SPRAT: RUNTIME PROCESSOR SELECTION FOR ENERGY-AWARE COMPUTING

The 3rd International Workshop on Automatic Performance Turning
October 1st , 2008, Tsukuba International Congress Center, EPOCHAL TSUKUBA, Japan

Hiroyuki Takizawa*, Katuto Sato, and Hiroaki Kobayashi
Tohoku University
* tacky@isc.tohoku.ac.jp

OUTLINE

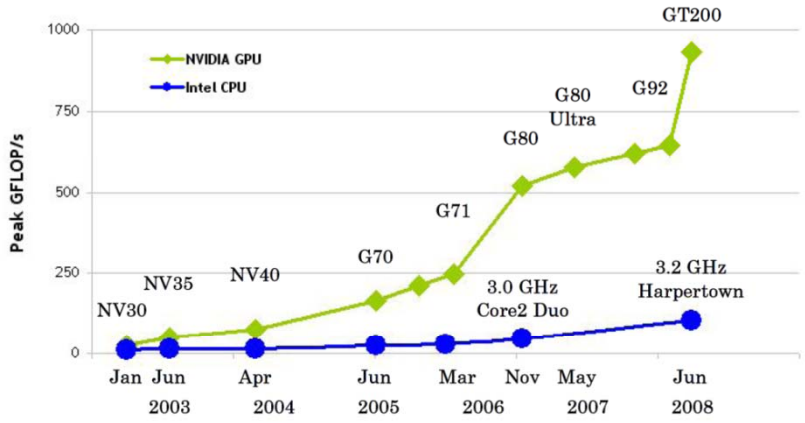
TOHOKU UNIVERSITY

- Introduction
  - PC = A hybrid computing system of CPU and GPU
- Stream Programming with Runtime Auto-Tuning (SPRAT)
  - A Programming Framework with Automatic Processor Selection
- Performance Evaluation
  - Processor Selection for Energy Saving
- Conclusion and Future Work

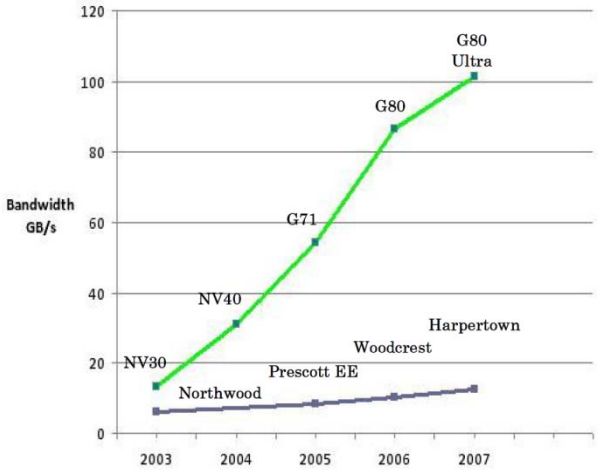
BACKGROUND

TOHOKU UNIVERSITY

- PC is a hybrid computing system of CPU and GPU.
  - CPU and GPU are very different processors.
    - GPU's strengths and weaknesses
      - ☺ Raw performance (GFLOPS)
      - ☺ High memory bandwidth
      - ☹ Poor programming flexibility
      - ☹ Data transfer overhead from/to main memory



Peak Performance (Source: NVIDIA)

