Thinking Outside of the Tera-Scale Box

Piotr Luszczek

Brief History of Tera-flop: 1997

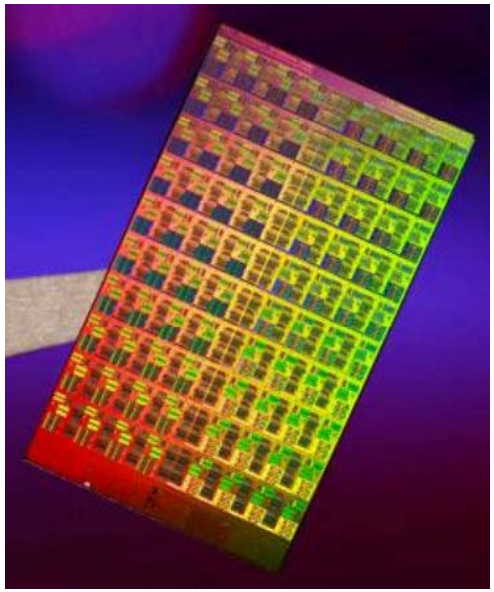
1997

ASCI Red

Brief History of Tera-flop: 2007

Intel Polaris

2007



1997

ASCI Red

Brief History of Tera-flop: GPGPU

GPGPU

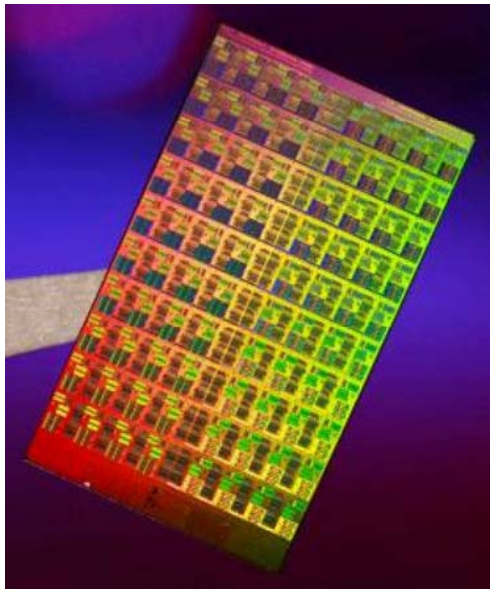
Intel Polaris

2007

2008

1997

ASCI Red



Tera-flop: a Dictionary Perspective

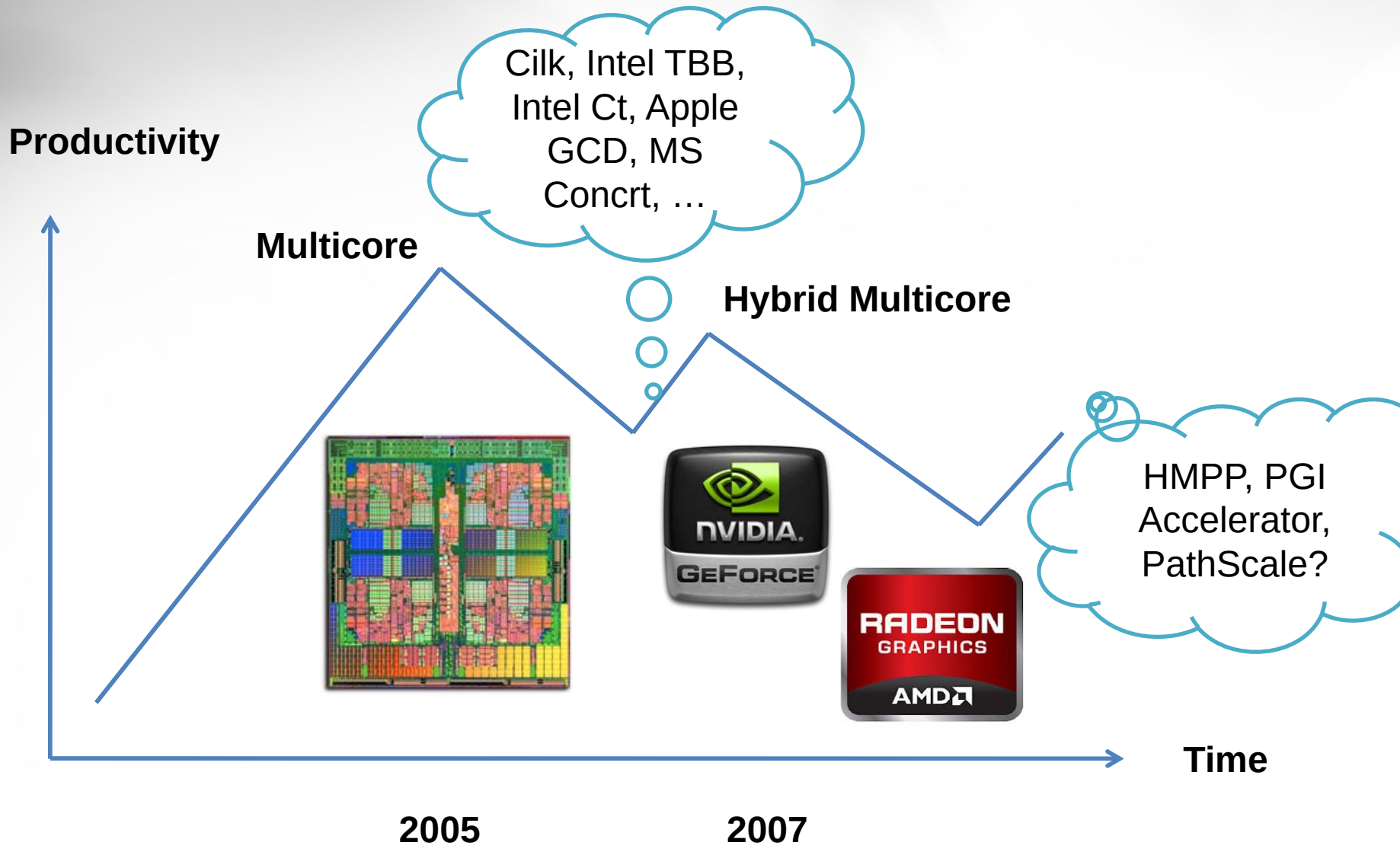
- Belongs to supercomputer expert
- Consumes 500 kW
- Costs over $1M
- Requires distributed memory programming

