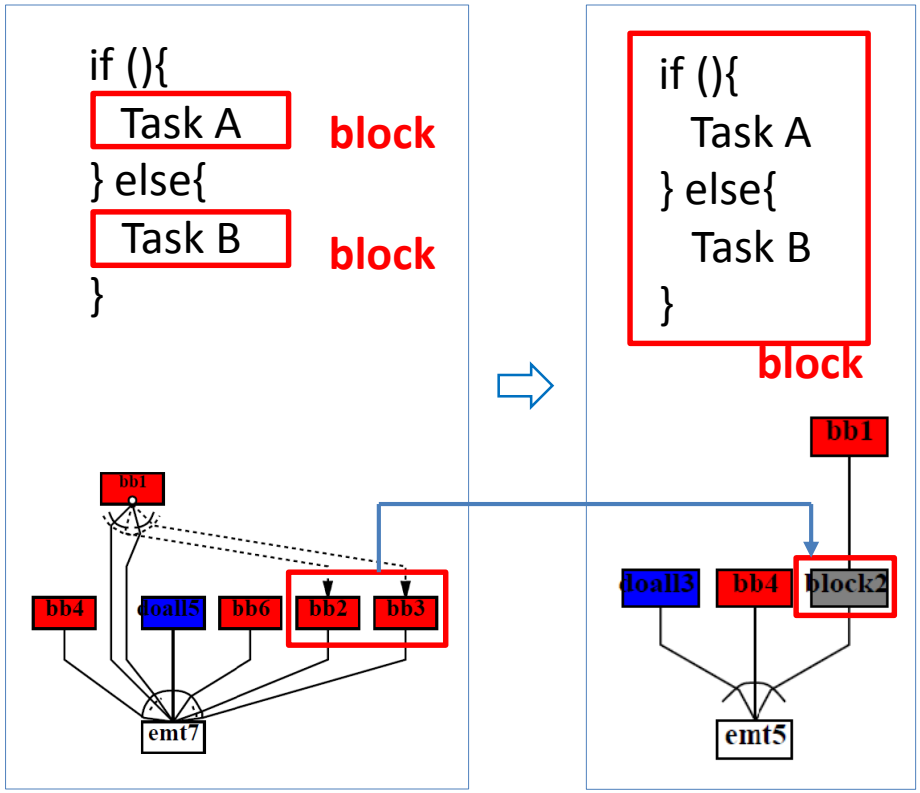
Merge MTs to hide conditional branches to apply static scheduling

Branch decomposition

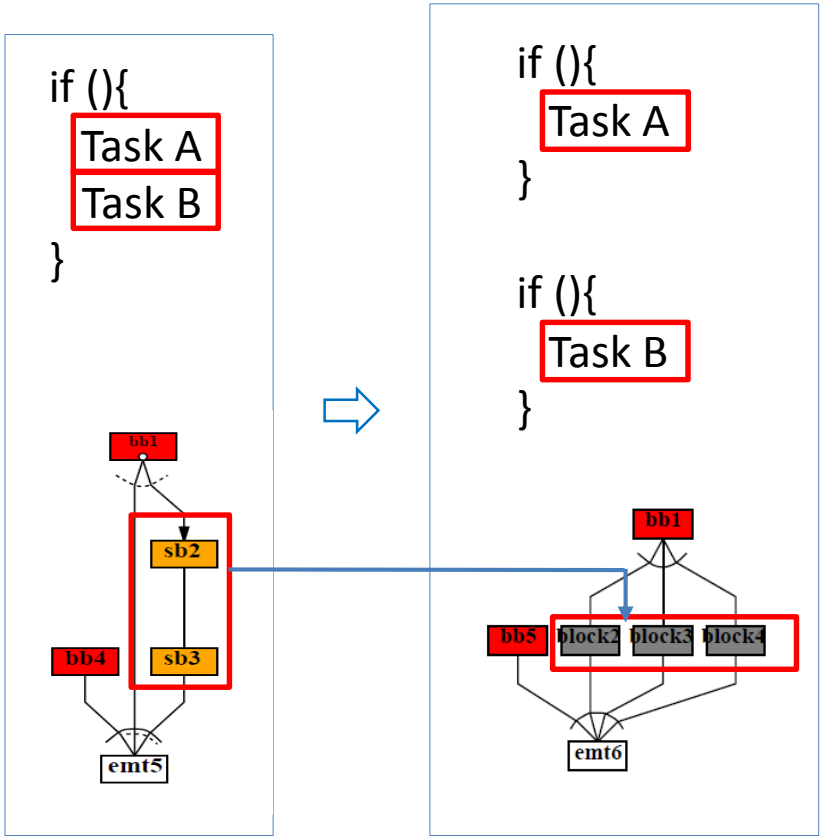
Duplicate and decompose the if statements and extract the parallelism of the body of the if statement.