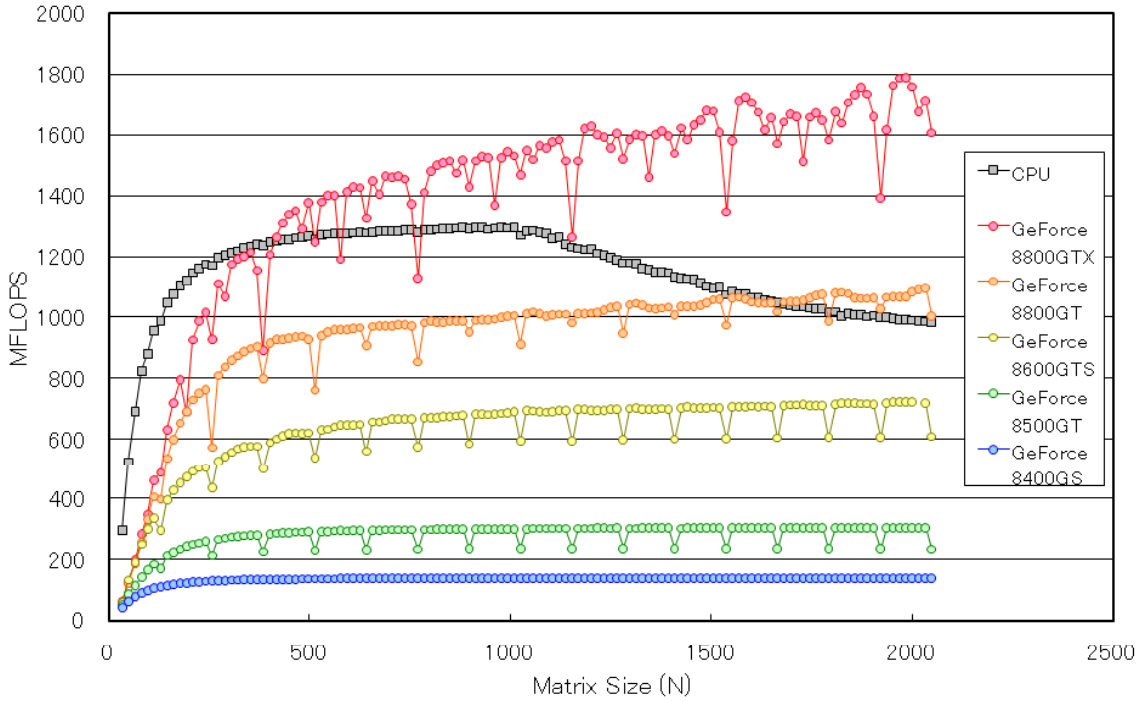


Memory Bandwidth (Source: NVIDIA)

MOTIVATION

TOHOKU UNIVERSITY

- The performance difference between CPU and GPU depends on the problem size.



- The problem size is often determined at runtime.
- The intersection point moves if the hardware configuration changes.

Which processor should be used for execution?

THIS WORK

TOHOKU UNIVERSITY

- Goal: programming framework for hybrid computing systems
  - Programmers do not care about which processor executes the code.
    - Each processor has its own strengths and weaknesses
    - Appropriate selection depends on hardware configuration and problem size.

  ⇨ ***Runtime Processor Selection***

- **SPRAT: Stream Programming with Runtime Auto-Tuning**
  - A SPRAT code is automatically translated into CPU and GPU codes.
  - Runtime processor selection based on performance prediction

GPU may work fast but also increases the power consumption.

⇨ ***GPU should be used only if it works fast enough to outweigh the additional power consumption.***

SPRAT LANGUAGE

TOHOKU UNIVERSITY

- A domain-specific language for stream processing.
  - C with additional keywords for stream processing
    - Similar to BrookGPU (Buck et al, 2004)
      - does not need any architecture-specific description
    - The execution time of a kernel is predictable.
      - Linear to the number of stream elements.

SPRAT decides which processor executes kernels.

SPRAT CODE

TOHOKU UNIVERSITY

- **Kernel**

  - Special function running on either CPU or GPU
    - `kernel map copy(…)`

- **Stream**

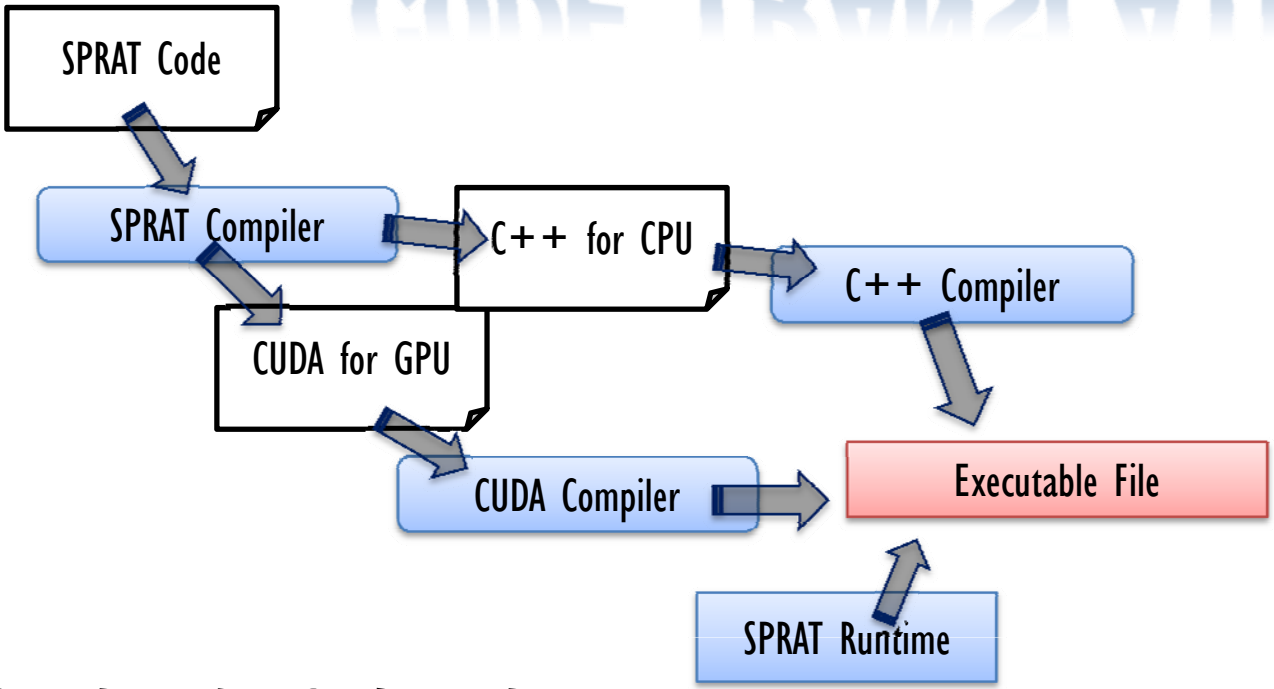
  - Data container accessed by kernels
    - `stream <float> sa, sb`

