

Performance Tuning on Heterogeneous HPC System

Haibo Xie, Ph.D xiehb@inspur.com

Manager, Application Promotion Division

HPC business, Inspur Group

Outline

- Heterogeneous HPC
 - Inpsur's viewpoint
- Performance tuning
 - a case study
- Summary

Inspur today



Employee
Over 10,000

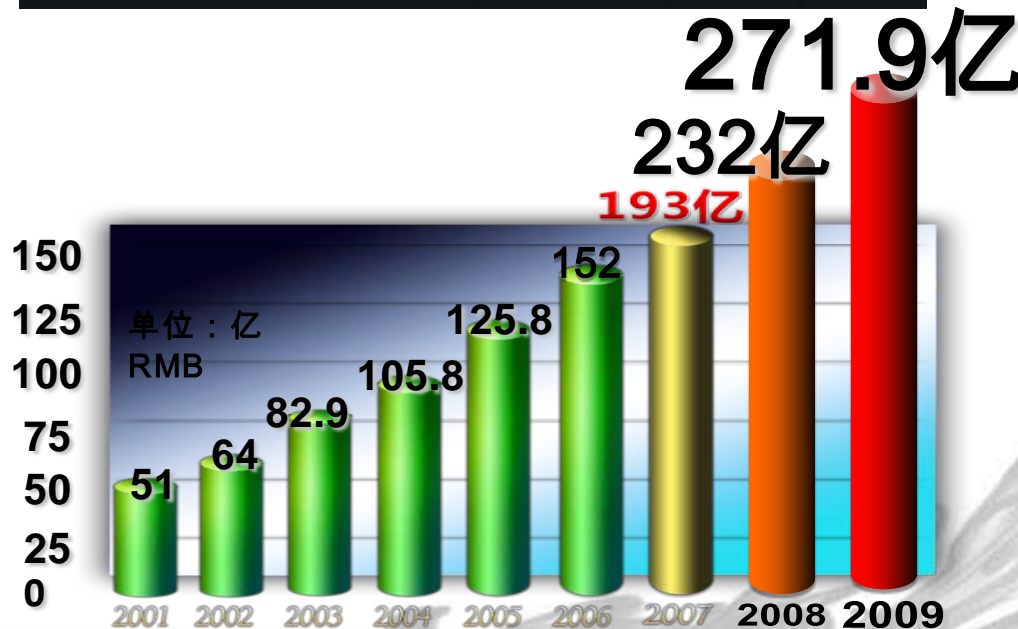
Until 2009

Revenue
27 B RMB

2009

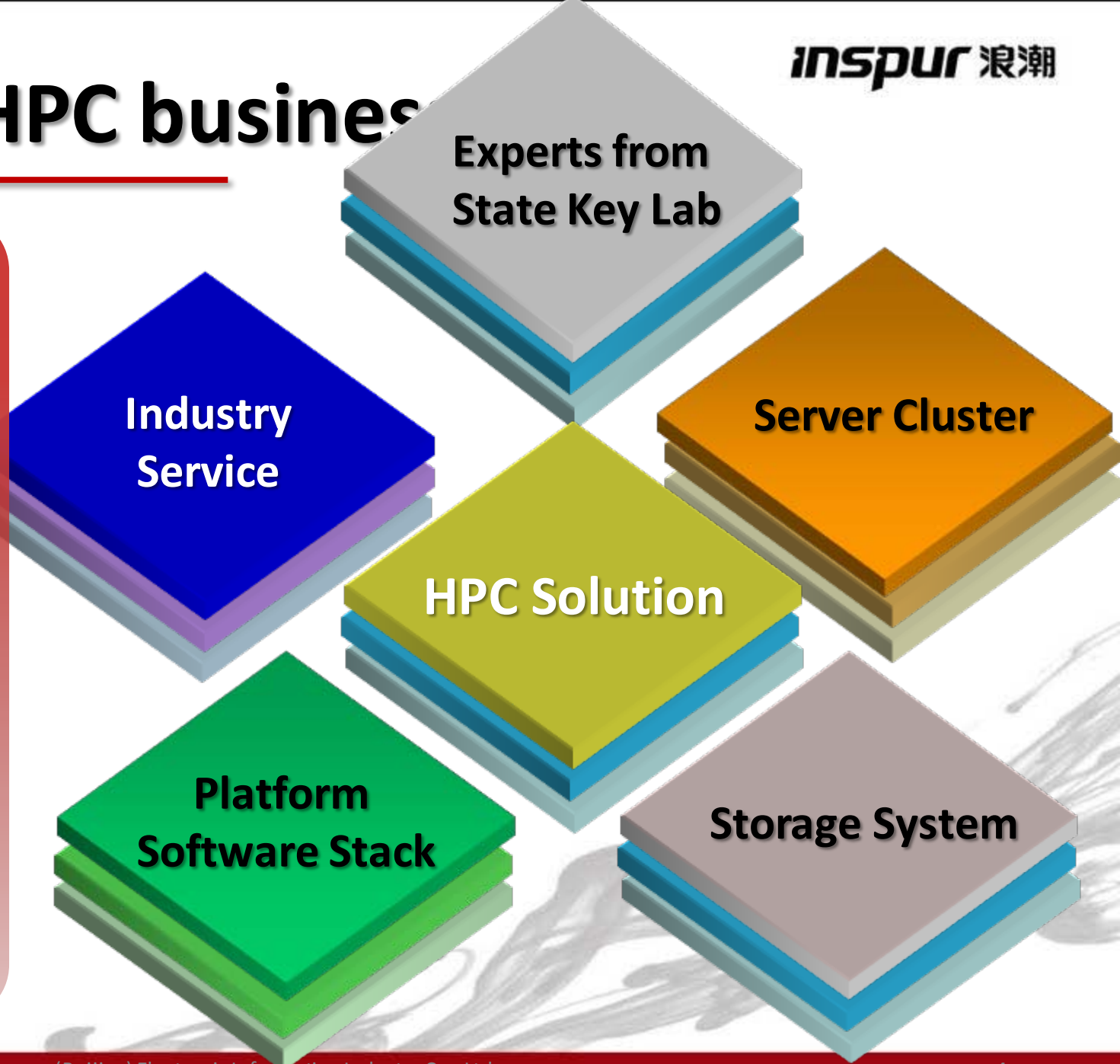
3 public
companies

Until 2009



Inspur HPC business

**Make
HPC
Easy**



Heterogeneous HPC in Inspur



NF5588



NF5588

Not only hardware

- NF5188 (Rack-based), NF5588 (Desktop),

Software guys are also here

- Dedicated software team – Application Promotion Division

Application/Software matters

Challenges in Heterogeneous HPC

Tools

Programming Model

Tool-chain

Platform Software

System Deployment

Monitor

Job Management

Algorithm Reuse

Refine Parallelism Granularity