- Memory: 1 GiB or more

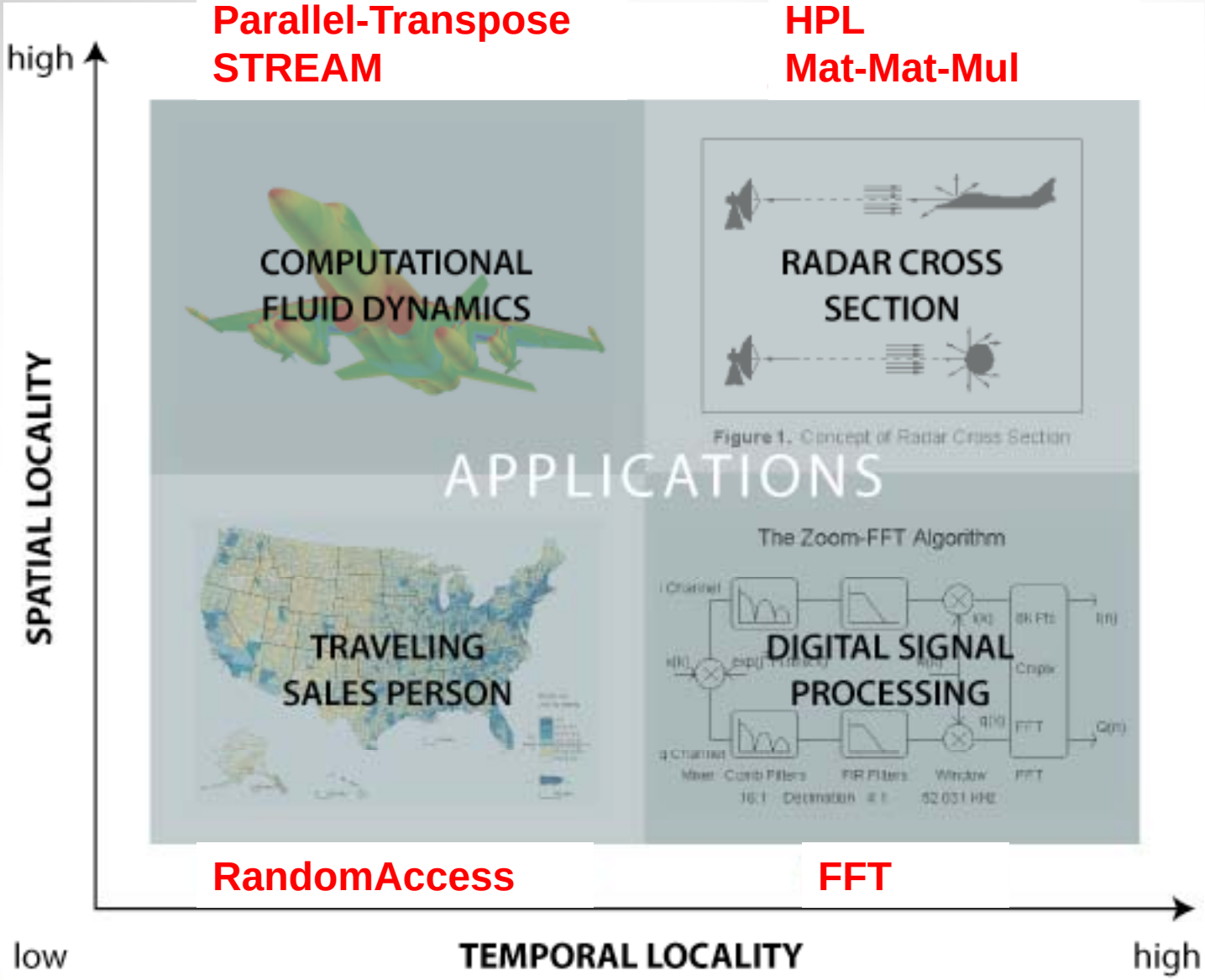
- Belongs to gaming geek
- Consumes under 500 W
- Costs under $200
- Requires only a CUDA kernel

- Memory: 1 GiB or more

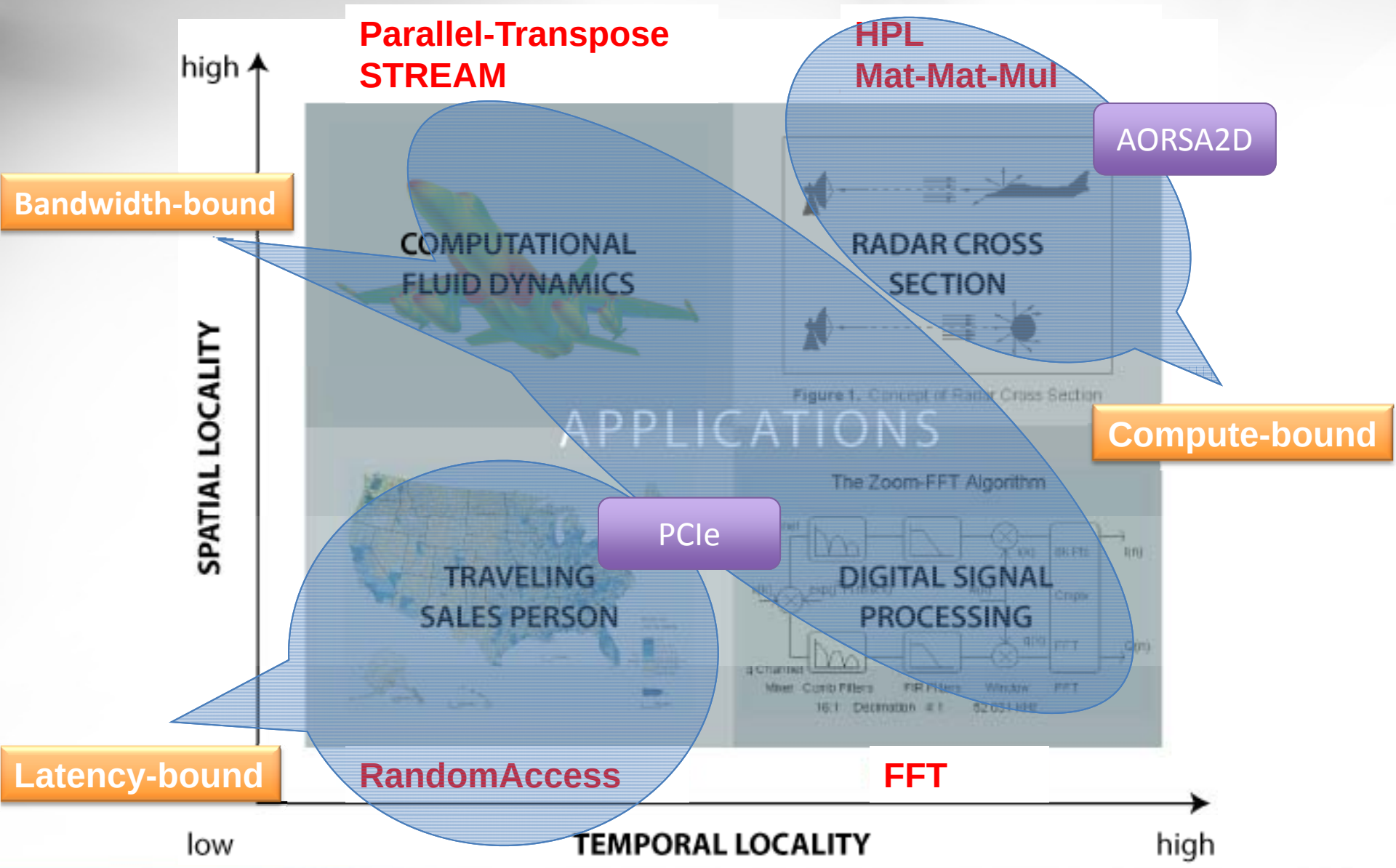
Double-Dip Recession?



