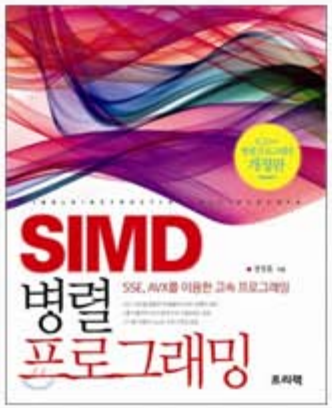
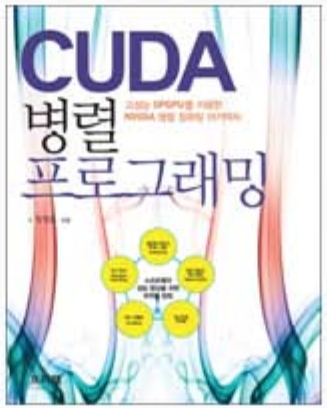
병렬 컴퓨팅과 Xeon Phi

정영훈

강사 소개

- (주)디아이티-Digital Image Tech.
  - 중앙연구소 차장
  - 전: MGame, NHN Games
- 연세대학원 컴퓨터공학 석사
- 저서 : SIMD 병렬프로그래밍
  OpenMP 병렬프로그래밍
  CUDA 병렬프로그래밍
  Kinect 프로그래밍



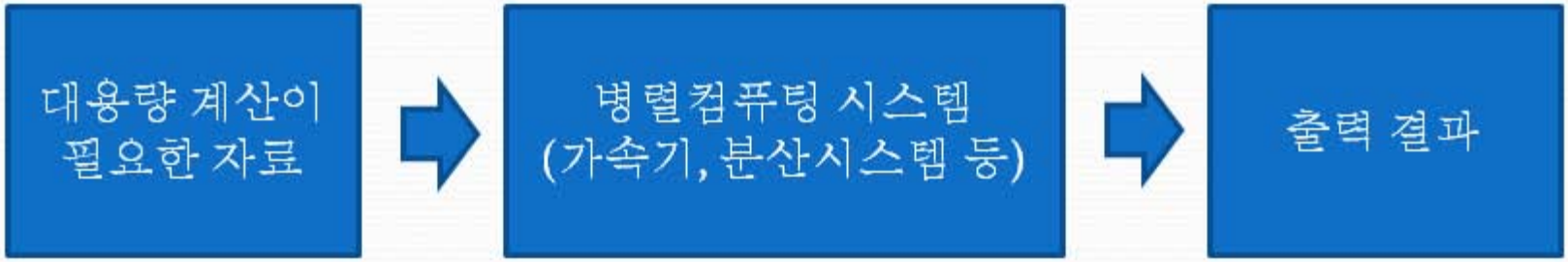


목차

1. 병렬컴퓨팅 시스템
2. 병렬컴퓨팅 분류 및 특징
3. Xeon Phi 개요
4. Xeon Phi vs CUDA
5. Xeon Phi 장점
6. 개선이 필요한 사항

1.병렬컴퓨팅 시스템

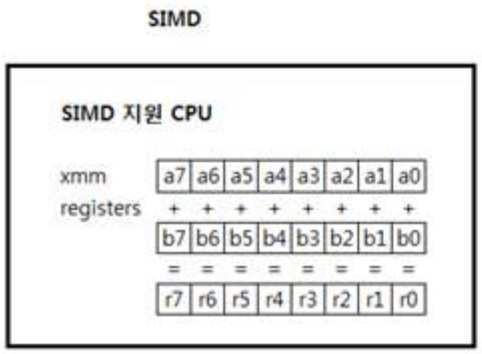
- 병렬 컴퓨팅 시스템

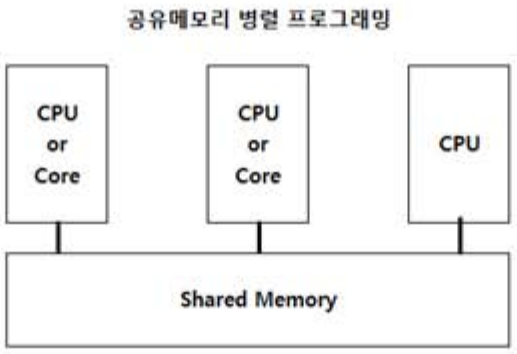


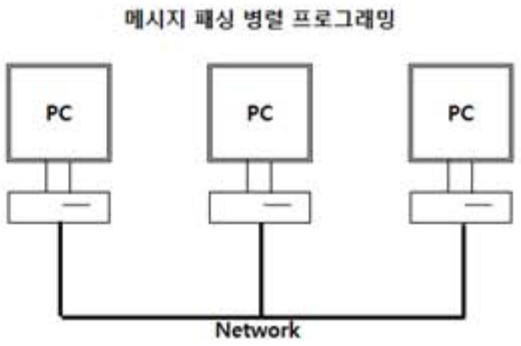
  - 대용량 연산이 필요한 자료를 가속기 또는 병렬, 분산 시스템을 통해 계산 후 결과를 돌려줌

2-1.병렬 시스템 분류

- CPU
  - 명령어 수준의 가속 : SIMD
  - 멀티 코어 : OpenMP, Cilkplus, TBB
    , pThread, CreatThread
- 분산 시스템
  - MPI





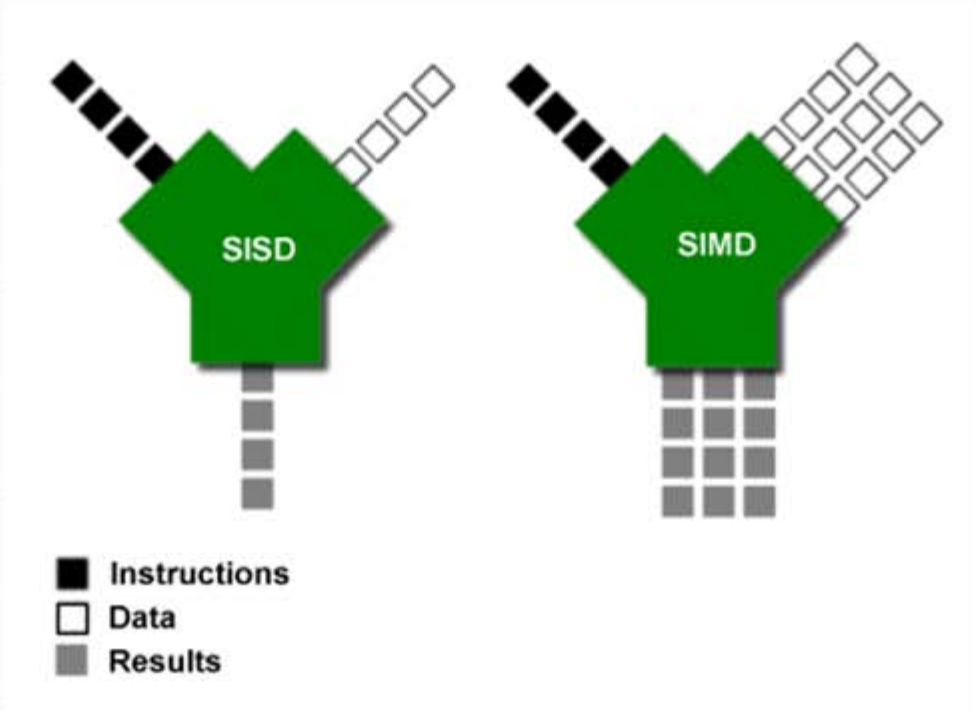


2-1.병렬시스템 분류

- GPGPU
  - CUDA, OpenCL (CPU, GPU)

