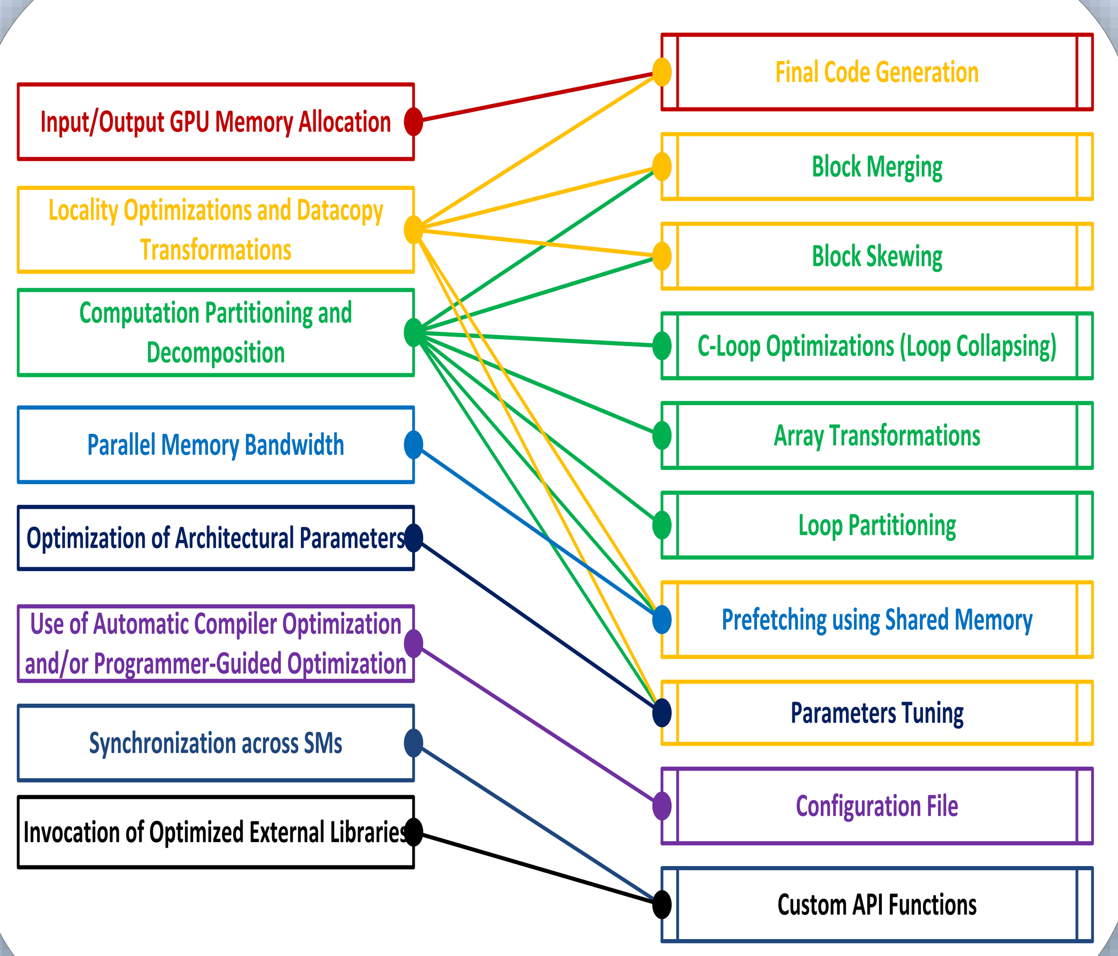
Optimization Specifications for CUDA Code Restructuring Tool

Ayaz. H. Khan. Computer Science Department, College of Computer, Qassim University, **Email:** ay.khan@qu.edu.sa.