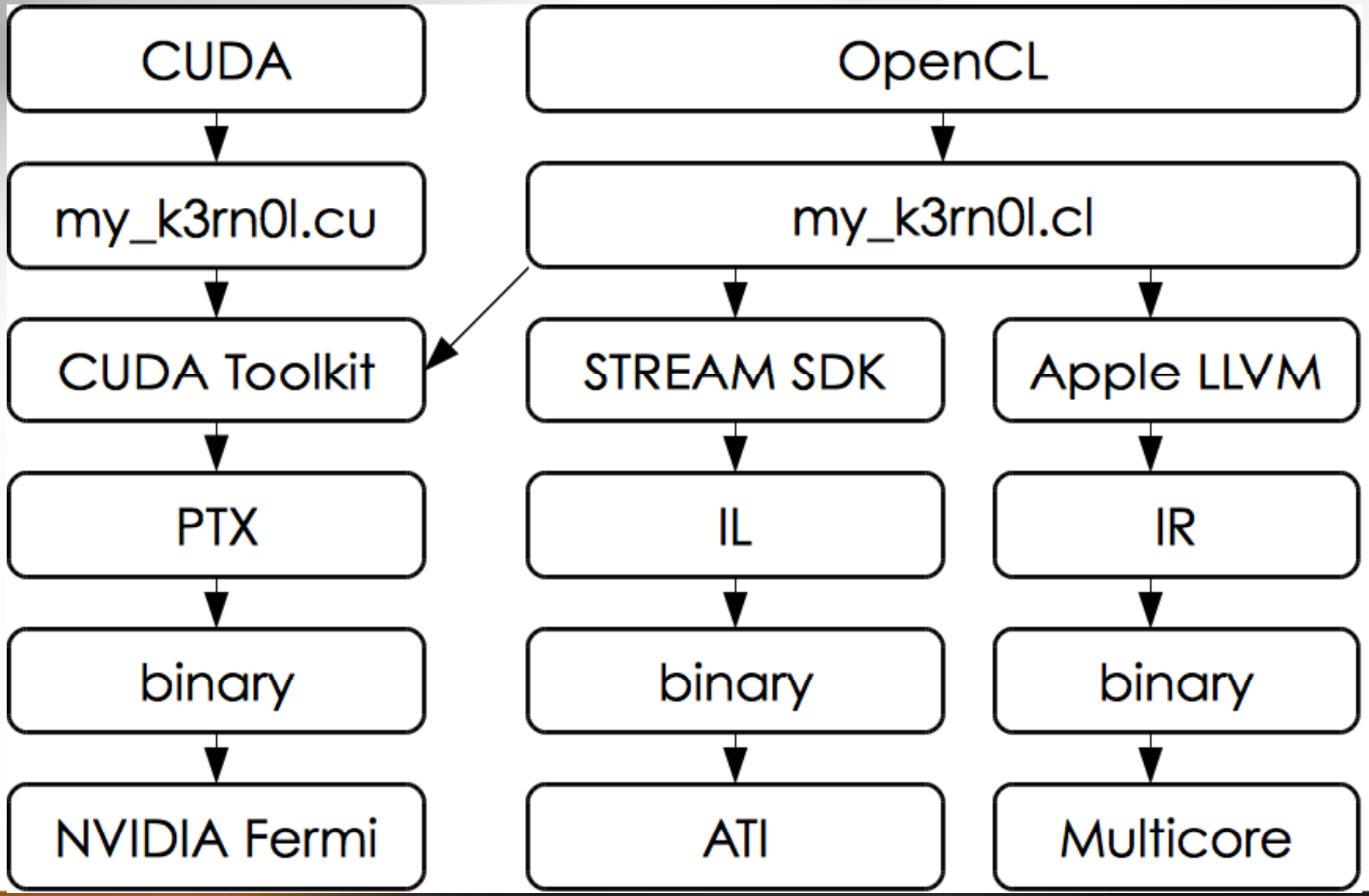
Locality Space in HPC Challenge



Locality Space in HPC Challenge: Bounds

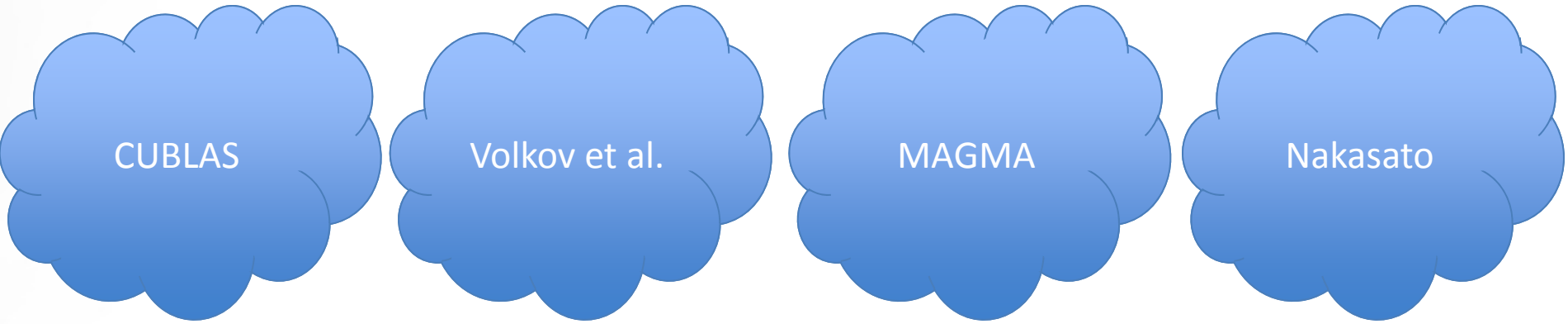


CUDA vs. OpenCL: Similar Software Stacks

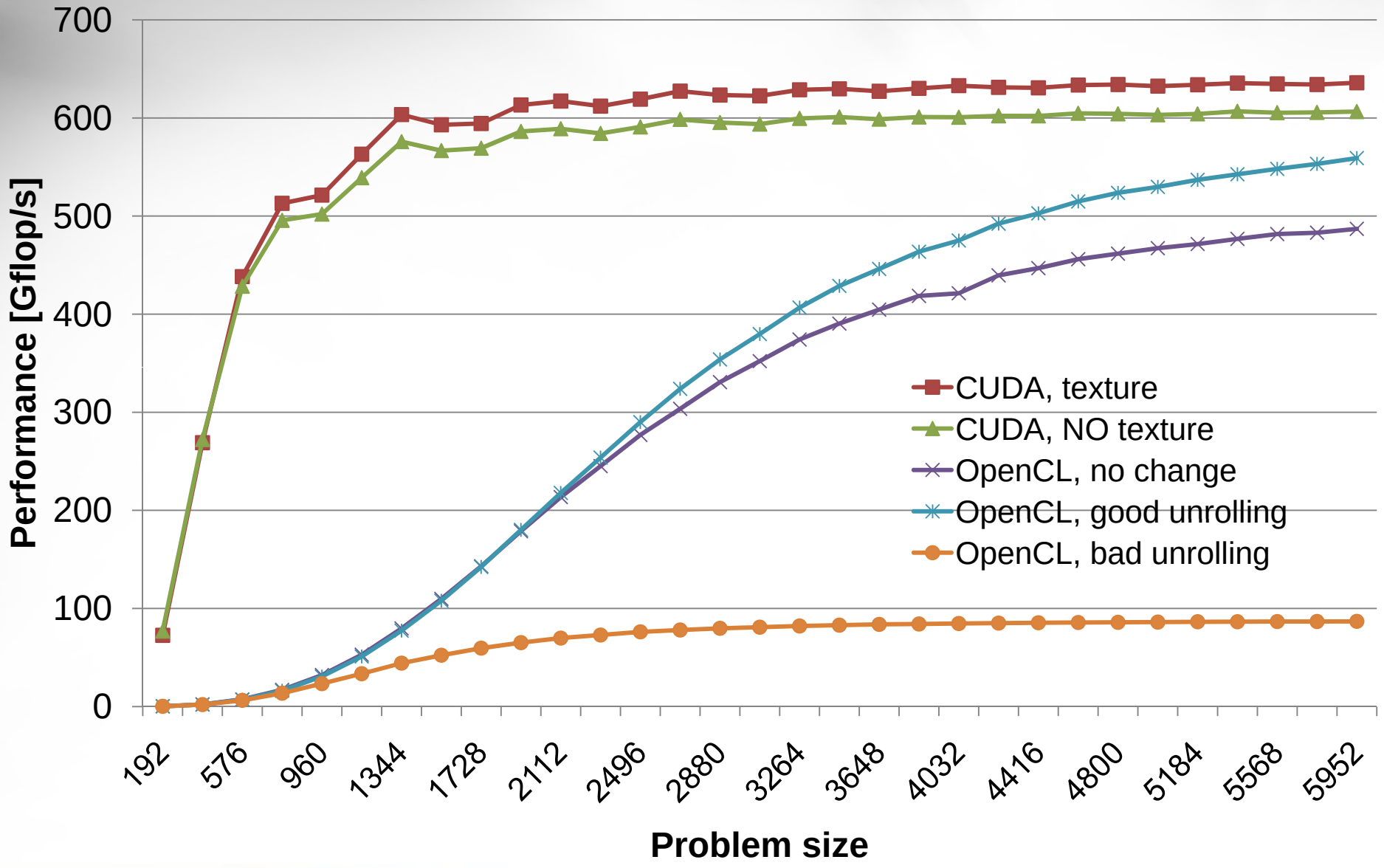


Matrix-Matrix Multiply: the Simple Form

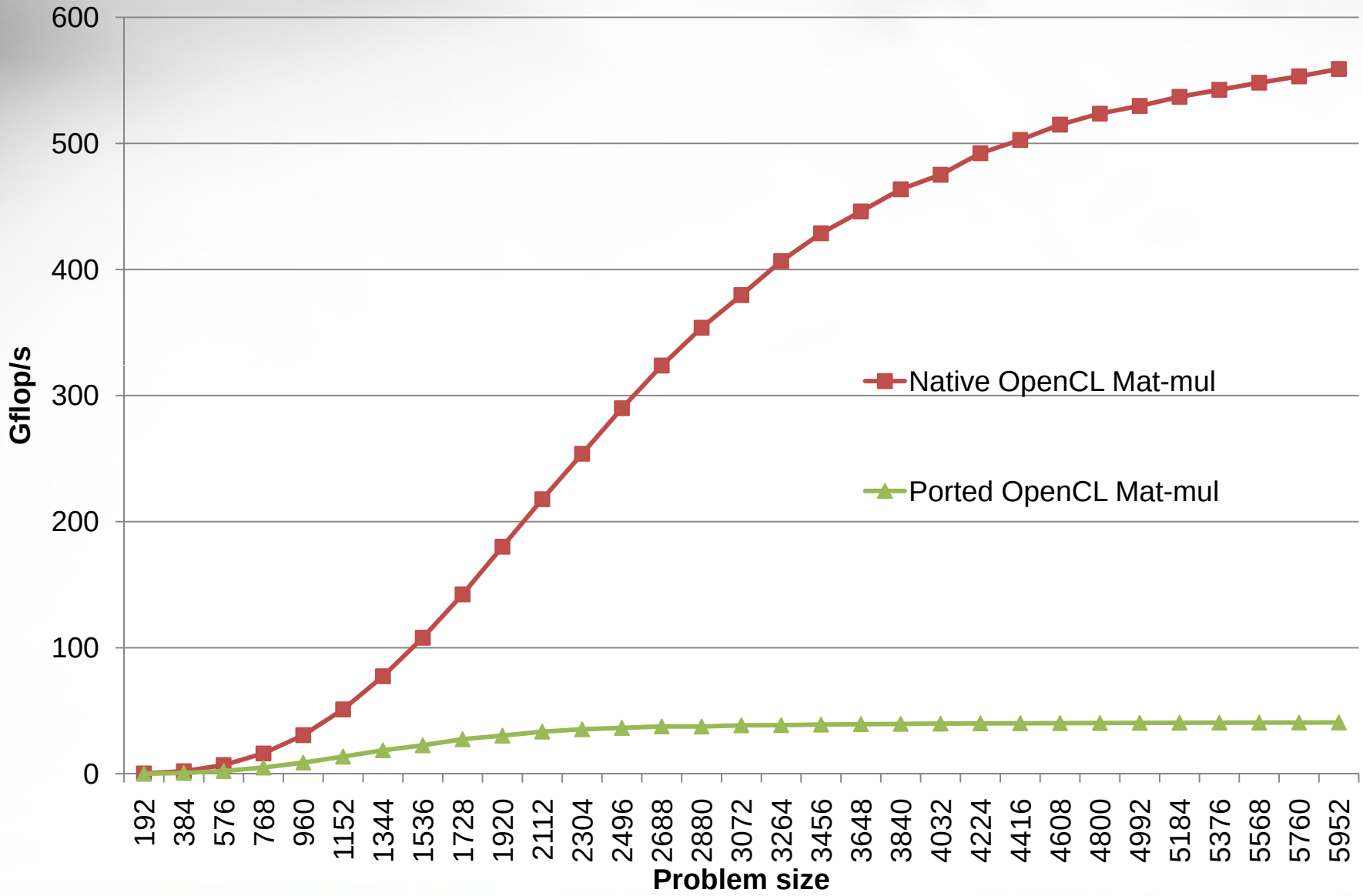
```
for i = 1:M
  for j = 1:N
    for k = 1:T
      C(i, j) += A(i, k) * B(k, j)
```