2-2.SIMD

- 하나의 명령어로 여러개의 데이터를 연산

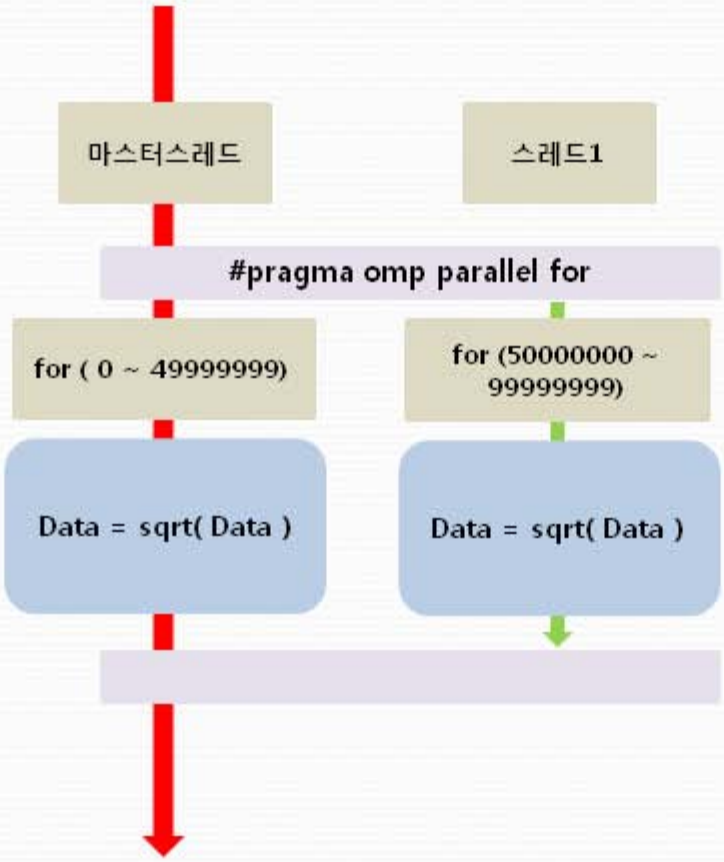


2-2.SIMD intrinsic code

```
int main(int argc, char* argv[])
{
    __declspec(align(16)) short A[8] = {2,4,6,8,10,8,6,4};
    __declspec(align(16)) short B[8] = {1,2,3,4,5,6,7,8};
    __declspec(align(16)) short R[8] = {0};

    __m128i xmmA = _mm_load_si128((__m128i*)A);
    __m128i xmmB = _mm_load_si128((__m128i*)B);

    __m128i xmmR = _mm_add_epi16(xmmA,xmmB);          //덧셈 연산

    _mm_store_si128((__m128i*)R,xmmR);                //결과 출력
    printf("Add : %d, %d, %d, %d, %d, %d, %d, %d\n"
           ,R[7],R[6],R[5],R[4],R[3],R[2],R[1],R[0]);
}
```

2-2.OpenMP Fork-join 구조

- 공유메모리 병렬프로그래밍 표준



2-2.OpenMP #Pragma 지시어

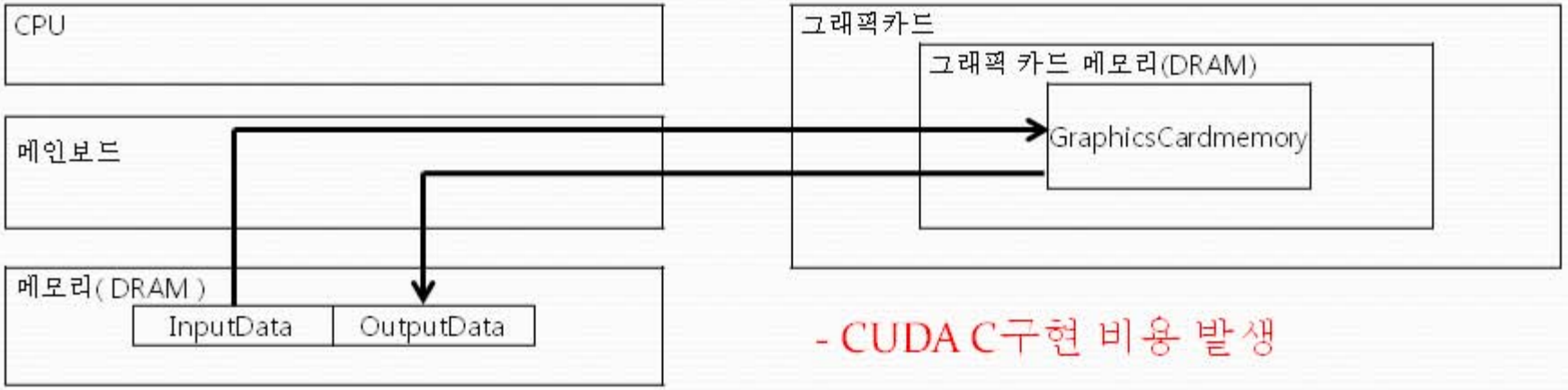
```
int _tmain(int argc, _TCHAR* argv[])
{
    const int MAX = 100000000;
    float* Data;
    Data = new float[MAX];

    int i = 0;
    for(i = 0; i< MAX; i++)
            Data[i] = i;

#pragma omp parallel  for
    for(i = 0; i< MAX; i++)
            Data[i] = sqrt(Data[i]);

    printf( "Data : %f, %f, %f, %f, %f\n",
            Data[0], Data[1], Data[2], Data[3], Data[4]);

    delete Data;
    return 0;
}
```