New Architecture Mapping

Application Design

(New) Algorithm
+ Legacy Software
+ New Hardware

Methodology

Respect to the code base

Phase 1: Focus on hotspot, keep software architecture untouched



Accelerate the code module

Phase 2: Explore the massive parallelism Algorithm porting

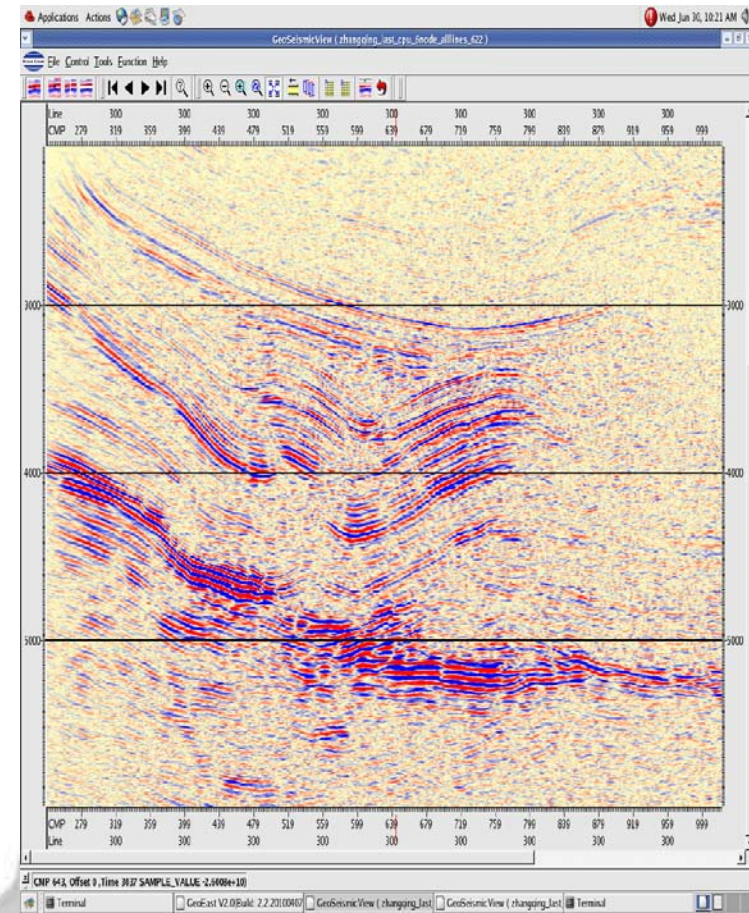


Integration with the application

Phase 3: Coarse (MPI) and fine (CUDA) granularity tuning integration

Case study

- Pre-stack Time Migration (PSTM)
- Key component of Seismic information processing
 - Used in petroleum prospection
- Joint R&D project with BGP and Inspur
 - BGP, 3-rd petroleum prospection company globally
 - Inspur, leading HPC solution vendor in China



View from 10,000-m height

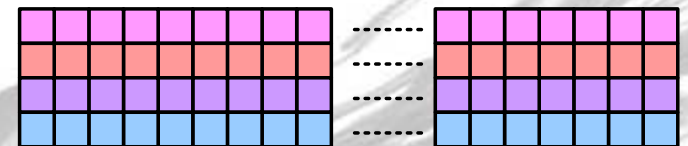
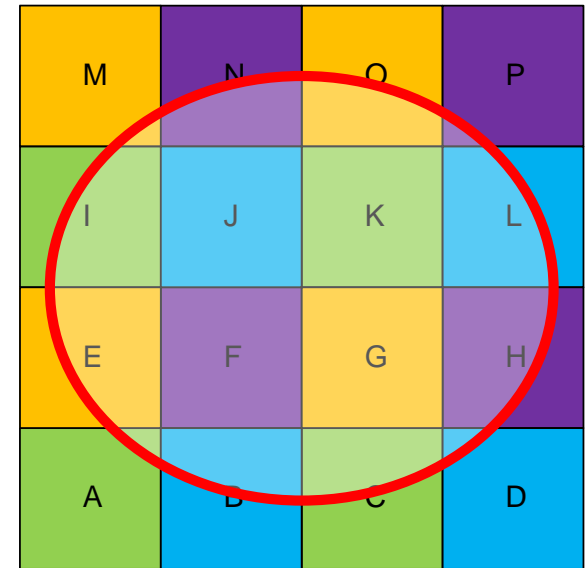
- Valuable case to demonstrate Heterogeneous HPC system
 - MPI-based Cluster-oriented Application
 - Hundreds of nodes are devoted to the PSTM
- Ideal algorithm case for GPU
 - Intrinsic nature of high parallelism algorithm, ideal data dependency
 - Computation-sensitive, good scalability
 - Hotspot is so HOT

Considerations

- CUDA algorithm porting
 - Biggest issue
- Down the overhead
 - PCI-E I/O, side effect introduced by accelerator
- CPU/GPU work together
 - Two level of concurrency
 - Task partition
 - Data partition
 - Handle the issue of different architecture existing together
- MPI tuning
 - Load balance
 - Impact of importing accelerator

View from 10-m height

- Data dispatched among nodes
- pthread within a single node
 - Each thread performs a hotspot function
 - Strategy with DATA Partition
 - Fine granularity parallelism
- Questions:
 - How to active GPU device?
 - Redefine DATA strategy?
 - Eliminate PCI-E I/O?