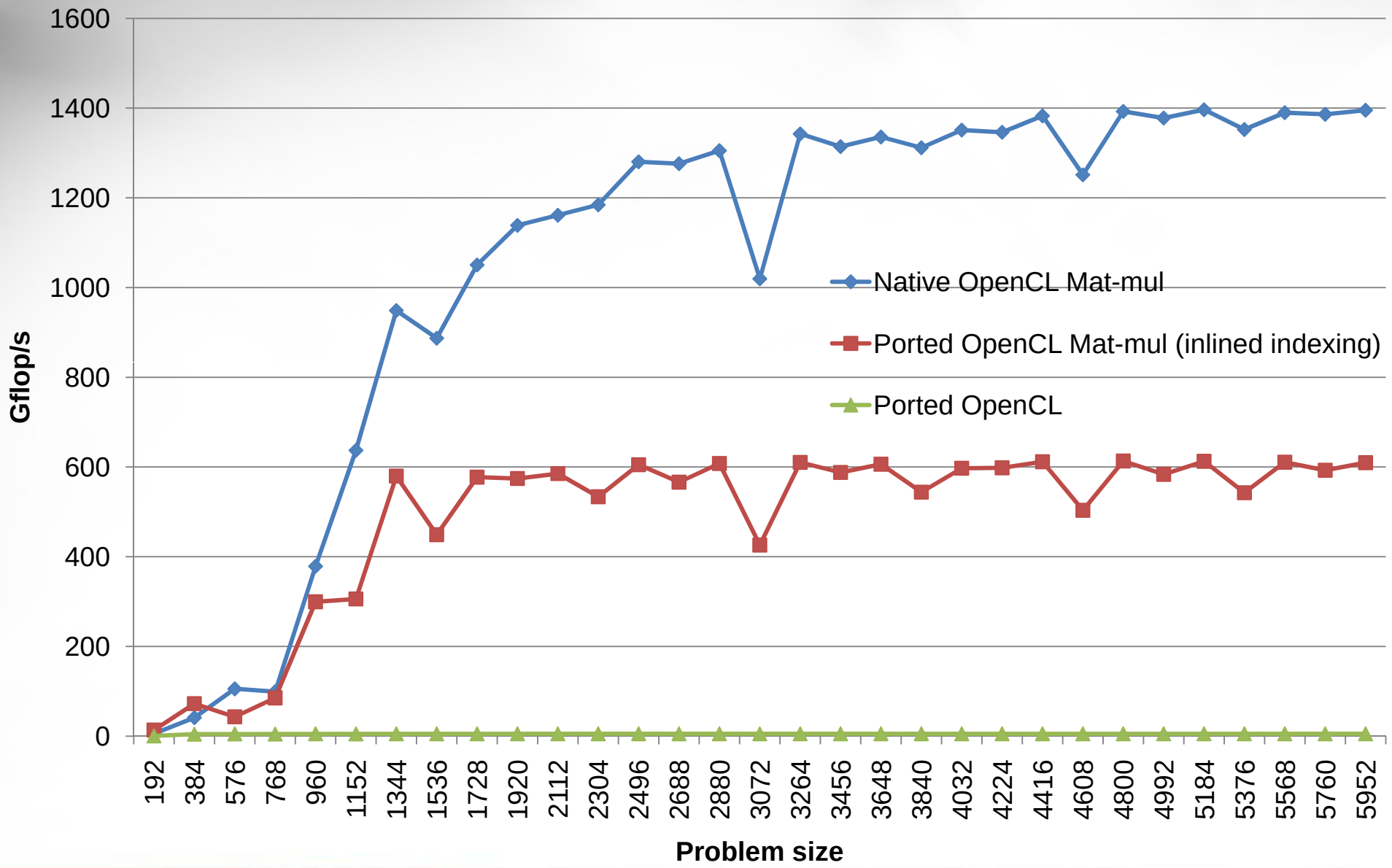
Fermi C2050: from CUDA to OpenCL



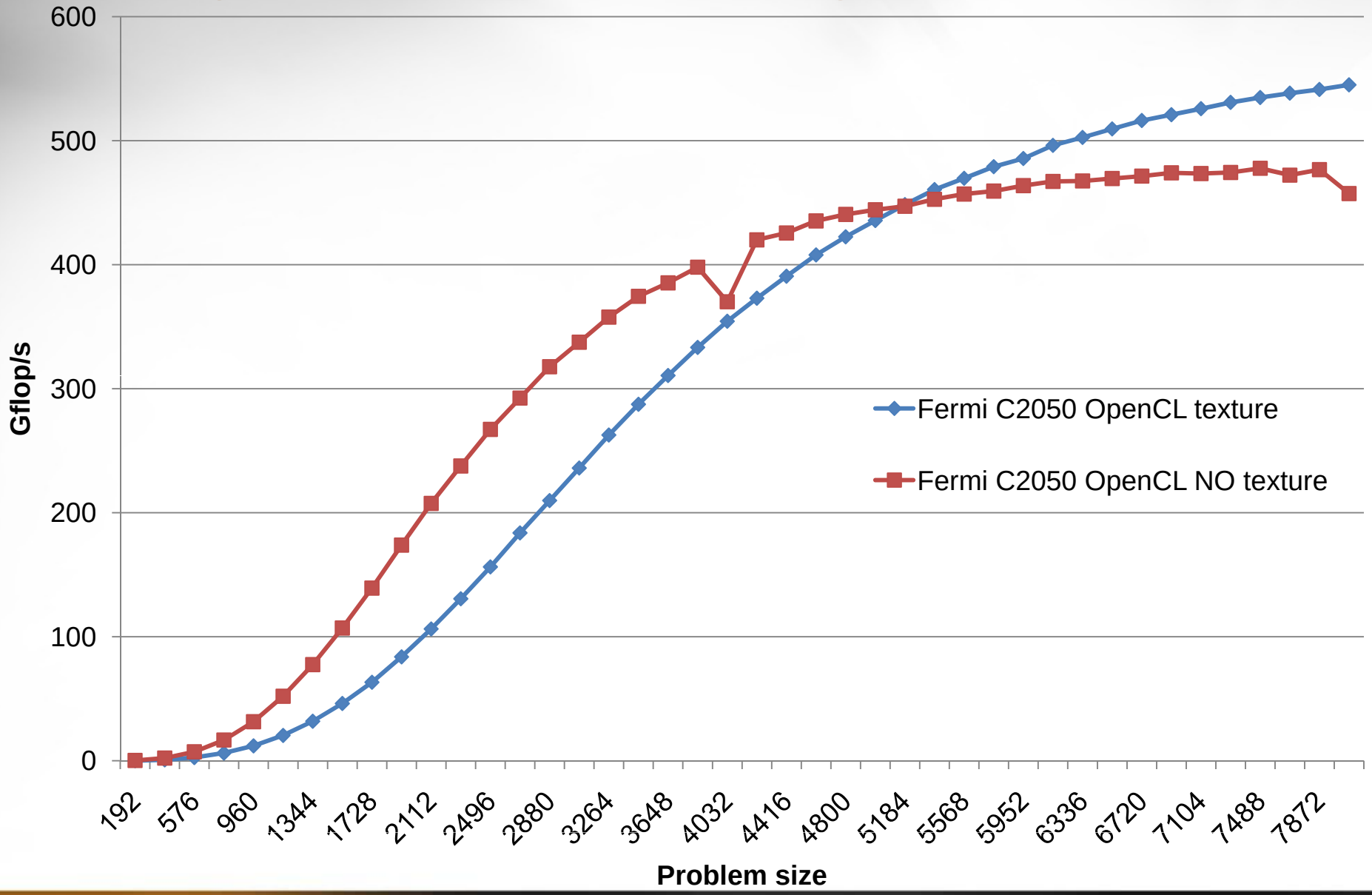
Fermi C2050: Simple OpenCL Porting



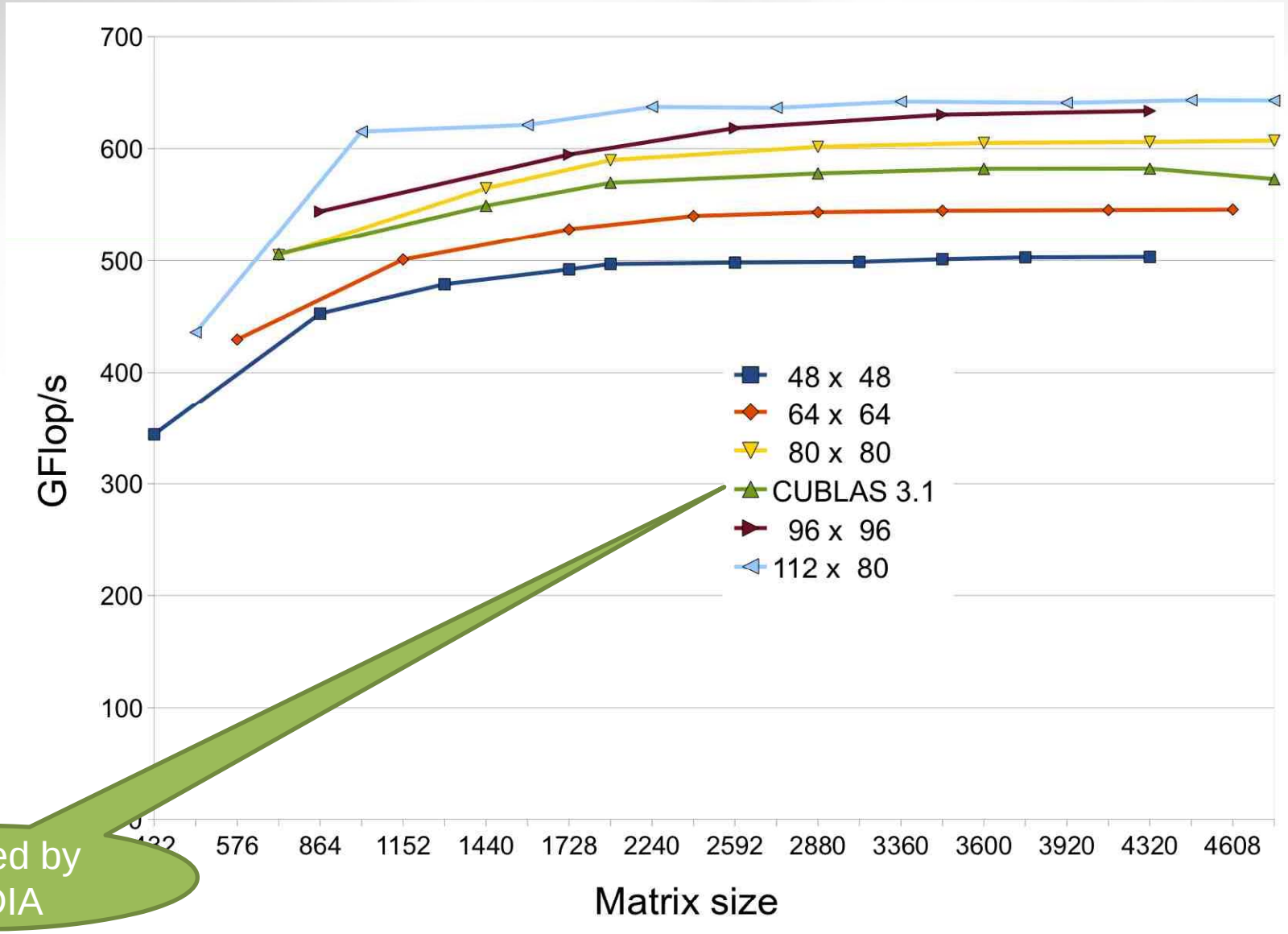
ATI 5870: Simple OpenCL Porting



Making Case for Autotuning: Use of Texture



Making Case for Autotuning: Blocking Factors



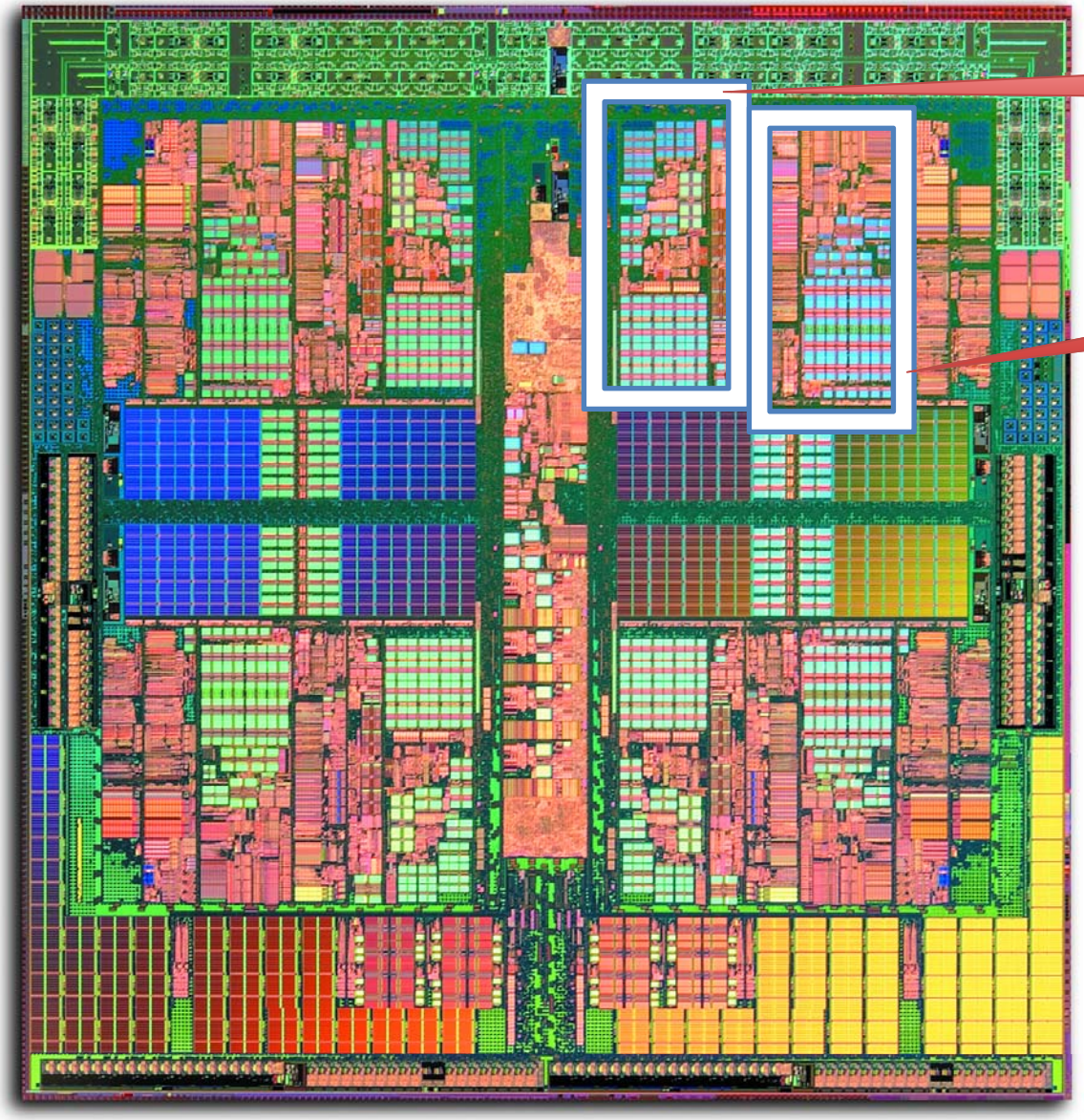
Provided by NVIDIA

Mat-Mul Portability Summary

	Achieved Performance Single precision	Achieved Performance Double precision
ATI OpenCL	52%	57%
ATI IL	74%	87%
NVIDIA CUDA	63%	58%
NVIDIA OpenCL	57%	49%
Multicore (vendor)	79%	79%
Multicore (autotuned)	46%	40%