- **Data management**

  - Copy data from/to streams
    - `streamRead, streamWrite`

```
kernel map copy(in stream<float> sa,
                out stream<float> sb)
{
    sb = sa;
}

int main(int argc, char **argv)
{
    float aa[N][N];
    float ab[N][N];
    stream<float> sa(N,N);
    stream<float> sb(N,N);

    InitializeArray(aa);
    streamRead(sa, aa);

    copy(sa, sb);

    streamWrite(sb, ab);
    PrintArray(ab);

    return 0;
}
```

CODE TRANSLATION



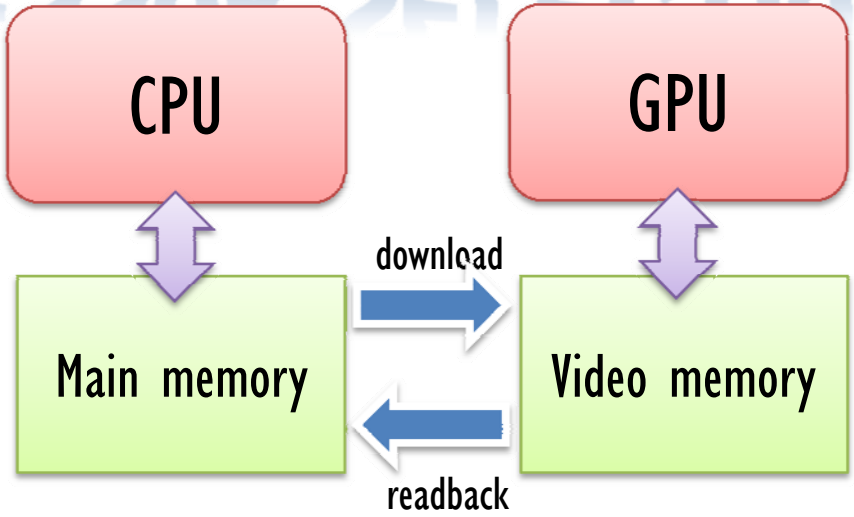
Sample code of a kernel

```
kernel map filter(gather stream<float> source, out stream<float> dest, float coeff)
{
    dest = source[-1][-1] + source[-1][ 0] + source[-1][ 1]
         + source[ 0][-1] + coeff * source + source[ 0][ 1]
         + source[ 1][-1] + source[ 1][ 0] + source[ 1][ 1];
    return;
}
```

A simple code is automatically translated into CPU and GPU codes.

Provides the opportunity to select which processor runs the code.

PROCESSOR SELECTION (1/2)



- Processor should be switched only if the total energy consumption reduces.
  - The data transfer increases the execution time and hence the energy consumption.
    - Frequent processor switching and data transfer will increase the energy consumption.
  - GPU must be much faster than CPU.
    - GPU generally needs more power than CPU!