View from 0.1-m height

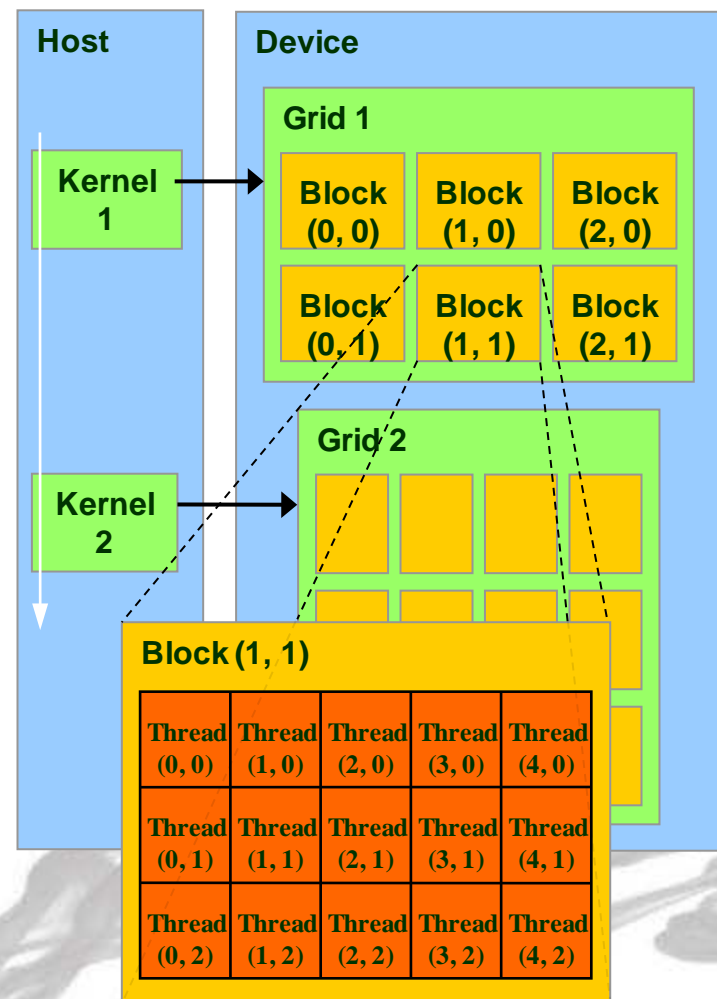
- Existing algorithm is NOT suitable to the GPU micro-architecture
 - Accelerated arithmetic (dynamic interpolation)
- Question:
 - Algorithm refine?
 - Thread model?
 - On-chip load balance?
 - Memory hierarchy?
 - Register pressure?

Design inside-out

- Algorithm
 - Using static interpolation instead of dynamic interpolation
 - Introduce more computation but more accurate results
- Trade computation complexity with computation speed
 - HAPPENED with GPU!

GPU Tuning

- Thread model
 - Try and decide
- On-chip load balance
 - Revisit thread model (give a right block number)
 - Typical case, not general
 - HAPPENED with GT-200, not Fermi