Abstract

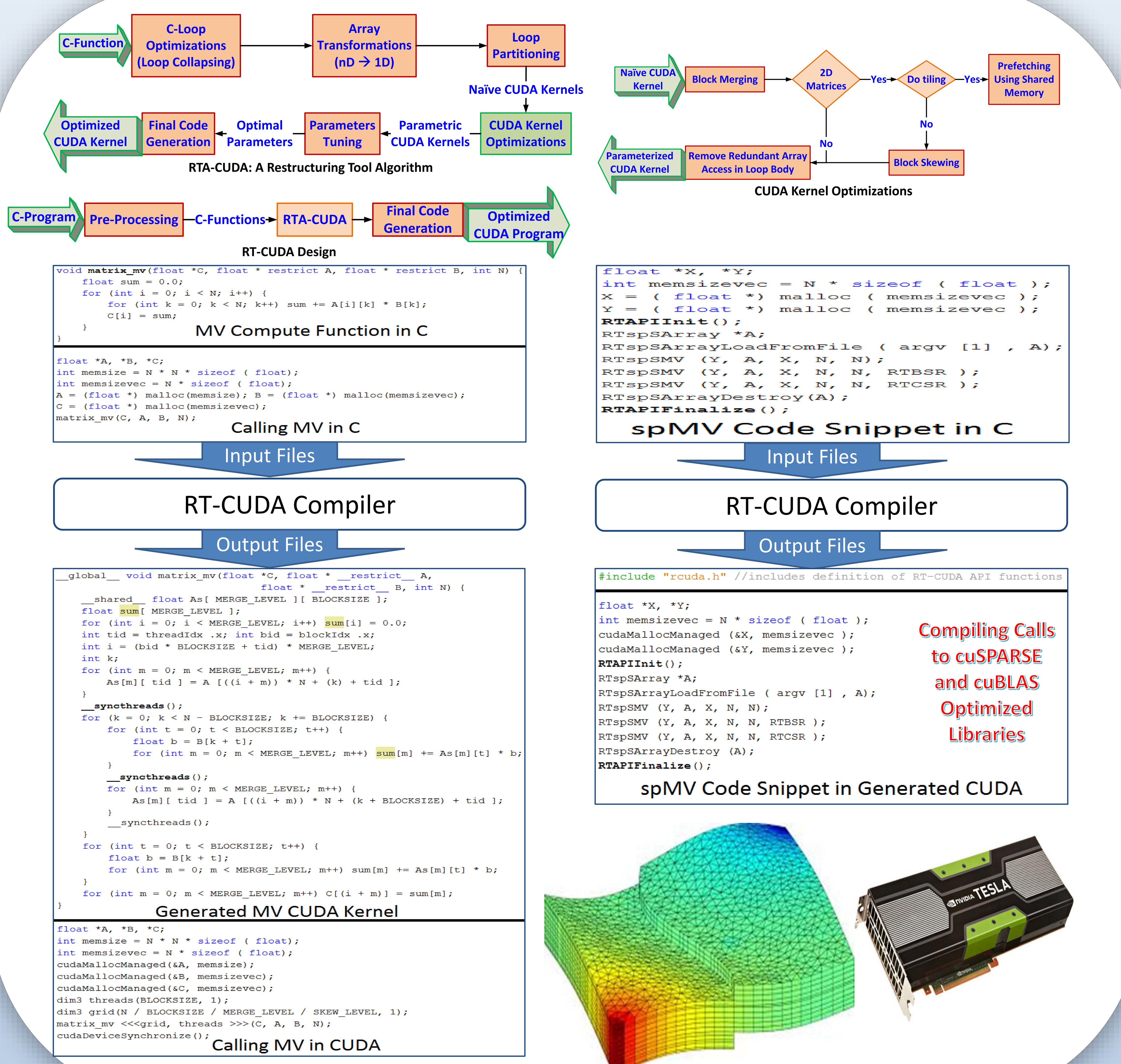
In this work we have developed a restructuring software tool (RT-CUDA) following the proposed optimization specifications to bridge the gap between high-level languages and the machine dependent CUDA environment. RT-CUDA takes a C program and convert it into an optimized CUDA kernel with user directives in a configuration file for guiding the compiler. RT-CUDA also allows transparent invocation of the most optimized external math libraries like cuSparse and cuBLAS enabling efficient design of linear algebra solvers. We expect RT-CUDA to be needed by many KSA industries dealing with science and engineering simulation on massively parallel computers like NVIDIA GPUs.