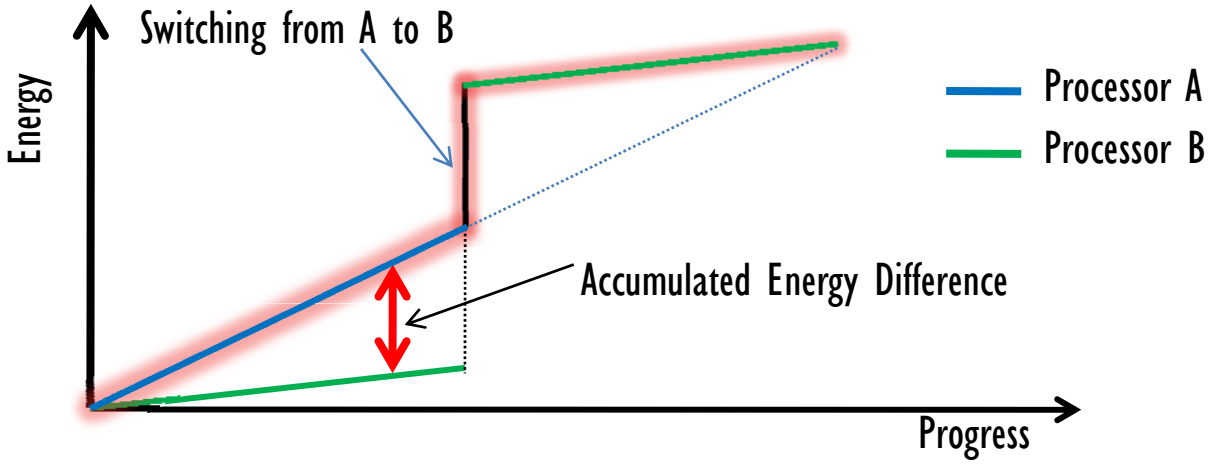
Is GPU fast enough to outweigh the power consumption and the data transfer overhead?

PROCESSOR SELECTION (2/2)

- Assumptions
  - Power consumption of each processor is constant for any kernel.
  - Kernel execution time is in proportion to the number of stream elements.
- Accumulated Energy Difference (AED)
  - How much the other processor can save the energy.

$$\tau_p \leftarrow \max\{\tau_p + (P_{q,ki}T_{q,ki} - P_{p,ki}T_{p,ki}), 0\}$$

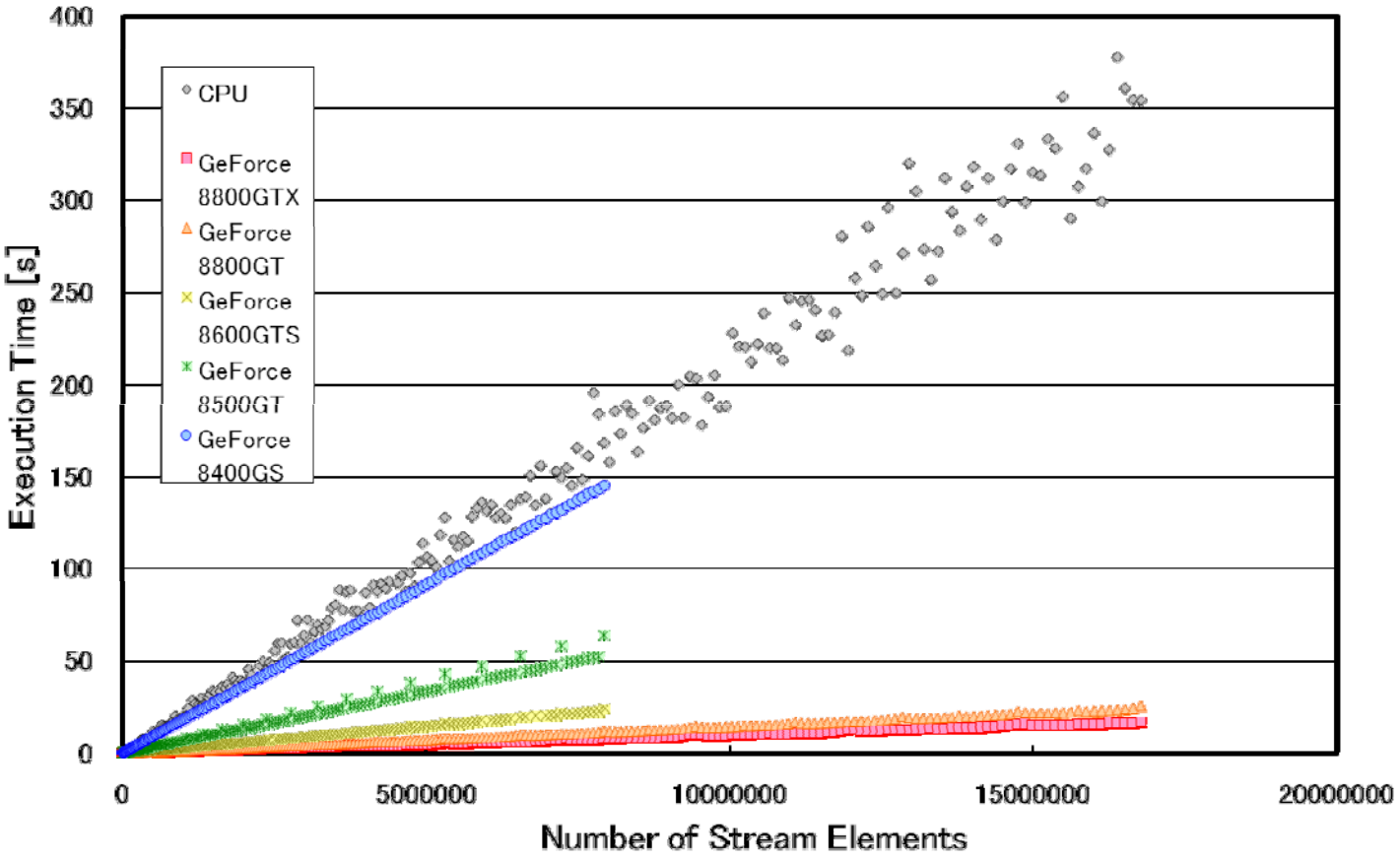
SPRAT switches the processor if AED exceeds extra energy consumption for data transfer.