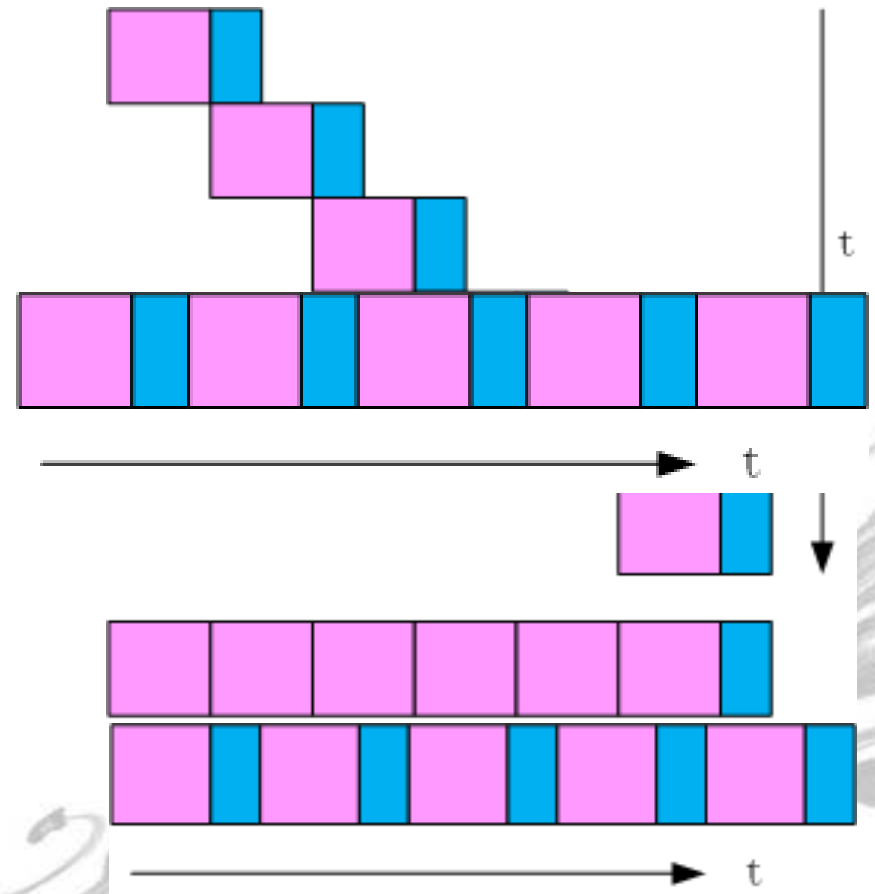


PCI-E traffic overlap

- GPU computation overlap PCI-E traffic