Specifications



Mapping of Optimization Specifications with Code Transformations

RT-CUDA Compiler and Its Application to Scientific Simulation



Compiling Native C-code into optimized CUDA

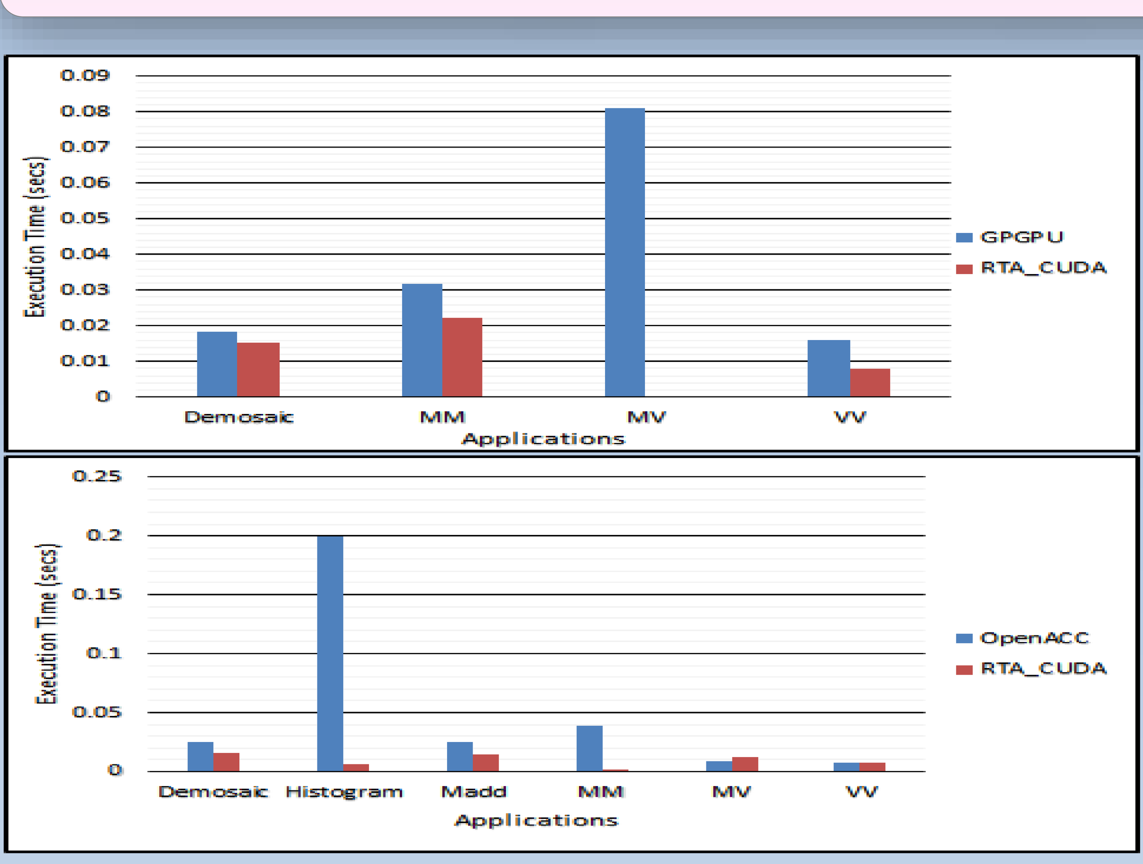
Ease and Efficient Library Coding of Large Scale Solvers