Macro task fusion and Branch decomposition

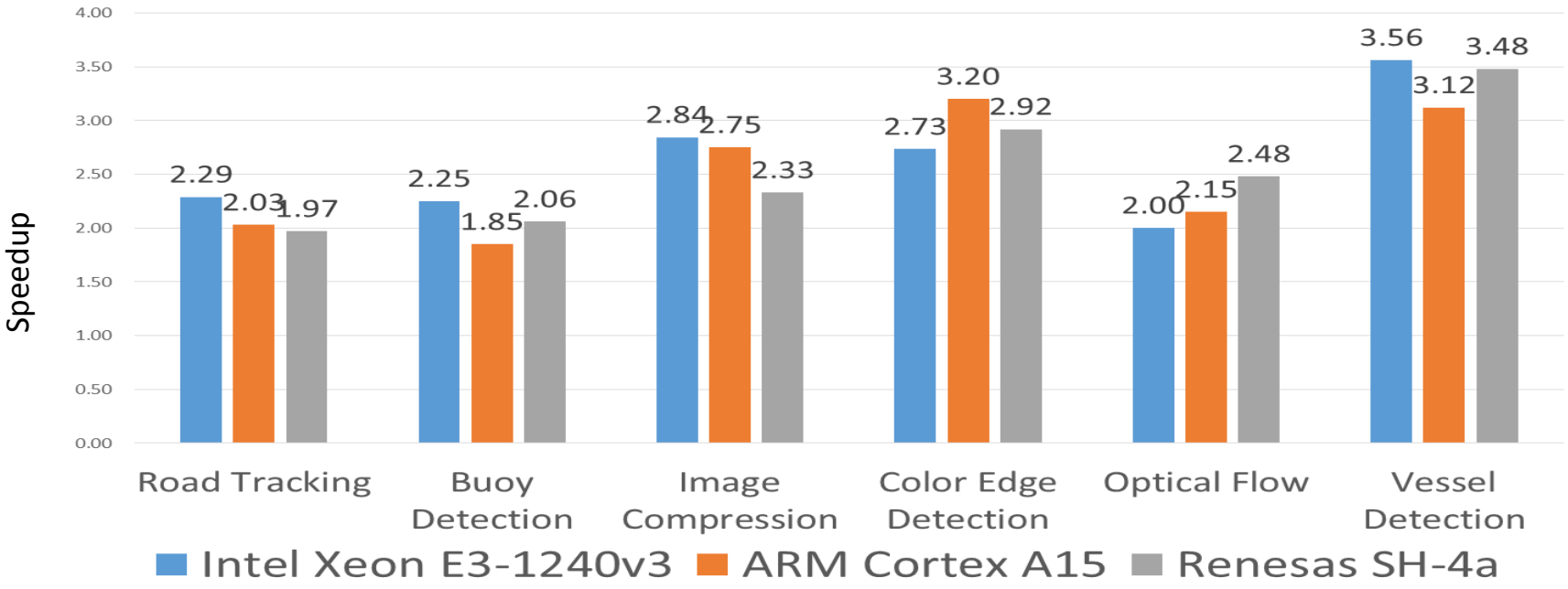
Macro task fusion



Branch decomposition

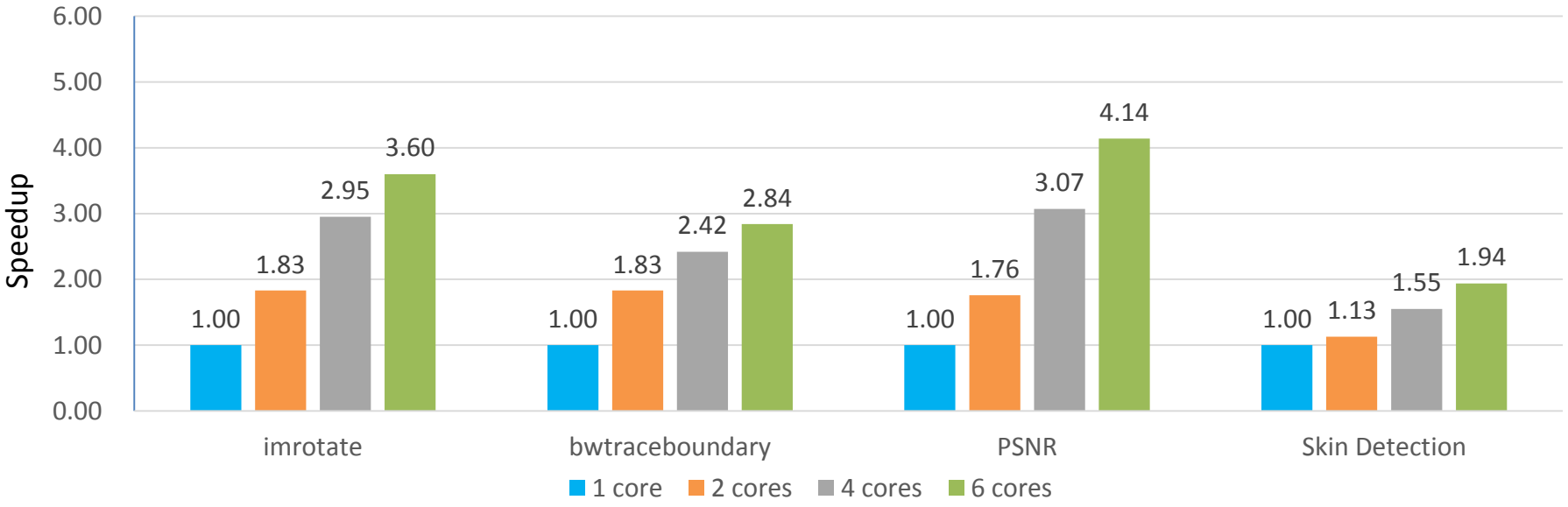


Speedups of Simulink Image Processing on Various 4core Multicores (Intel Xeon, ARM Cortex A15 and Renesas SH4A)



Road Tracking, Image Compression : http://www.mathworks.co.jp/jp/help/vision/examples
Buoy Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/44706-buoy-detection-using-simulink
Color Edge Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image--actual-color--not-converting-to-grayscale-/
Vessel Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction/

Speedups of MATLAB Image Processing on Intel Xeon E5-2650 v4



imrotate: http://www.mathworks.co.jp/help/images/ref/imrotate
bwtraceboundary : http://www.mathworks.co.jp/help/images/ref/bwtraceboundary
PSNR: http://www.mathworks.co.jp/matlabcentral/fileexchange/64151-structure-similarity-ssim-and-psnr
Skin Detection: http://www.mathworks.co.jp/matlabcentral/fileexchange/28565-skin-detection

Conclusion

- OSCAR Compiler parallelizes MATLAB/Simulink models automatically
  - Use multi-grain parallelism
    - Loop parallelism
    - Coarse grain parallelism
  - Use static scheduling to reduce runtime overhead
    - Macro task fusion
    - Branch decomposition

- 1.85-3.56x speedup for Simulink
  - Road Tracking, Buoy Detection, etc.

- 1.94-4.14x speedup for MATLAB
  - Imrotate, bwtraceboundary, etc.