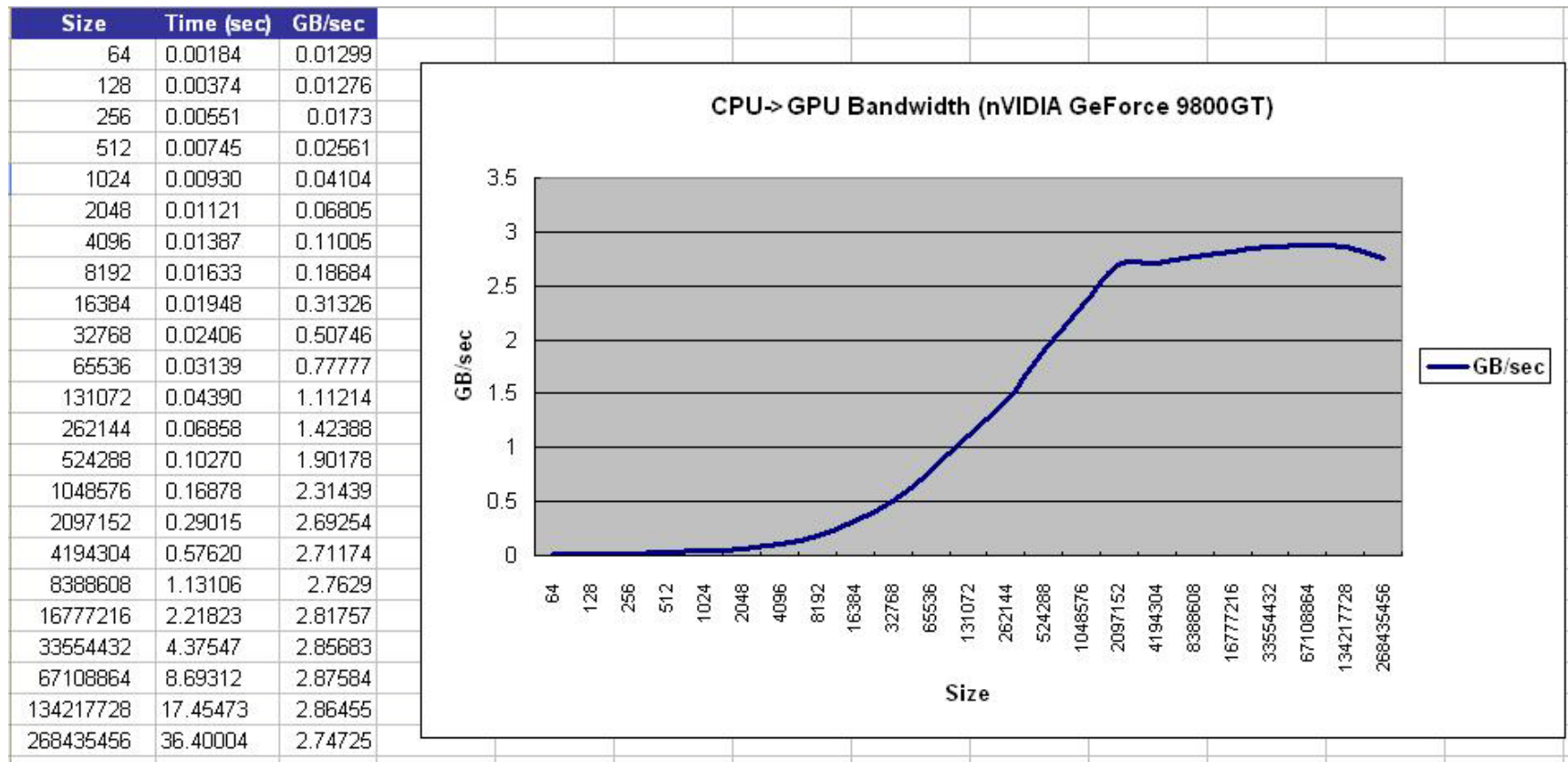
```

for p:1 to N do
  PSTM_GPU(p);
  IO(d) ANSYC(d);
end for
  