Now, let’s think outside of the box

Recipe for Performance Before Multicore



Load/Store

Instr. scheduling

Wider superscalar instr. issue

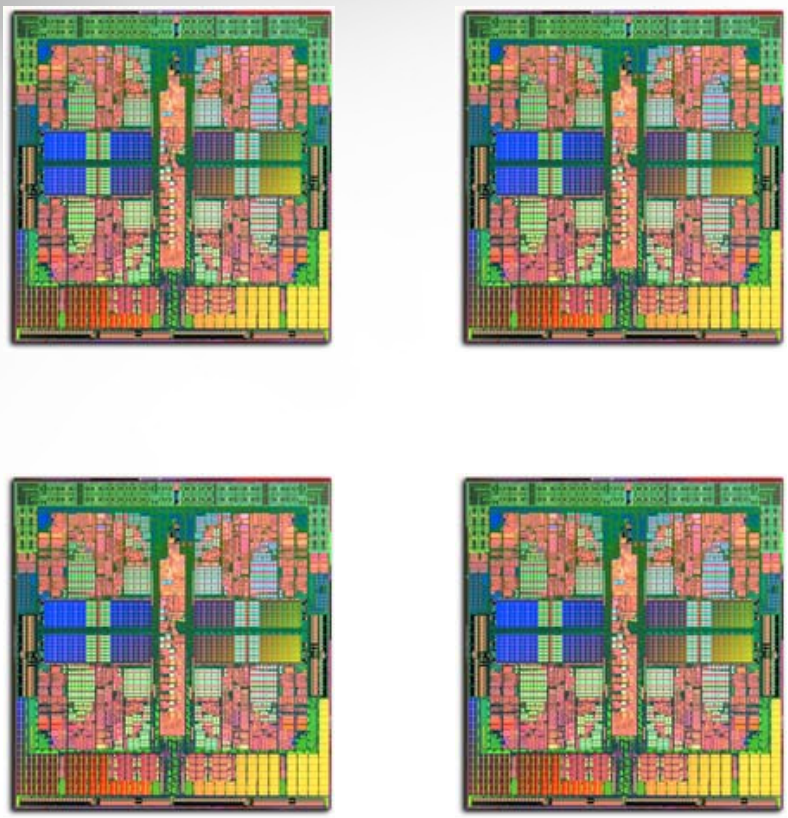
+

Increase clock frequency

+

Increase number of transistor

Recipe for Performance After Multicore

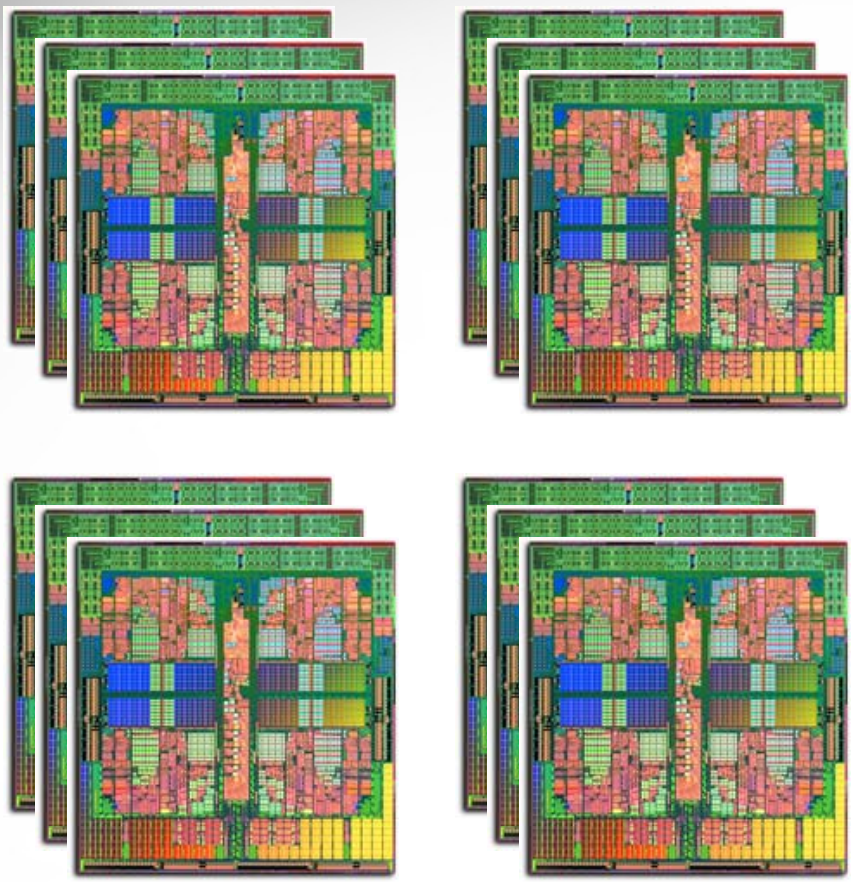


More cores

+

Parallel software

Recipe for Future Performance



More cores

+

Better sequential software

+

DAG Runtime

From Assembly to DAG: PLASMA

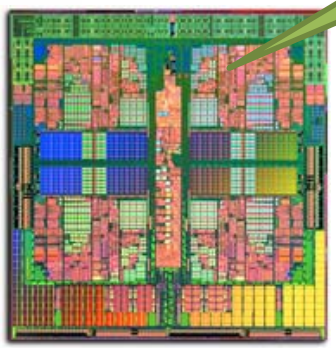
- Typical loop in assembly
  - load R0, #1000
  - divf F1, F2, F3
  - load R1, #1000
  - divf F2, F4, F5
  - dec R1
  - jump …