PERFORMANCE MODELING

TOHOKU UNIVERSITY

- Kernel execution time is proportional to the stream length.



SYSTEM CONFIGURATION

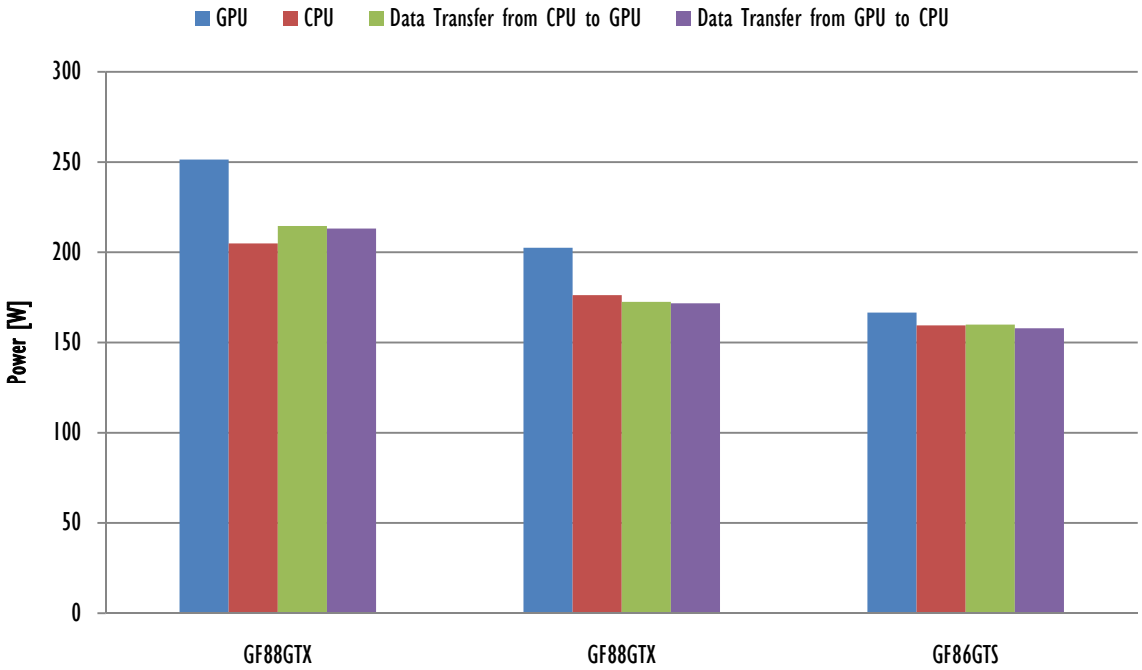
- Evaluated System
  - OS: CentOS (driver version 173.13)
    - C++ compiler: gcc 4.2.1
    - CUDA compiler: nvcc-release 1.1 V0.2.1221
  - CPU: Intel Core 2 Quad Q6600 (2.4GHz)
  - Mem: DDR2-800 4GB
  - GPU:

Model	# SMs	Mem [MB]	BW [GB/s]
GeForce 8800GTX	16	768	86.4
GeForce 8800GT	14	512	57.6
GeForce 8600GTS	4	256	32.0

POWER CONSUMPTION

TOHOKU UNIVERSITY

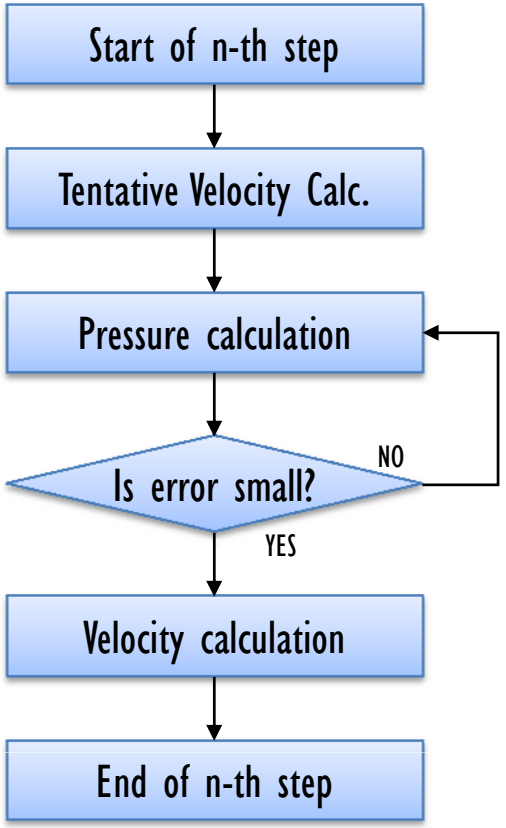
- Power Consumption of Each System Configuration
  - Each processor is loaded iteratively running the SAXPY kernel.
  - HIOKI HiTESTER 3334 is used for the measurement.