```



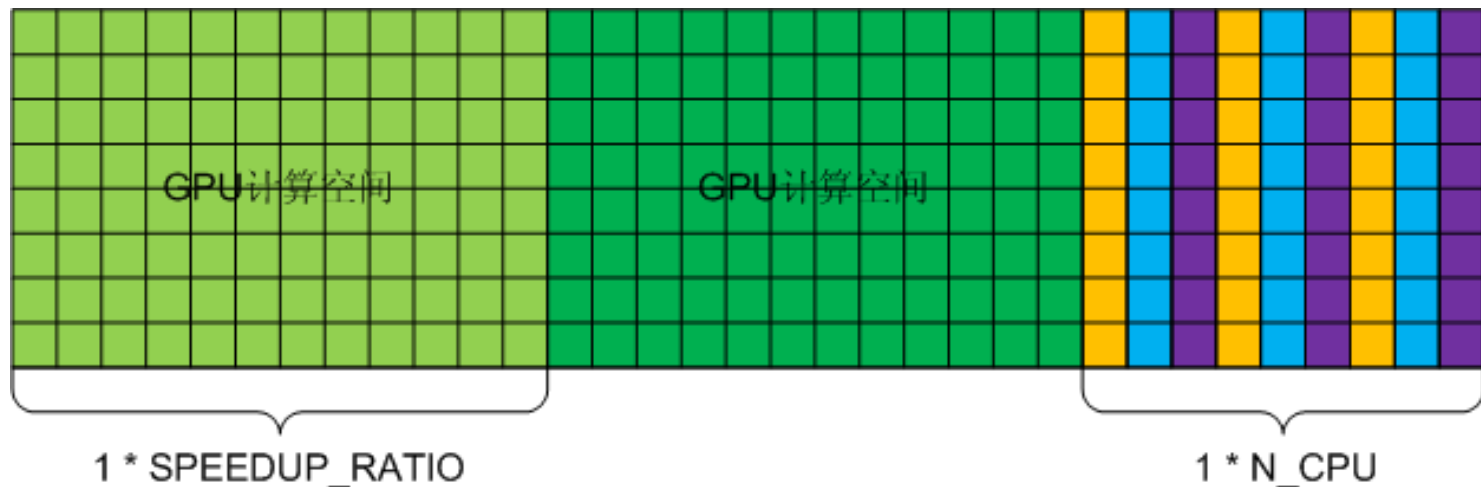
PCI-E efficiency

- Choose the right data amount for PCI-E efficiency



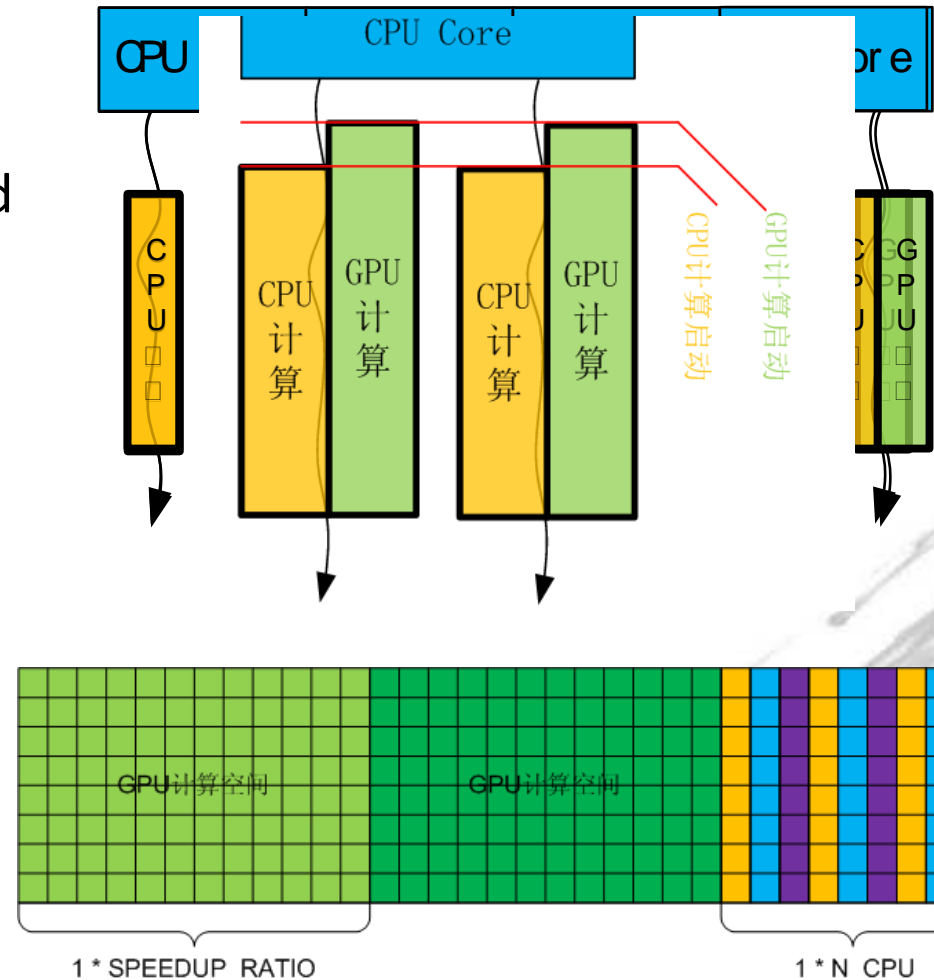
Outter - data partition

- DATA partition strategy
 - More DATA belongs to the GPU
 - Load balance
 - Dynamic partition vs. static partition