Sequential instruction stream

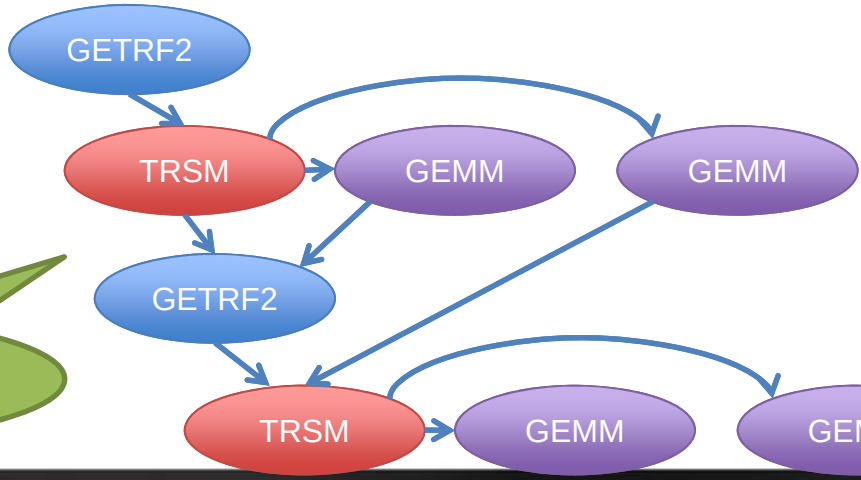
- LU factorization
  - for i = 1:N
  - GETR2(A[i, i])
  - for j = i+1:N
  - TRSM(A[i, i], A[i, j])
  - for k = i+1:N
  - GEMM(A[k,i],A[i,j],A[k,j])