APPLICATION (1/2)

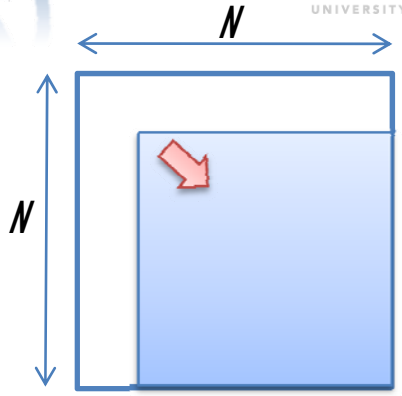
TOHOKU UNIVERSITY

- 2-dimencional incompressible fluid simulation ($N \times N$)
  - Fractional Step Method
    - consists of 8 kernels.
      - Most kernels are simple stream computations
    - Stream size of each kernel does not change.
      - The performance is predictable from the history.



GPU will work well for this application.
SPRAT will switch the processor from CPU to GPU.

APPLICATION (2/2)



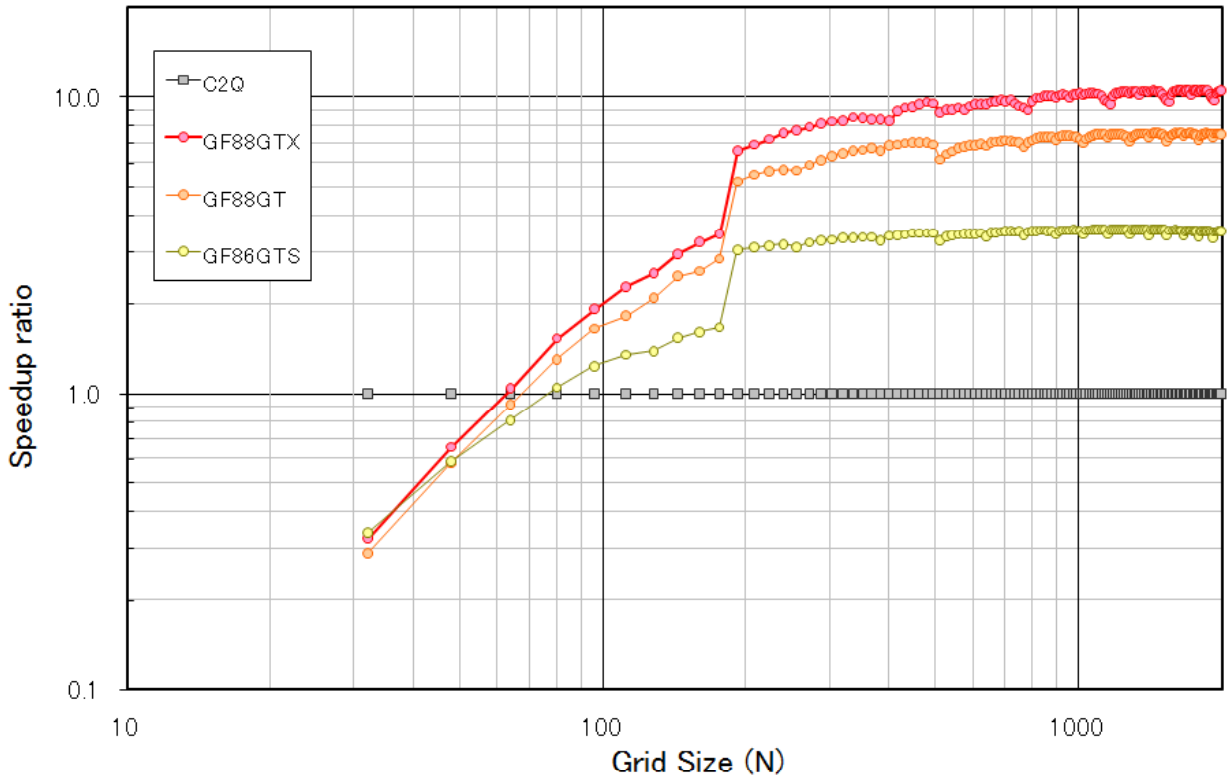
- LU decomposition
  - needs non-coalesced memory accesses
    - not suitable for GPU to run = CPU should run kernels.
  - The first kernel invocation is the most time-consuming
    - Stream size decreases at every kernel invocation.
    - GPU should be used only in the early stage of the execution.
      - But SPRAT does not switch the processor until AED exceeds data transfer overhead.