2-2.GPGPU

- Tesla kepler GK110
- 스레드의 끝판왕?
- 192x14 = 2688core

2-2.CUDA 아킬레스건

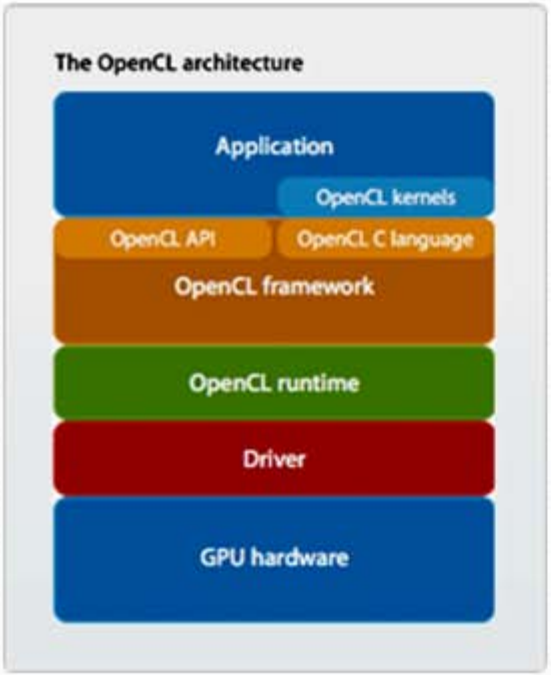
- Host , Device 구조

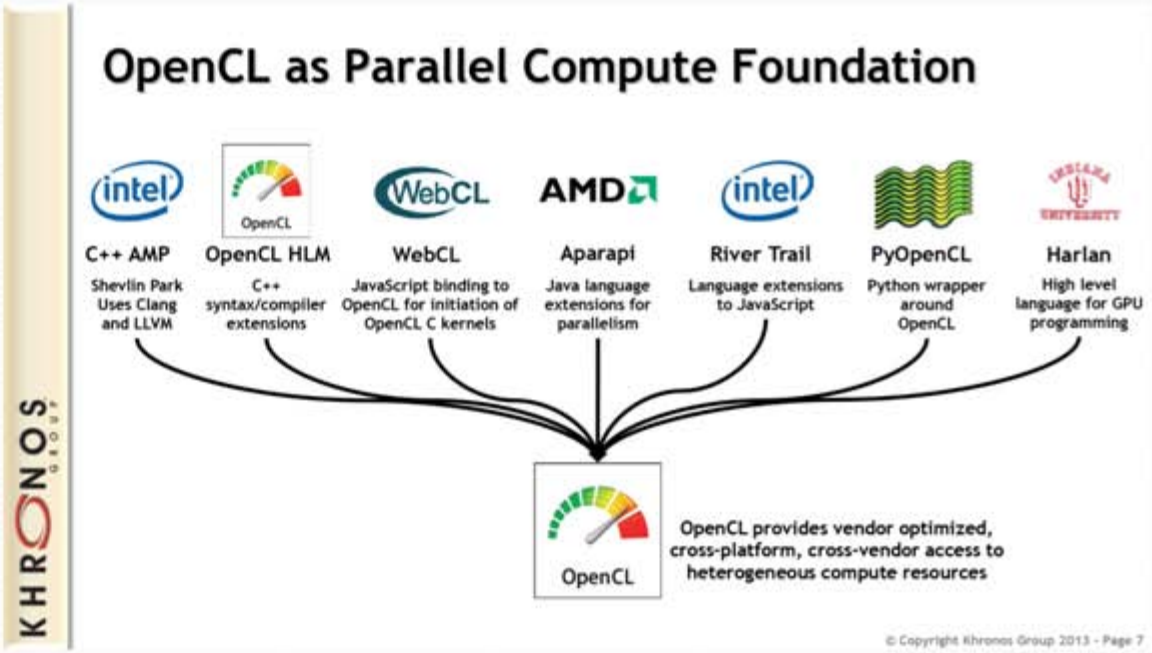


- CUDA C구현 비용 발생

2-2.OpenCL

- 이기종 병렬처리 표준
  - SIMD + OpenMP + CUDA 지원

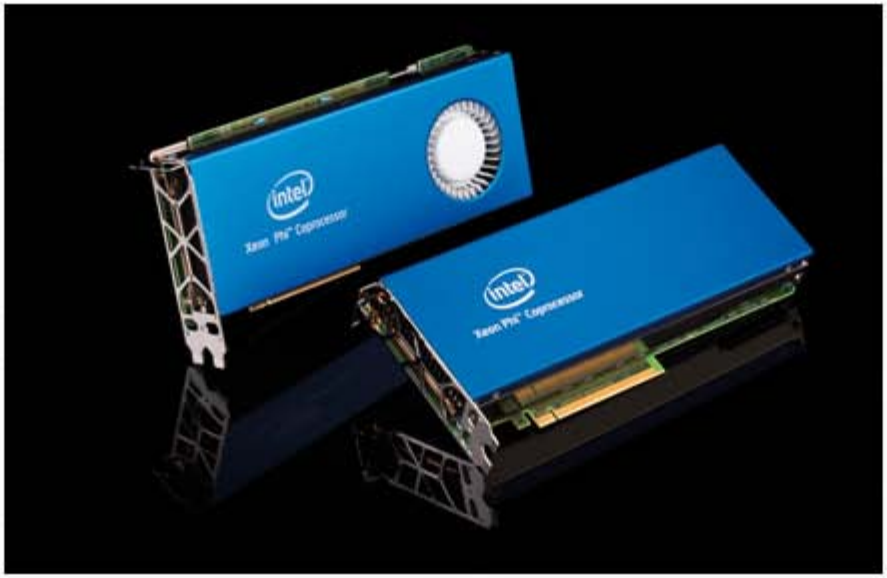




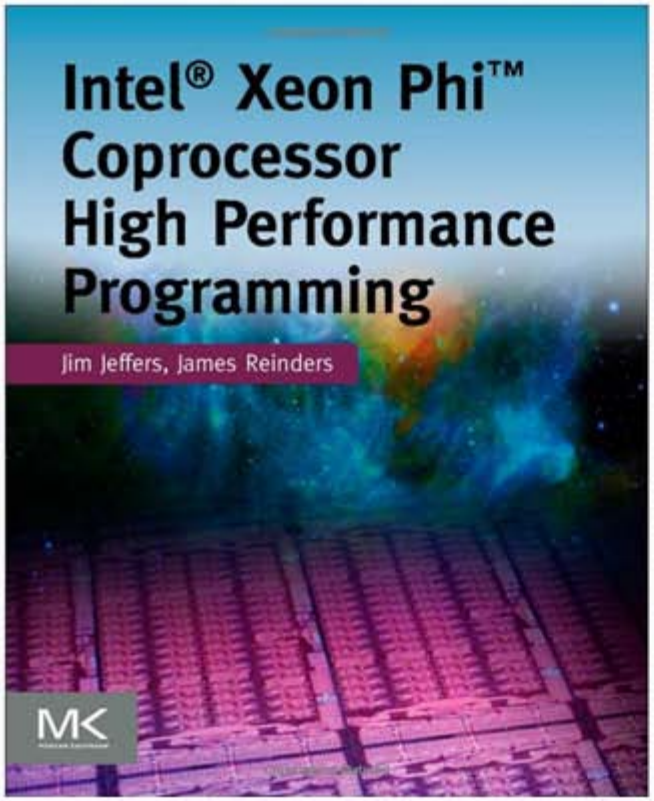
http://www.mat.ucsb.edu/a.forbes/teaching/mat594cm_S2010/benalunjones-rpi/sl_opencl_architecture.png