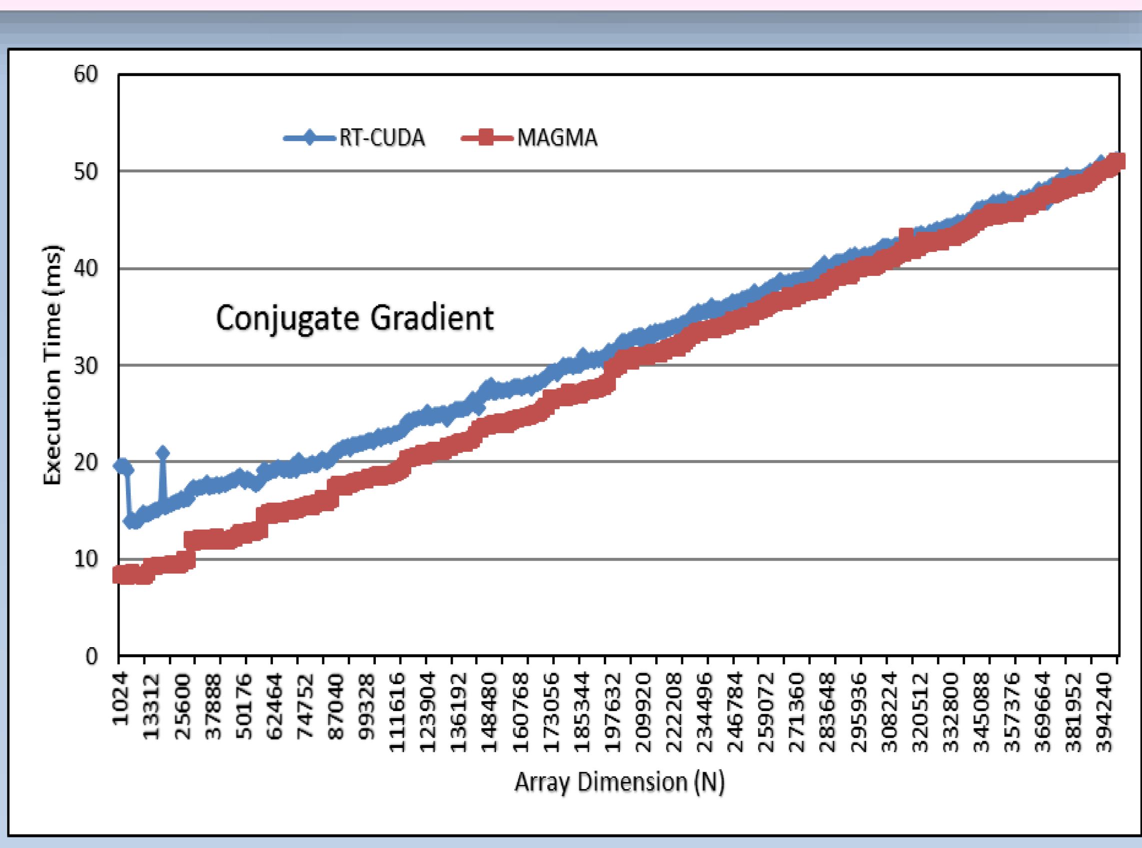
- RT-CUDA supports efficient development of sparse iterative linear solvers such as conjugate gradient to be used in reservoir simulation softwares
- RT-CUDA includes API functions to allocate and initialize sparse matrices with random sparsity as well as reading matrix from matrix market file
- RT-CUDA supports combination of user – defined functions and invoking highly optimized library functions including cuBLAS and cuSparse library functions as shown in the example above
- RT-CUDA hides architectural details of the underlying GPU device that helps traditional C programmers to develop parallel programs in a fast and efficient manner