This application has disadvantageous features for GPU and SPRAT.

RESULTS FOR CFD

TOHOKU UNIVERSITY

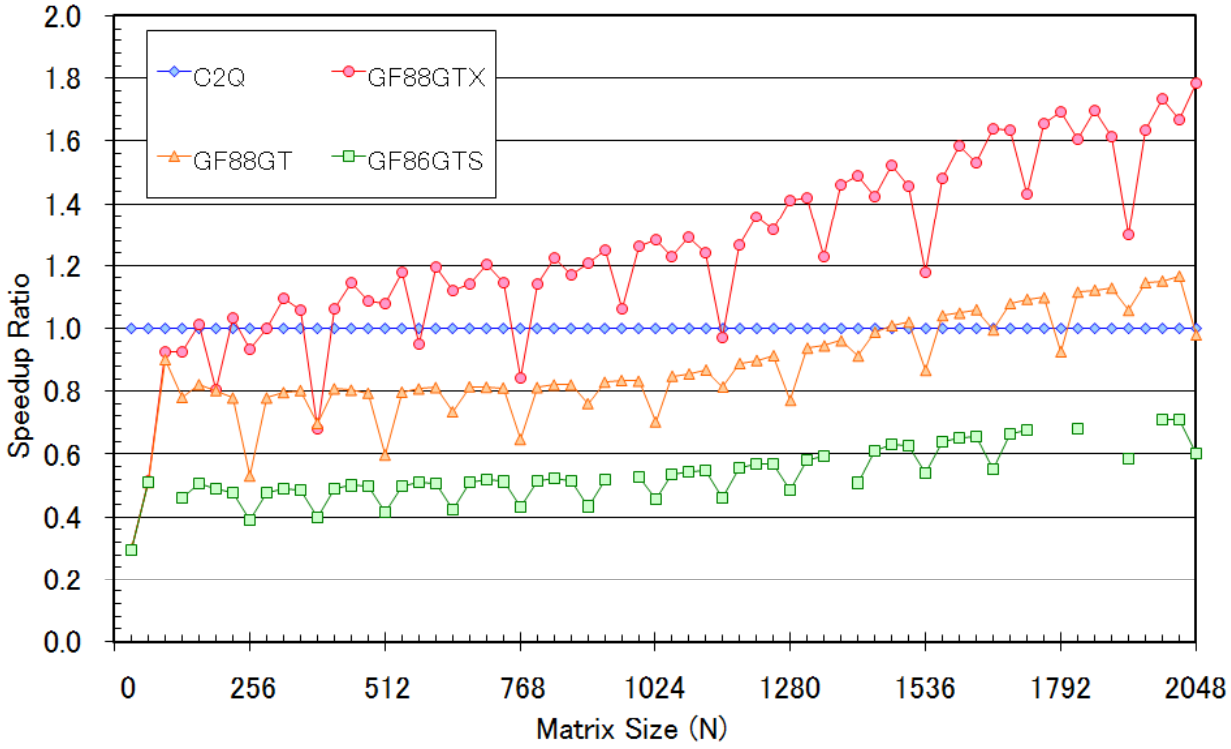
- GPUs improve the energy efficiency.



RESULTS FOR LU (1/2)