3.Xeon Phi

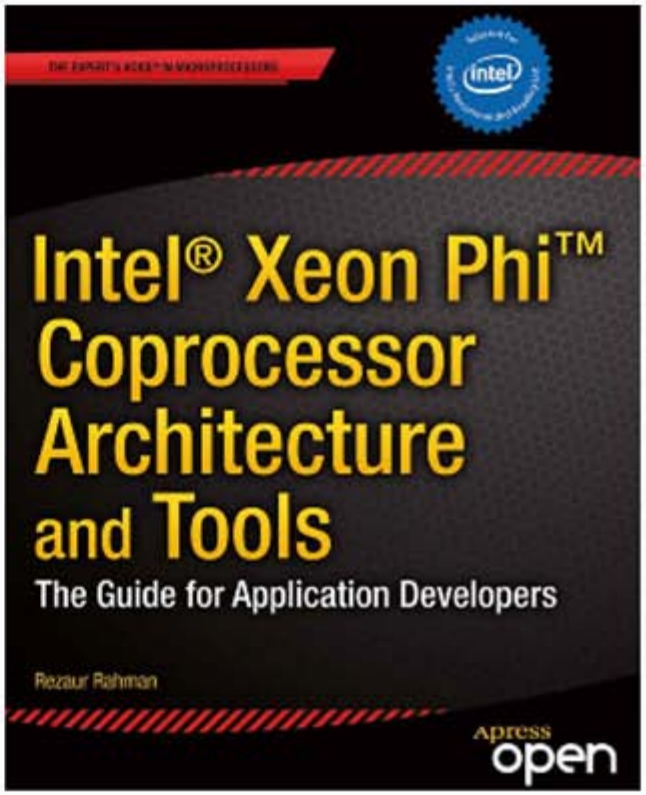
- Xeon + Phi
- Phi? - 개선된, 향상된.. +α
- coprocessor