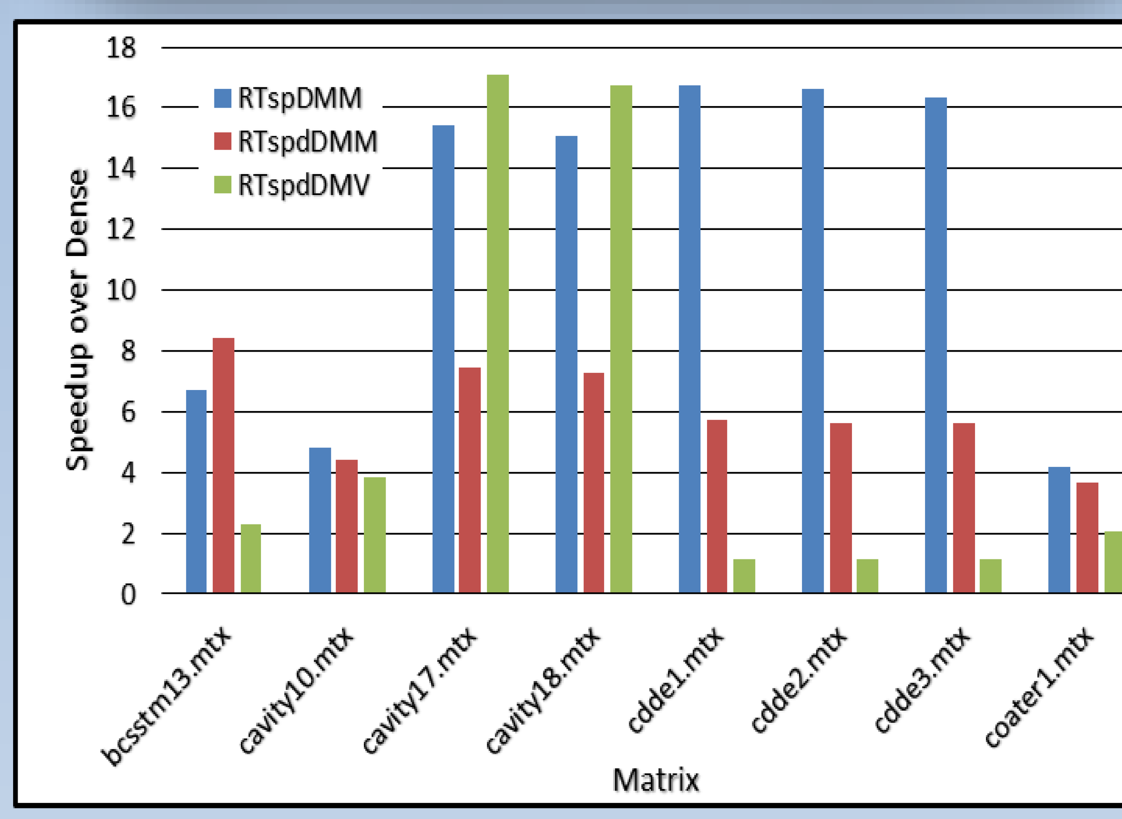
Conclusions

a) **RT-CUDA a software compiler with best possible kernel optimizations to bridge the gap between high-level languages and the machine dependent CUDA and GPUs**
b) **Obtained significant speedup over other compilers like OpenACC and GPGPU compilers**
c) **Enables transparent invocation of the most optimized external math libraries like cuSparse, and cuBLAS. For this, RT-CUDA uses interfacing APIs, error handing interpretation, and user transparent programming**
d) **R-CUDA facilitates the design of efficient parallel software for developing parallel simulators (reservoir simulators, molecular dynamics, etc.) which are critical for Aramco and Oil and Gas industry in KSA**
e) **RT-CUDA needed by many KSA industries dealing with science and engineering simulation on massively parallel computers like NVIDIA GPUs and Intel manycores.**

Benchmarking RT-CUDA using LAPACK, cuSparse, cuBLAS, MAGMA







First Saudi Optimization Compiler for Efficient CUDA Programming on Graphic Processing Unit (GPU)