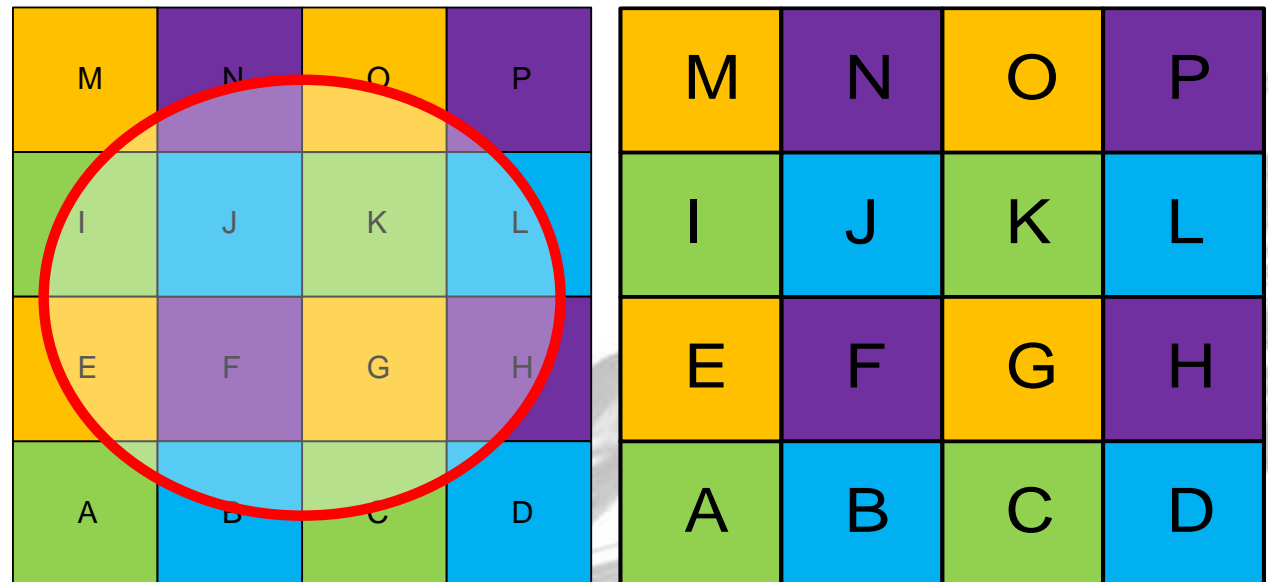
Node-level tuning

- Two level concurrency
 - Inter-thread concurrency
 - CPU thread and GPU thread do the SAME job
 - Natural behavior
 - Intra-thread concurrency
 - CPU and GPU do jobs simultaneously in a single thread
 - Physically separated, offer the opportunity to invoke/perform task simultaneously at different device



Cluster-level tuning

- Profile MPI IO time
- Fine control MPI DATA traffic
 - Overlap between IO time and computation time
- Load balance?



Performance results

- Computing Node: NF5188 (2 socket CPUs with 2 GPUs)
- CPU cores: Westmere E5620 x 2 x 6
- GPU: Tesla C1060 x 2 x 6

258

- Same computational complexity
- GPU vs. CPU thread

40

- Accelerated arithmetic
- GPU vs. CPU thread

~5

- Application speedup
- GPU cluster vs. CPU cluster

Data

Case	6_1022_Line_Test			
Scenario		MIN	MAX	INC
	Line	6	1022	1
	CMP	239	1335	1
	Samples	0	1503	1
	CRP	0	5500	100
	Image Size	1017*1097*1504*55*4Byte=343.79GB		
	Trace DATA	DATA1~5(Pre_data_line6-line1022)= 373.54GB		
Performance	Nodes	NF5188 CPU only	NF5188 CPU/GPU	
	6(Inspur)	938433.6(s)=260.676(h)	180475(s)=50.13(h)	

$$260.373/50.13 = 5.2$$

Other data

- NF5188 Cluster (6 nodes) vs. BGP-DN8 (12 nodes)

Case	Cluster	Cores	Time	Speed-up
401 Lines	BGP-DN8	8 * 12 = 96 cores	130h	130 / 19.64 = 6.62
	NF5188 Cluster	8 * 6 CPU cores + 2 * 6 Tesla C1060	19.64h	

1 NF5188 node = 13.3 BGP-DN8 node

- Application run well at BGP-HP-Siglo (E5462 + M1060)

Summary

- Efforts
 - 12 man-month
 - Design, documentation, coding, testing, performance tuning, system deployment
- Systemic turning
 - Core algorithm is the key factor
 - CPU code and systemic tuning is important
- More industry applications are desired
 - The more available software package, the more cost performance of heterogeneous HPC system



inspur 浪潮

谢谢!