Sequential task stream

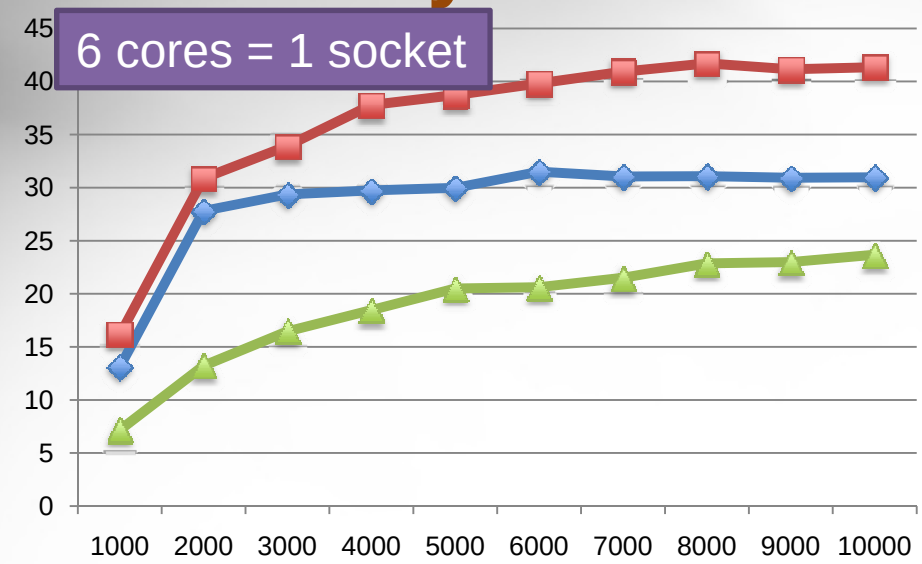
Instr. scheduling

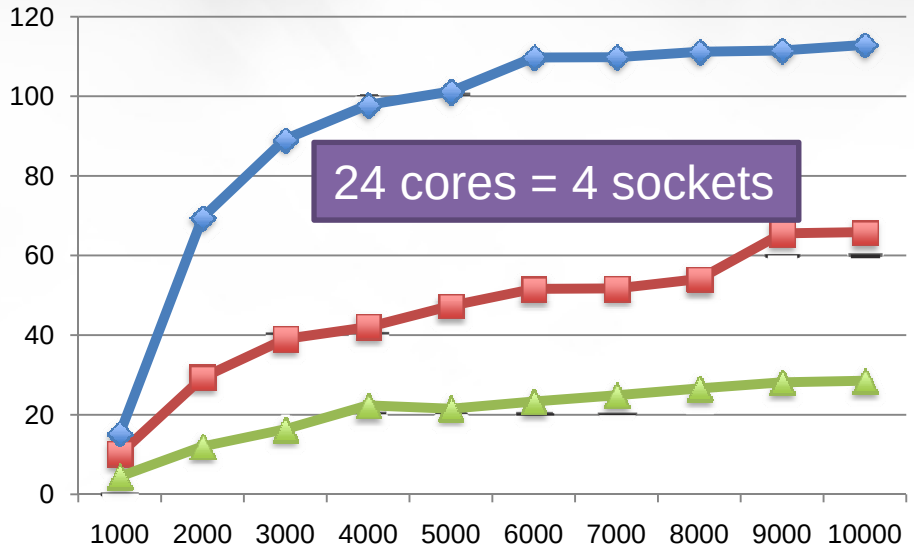


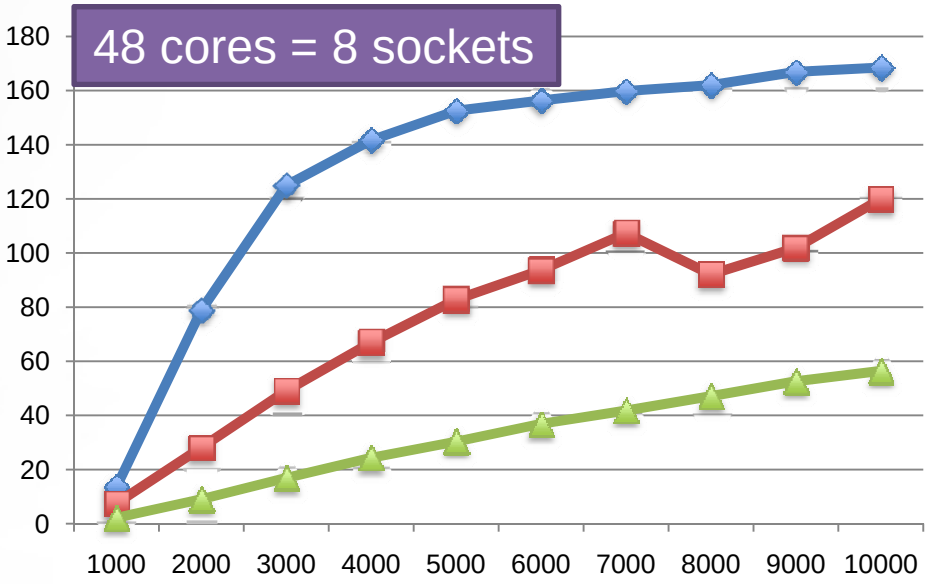
DAG scheduling



Scalability of LU Factorization on Multicore







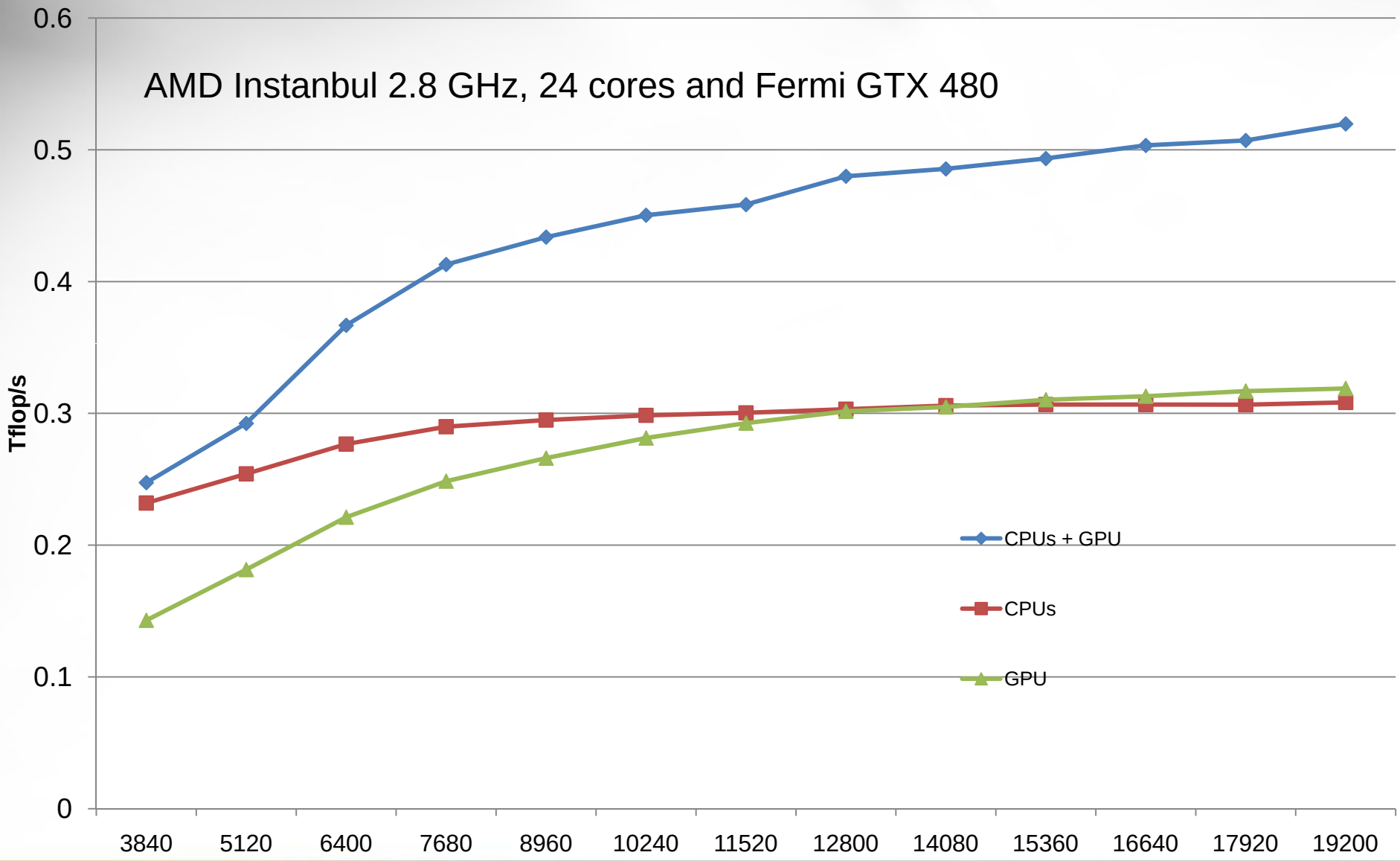
PLASMA

MKL 11.0

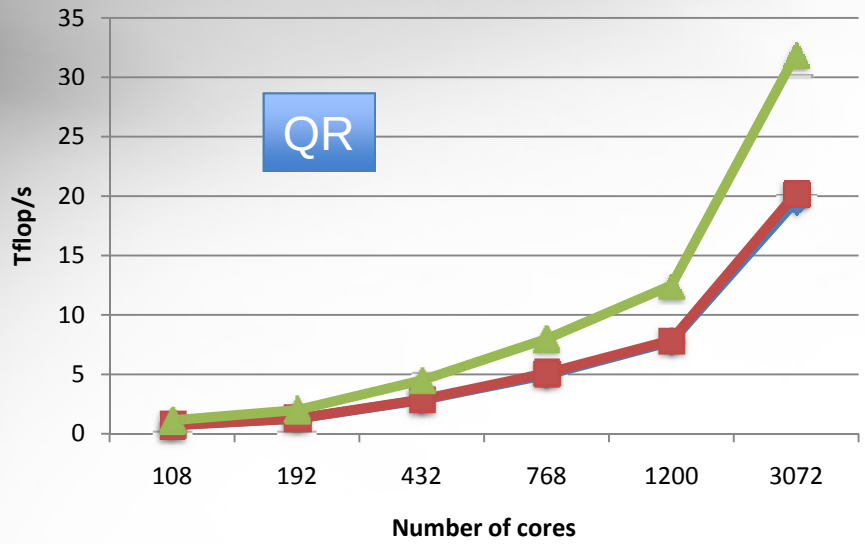
LAPACK

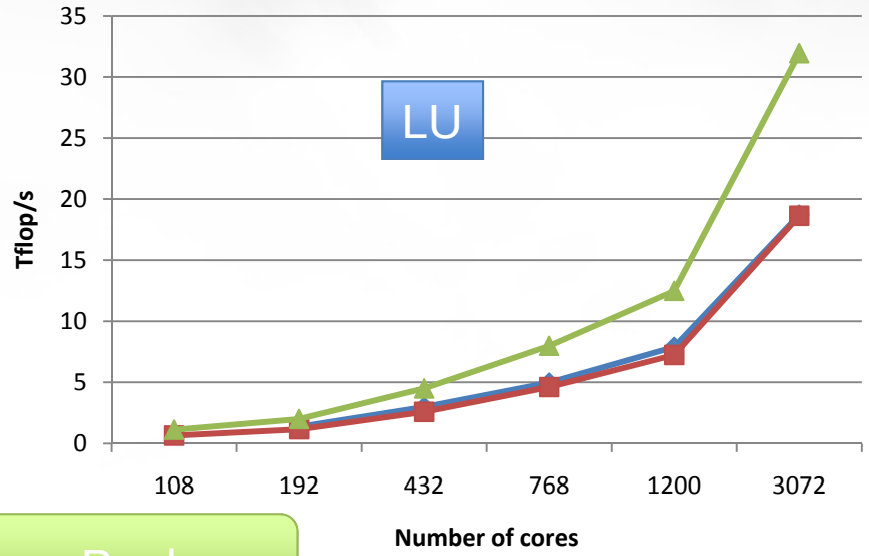
AMD Istanbul 2.8 GHz

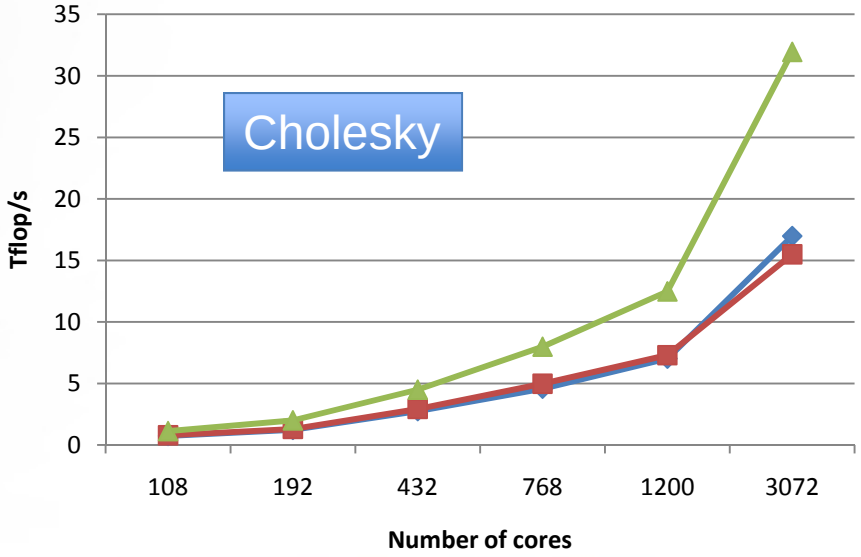
Combining Multicore and GPU for QR Fact.



Performance on Distributed Memory







Peak

DPLASMA

Cray LibSci

ORNL Kraken
Cray XT5
AMD Istanbul 2.6 GHz
Dual-socet 6-core
Interconnect: Seastar, 3D torus

People and Projects

- Jack Dongarra
- Mathieu Faverge
- Jakub Kurzak
- Hatem Ltaief
- Stan Tomov
- Asim YarKhan
- Peng Du
- Rick Weber

- MAGMA
- PLASMA
- DPLASMA and DAGuE
  - George Bosilca
  - Aurelien Bouteiller
  - Anthony Danalis
  - Thomas Herault
  - Perre Lemarinier