3.참고자료

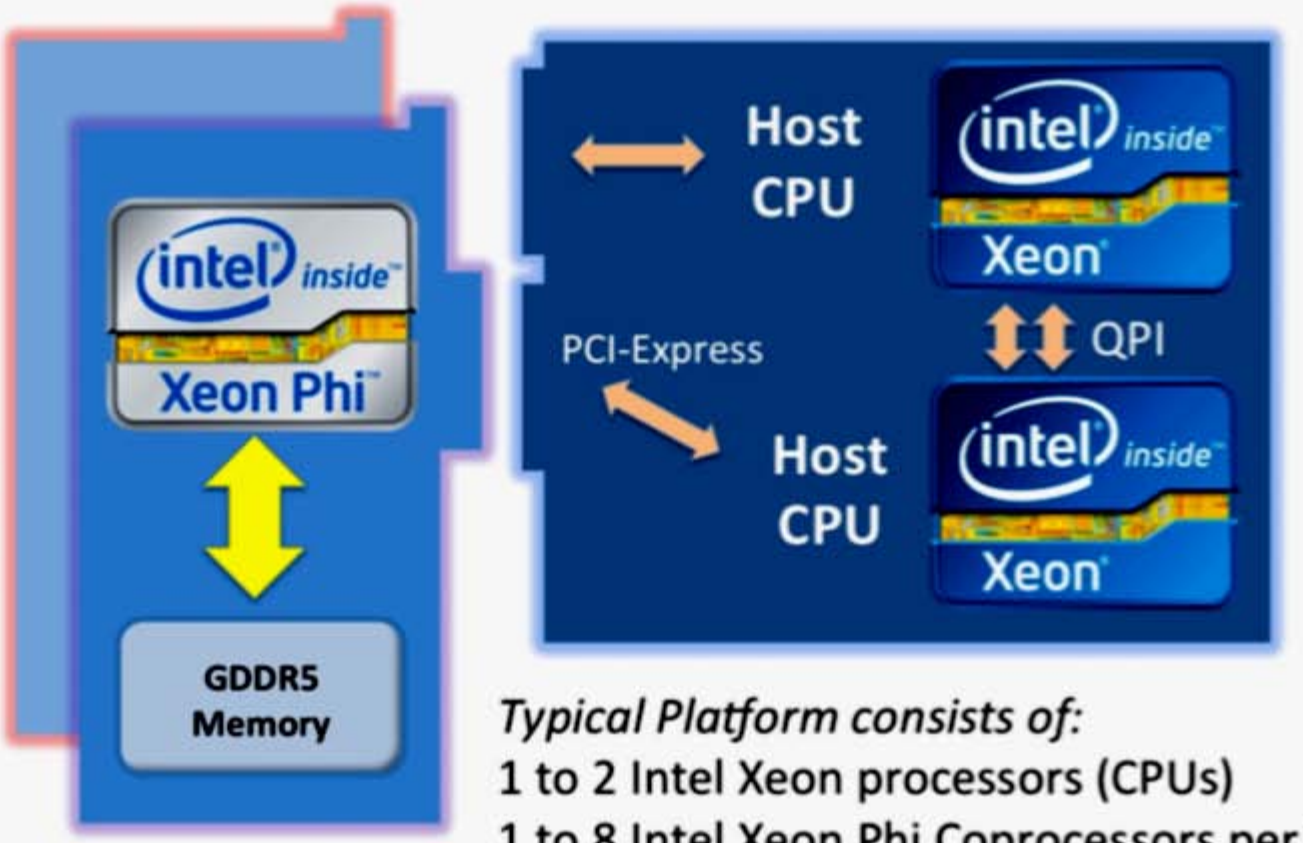


$60



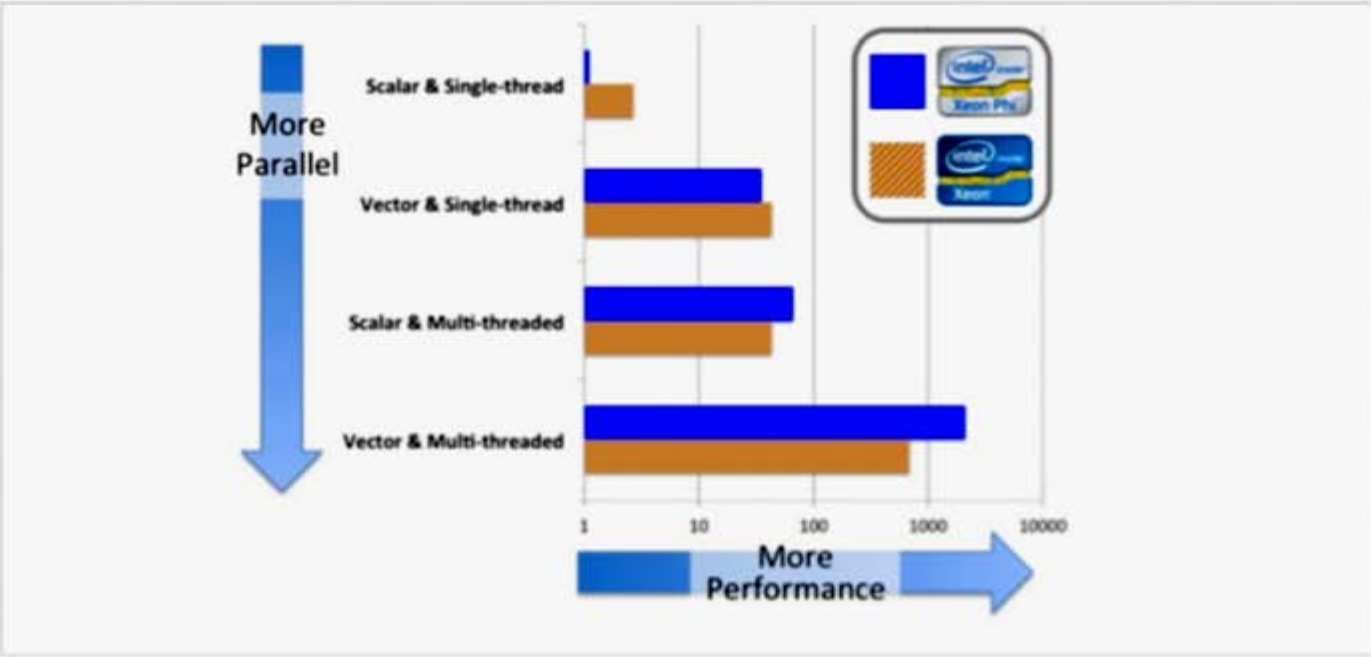
-ebook $0.12

3.Xeon Phi 구성도



Typical Platform consists of:
1 to 2 Intel Xeon processors (CPUs)
1 to 8 Intel Xeon Phi Coprocessors per host