TOHOKU UNIVERSITY

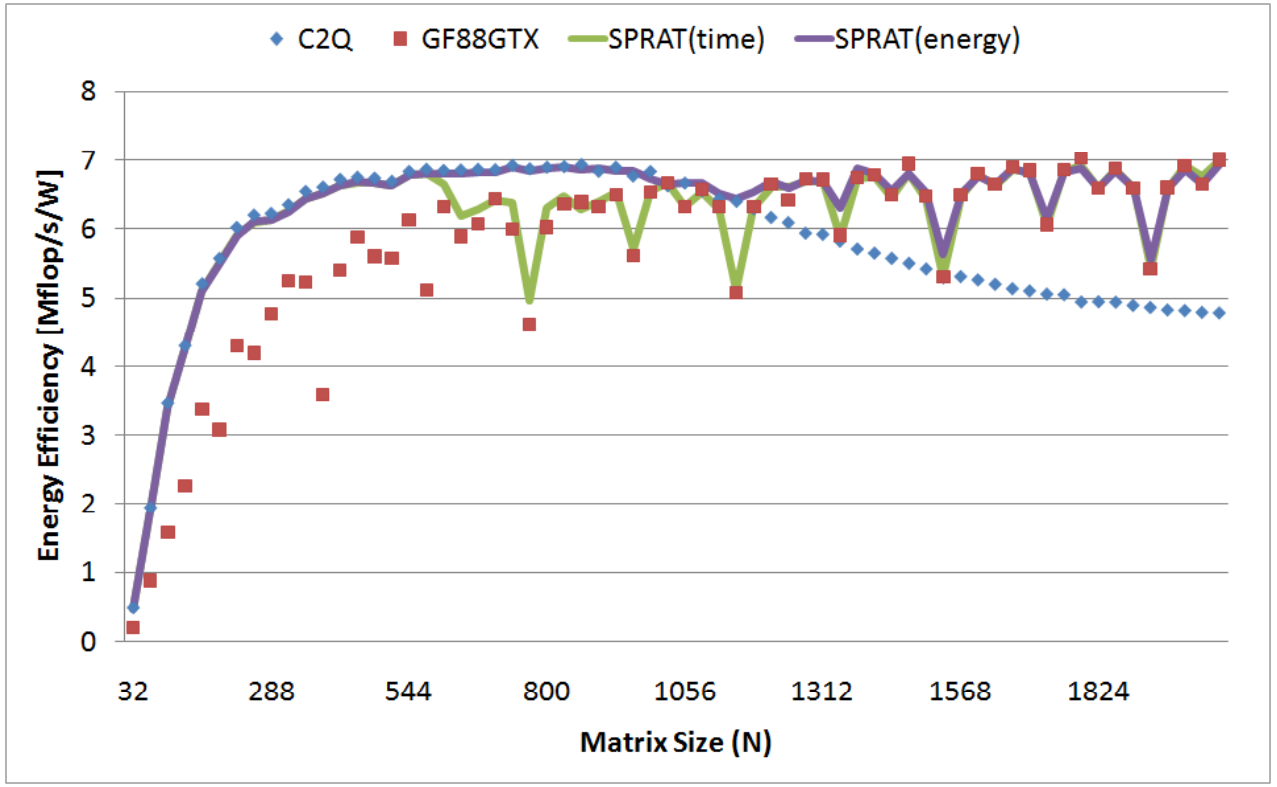
- Appropriate processor selection depends on the matrix size.



RESULTS FOR LU (2/2)

TOHOKU UNIVERSITY

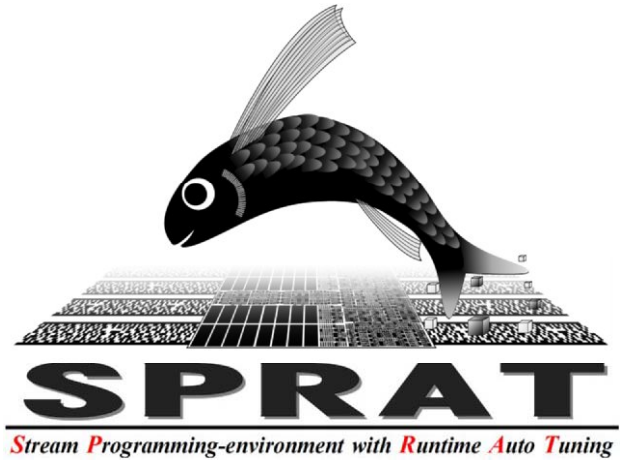
- SPRAT can select a more energy efficient processor for each matrix size.



CONCLUSION

TOHOKU UNIVERSITY

- **SPRAT: Stream Programming with Runtime Auto-Tuning**
  - SPRAT compiler automatically generates CPU and GPU codes.
  - SPRAT dynamically selects which processor runs the kernel.
    - The processor is switched from CPU to GPU only if improving the energy efficiency.



SPRAT is promising to automatically improve the energy efficiency of hybrid computing systems.

FUTURE WORK

TOHOKU UNIVERSITY

- Automatic optimization techniques
  - enable SPRAT compiler to generate more efficient code.
- Performance/energy prediction models
  - enable SPRAT to predict the performance/energy more accurately.
- Other accelerators
  - Cell broadband engine? FPGA? ClearSpeed? Vector Machines?

QUESTIONS?

TOHOKU UNIVERSITY

- Acknowledgments

The authors would like to thank Prof. Kentaro Sano of Tohoku University for valuable discussions on energy efficiency, and also Prof. Reiji Suda of the university of Tokyo for helpful comments on performance modeling.

This research was partially supported by Grants-in-Aid for Young Scientists(B) #19700020 and Scientific Research on Priority Areas #18049003, and by Strategic Information and Communications R&D Promotion Program (SCOPE-S) #061102002.