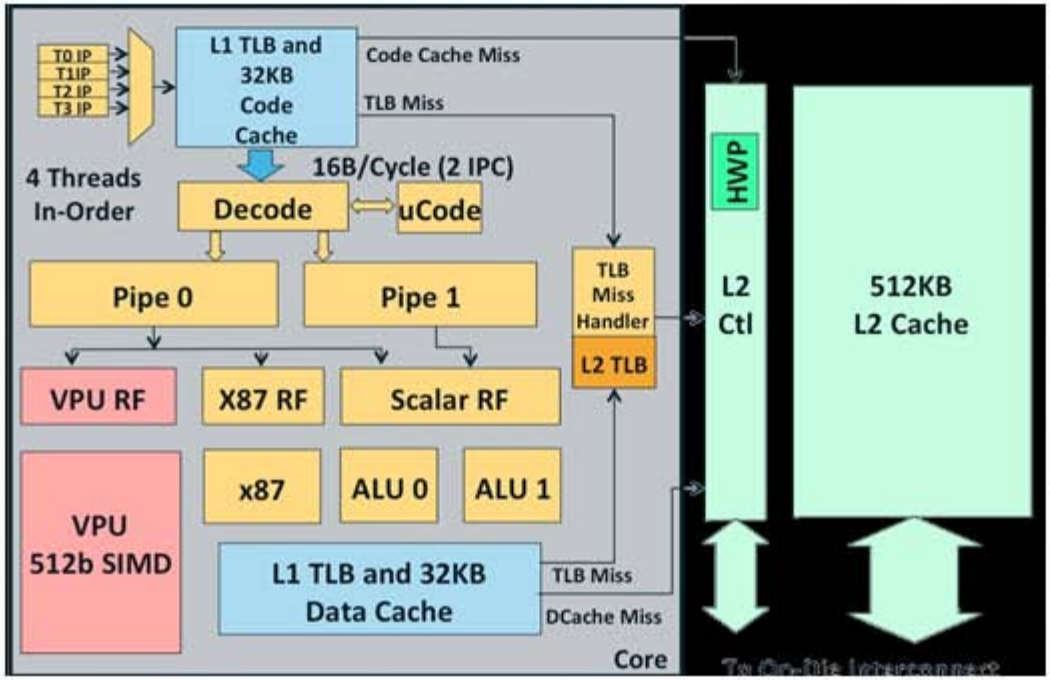
3.Xeon Phi 성능 향상



- SIMD + OpenMP가 중요
- auto vectorization

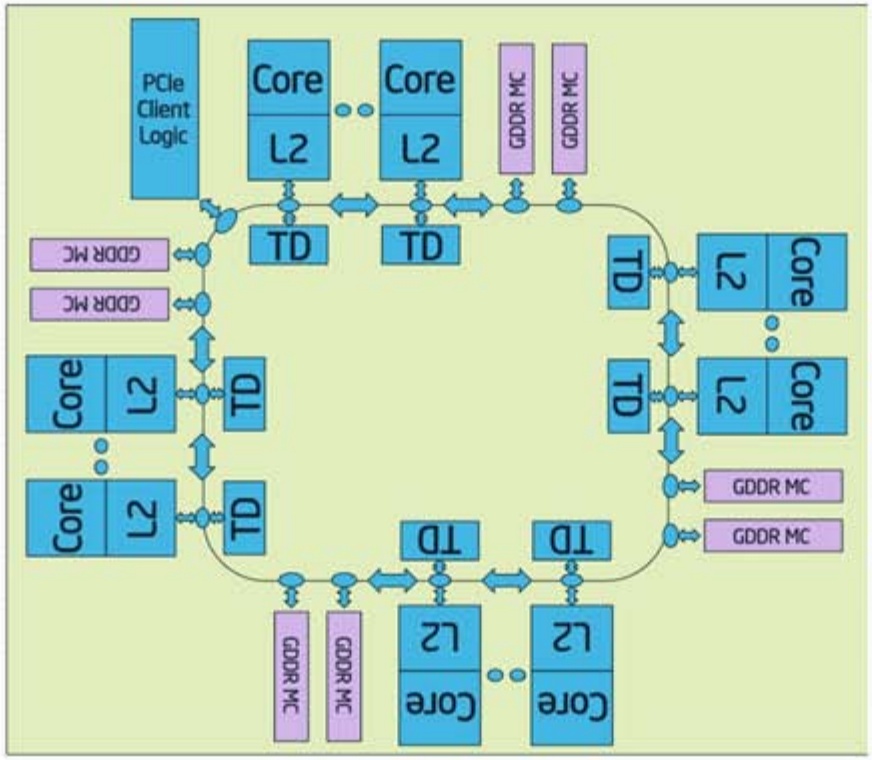
3.Xeon Phi – 1 Core

- The 4 threads are mostly a way to hide memory latency
- In the best case, two threads will execute in parallel.



* http://www.anandtech.com/show/6451/the-xeon-phi-at-work-at-tacc

3.On Chip Memory



- Eight memory channels (512-bit interface) support up to 8 GB of RAM, and PCIe logic is on chip

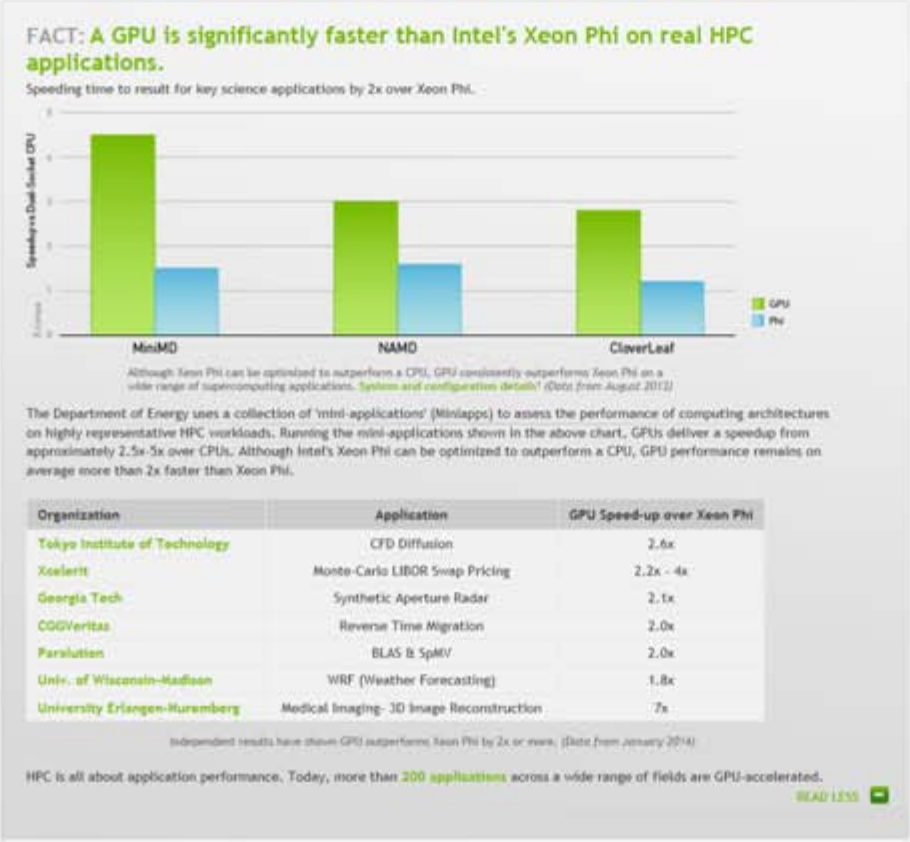
4.Xeon Phi vs Nvidia Tesla

구분	Xeon E5-2670	Xeon Phi 5110P	Tesla K20X
Cores	8	60	14 SMX
Logical Cores	16 (HT)	240 (HT)	2,688 CUDA cores
Frequency	2.60GHz	1.053GHz	735MHz
GFLOPs (double)	333	1,010	1,317
SIMD width	256 Bits	512 Bits	N/A
Memory	~16-128GB	8GB	6GB
Memory B/W	51.2GB/s	320GB/s	250GB/s
Threading	software	software	hardware

* http://blog.xcelerit.com/intel-xeon-phi-vs-nvidia-tesla-gpu/

4.Nvidia Tesla

- Xeon Phi 와 2배~7배 까지 성능차를 주장



FACT: **A GPU is significantly faster than Intel's Xeon Phi on real HPC applications.**

Speeding time to result for key science applications by 2x over Xeon Phi.

Although Xeon Phi can be optimized to outperform a CPU, GPU consistently outperforms Xeon Phi on a wide range of supercomputing applications. System and configuration details? (Data from August 2013)

The Department of Energy uses a collection of 'mini-applications' (Miniapps) to assess the performance of computing architectures on highly representative HPC workloads. Running the mini-applications shown in the above chart, GPUs deliver a speedup from approximately 2.5x-5x over CPUs. Although Intel's Xeon Phi can be optimized to outperform a CPU, GPU performance remains on average more than 2x faster than Xeon Phi.

Organization	Application	GPU Speed-up over Xeon Phi
Tokyo Institute of Technology	CFD Diffusion	2.6x
Xcelerit	Monte-Carlo LIBOR Swap Pricing	2.2x - 4x
Georgia Tech	Synthetic Aperture Radar	2.1x
CGGVeritas	Reverse Time Migration	2.0x
Parallution	BLAS & SpMV	2.0x
Univ. of Wisconsin-Madison	WRF (Weather Forecasting)	1.8x
University Erlangen-Nuremberg	Medical Imaging- 3D Image Reconstruction	7x

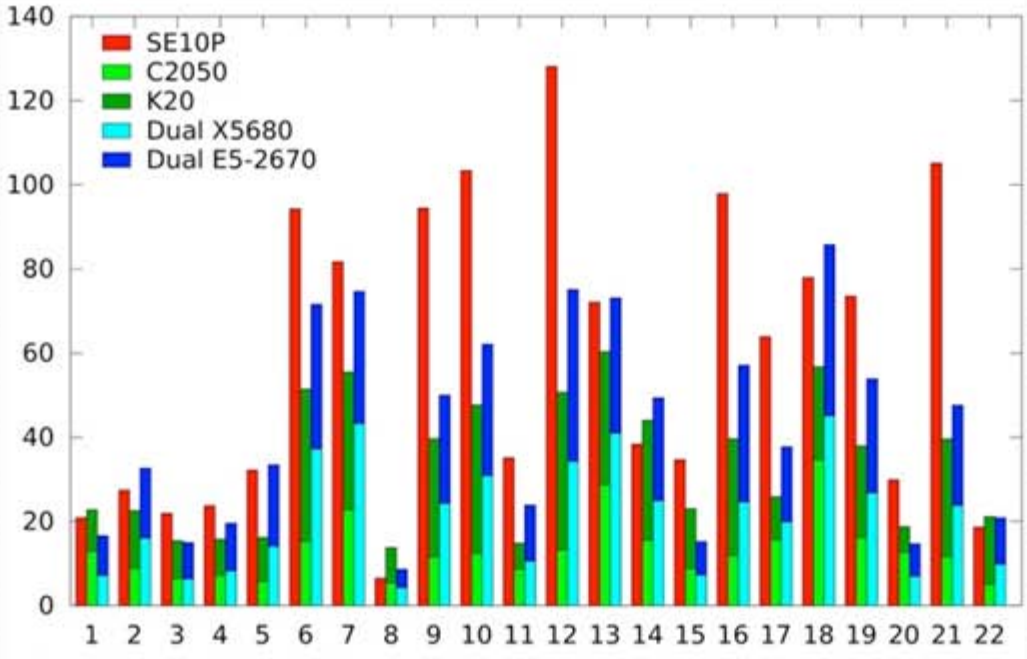
Independent results have shown GPU outperforms Xeon Phi by 2x or more. (Data from January 2014)

HPC is all about application performance. Today, more than 200 applications across a wide range of fields are GPU-accelerated.

READ LESS

4.intel : Ohio State University

- Tesla보다 2배 차이 Up
- Dual Xeon 보다 Tesla 성능이 떨어짐



"Solver" application

5.Xeon Phi 장점 (CUDA대비)

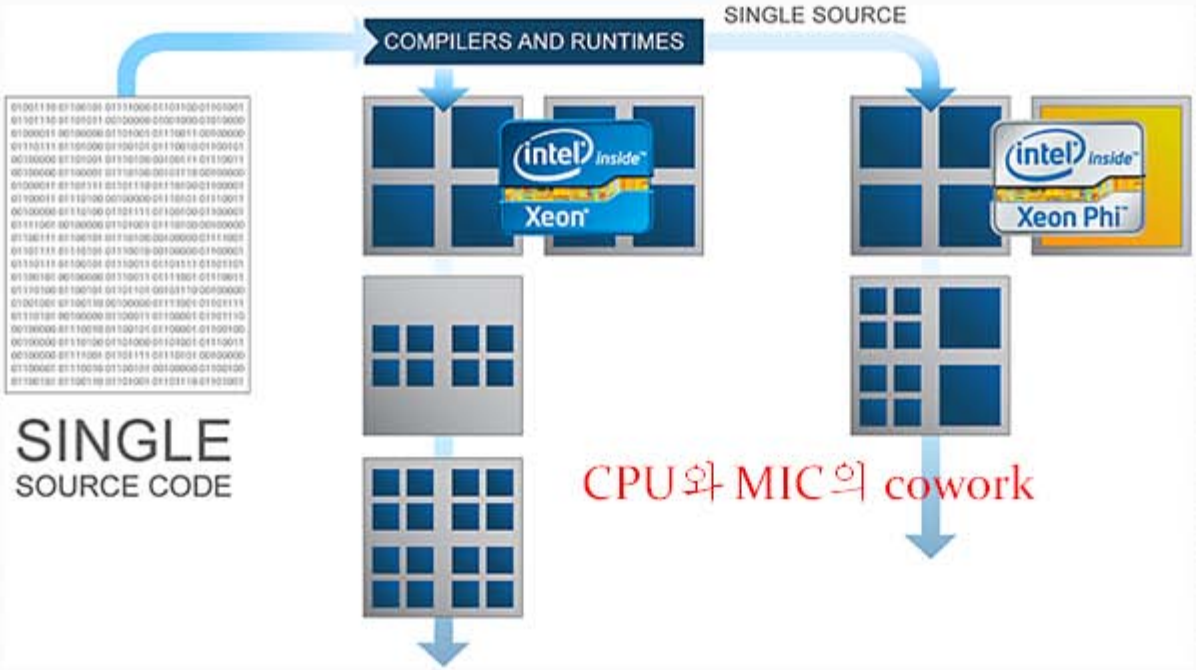
1. Ninja Gap의 축소
2. 변환과 튜닝의 두마리 토끼
3. processor 성능의 최적화
4. auto scaling

5-1.Ninja Gap의 축소

- Ninja Gap이란?
  - 프로그래머와 병렬컴퓨팅 전문가간의 Gap
- 동일한 architech
  - x86

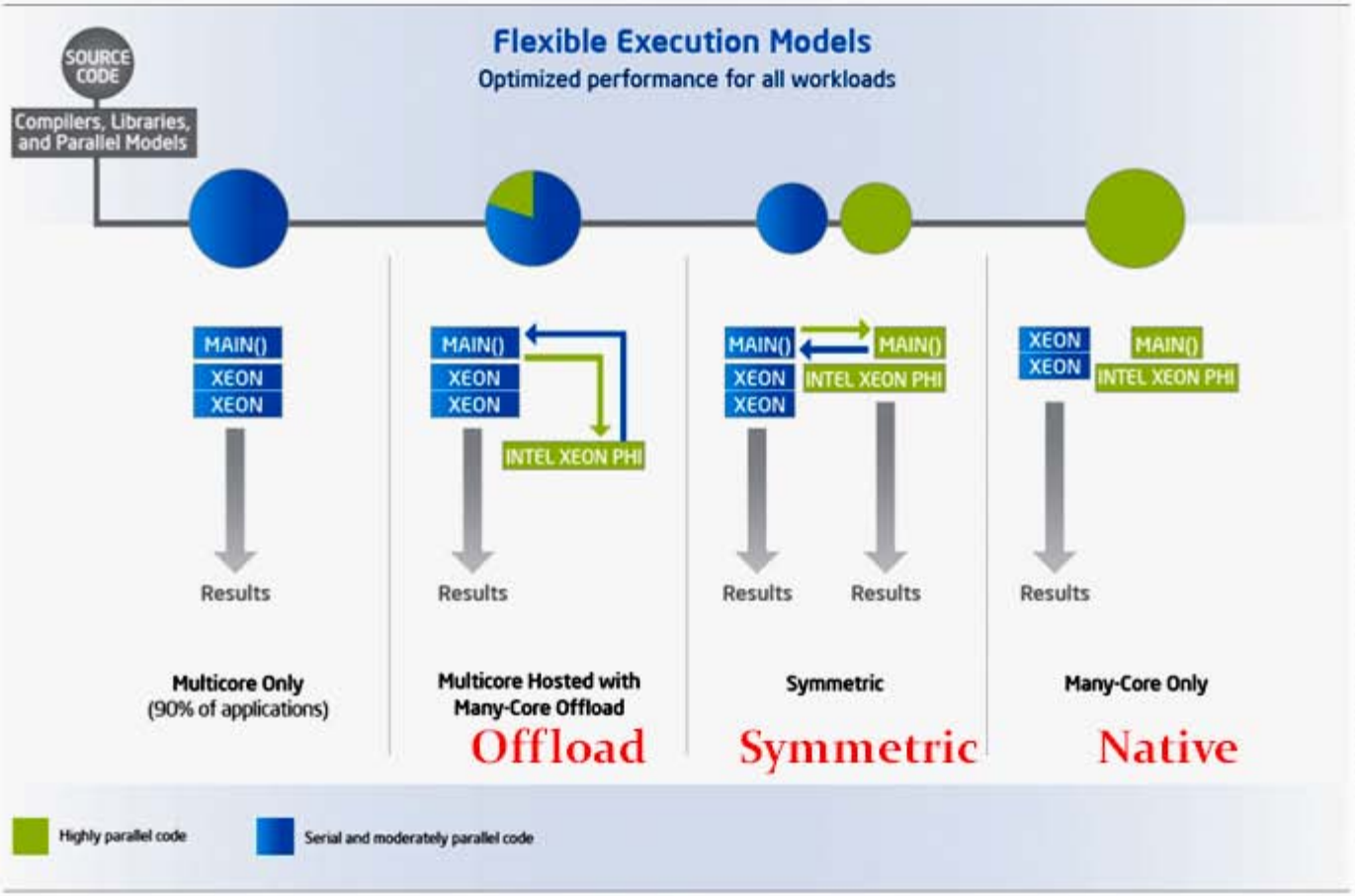
Ninja Gap

5-2.변환과 튜닝의 두마리 토끼



-CUDA C를 배울 필요가 없음
- OpenMP의 활용

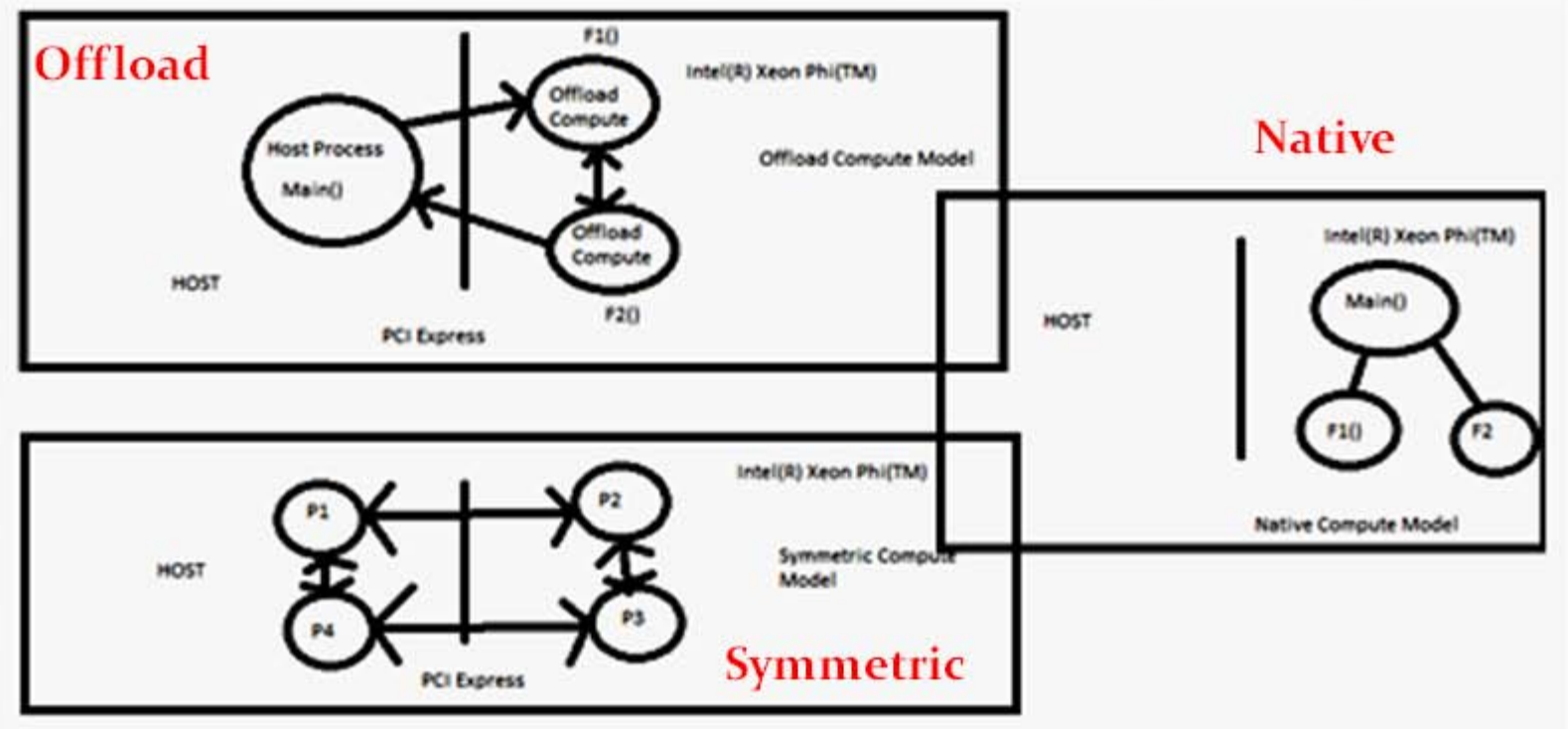
5-3.Multi mode



*http://www.servethehome.com/wp-content/uploads/2012/11/Intel-Xeon-Phi-Execution-Models-1024x675.png

5-3.Multi mode

- 목적 : Processor로 최대한 처리한 후 남은 것을 MIC로 보낸다.



* http://software.intel.com/sites/default/files/article/393203/12394-figure-1.png

5-4.auto scaling

vs CUDA

6.Xeon Phi 개선 할 점

1. 성능
   - CUDA를 넘은 속도가 필요
2. 캐시의 활용
   - L1, L2 캐시의 개방
3. Windows 지원
   - Windows Driver, 가상 OS 구조
4. Intel 컴파일러의 사용
   - Visual studio, GCC

감사합니다.