

CUDA를 활용한 옵션 가치 평가에 대한 초병렬 몬테카를로 시뮬레이션

유 현 곤 (박사과정)
연세대학교 수학과
yhgon@yonsei.ac.kr

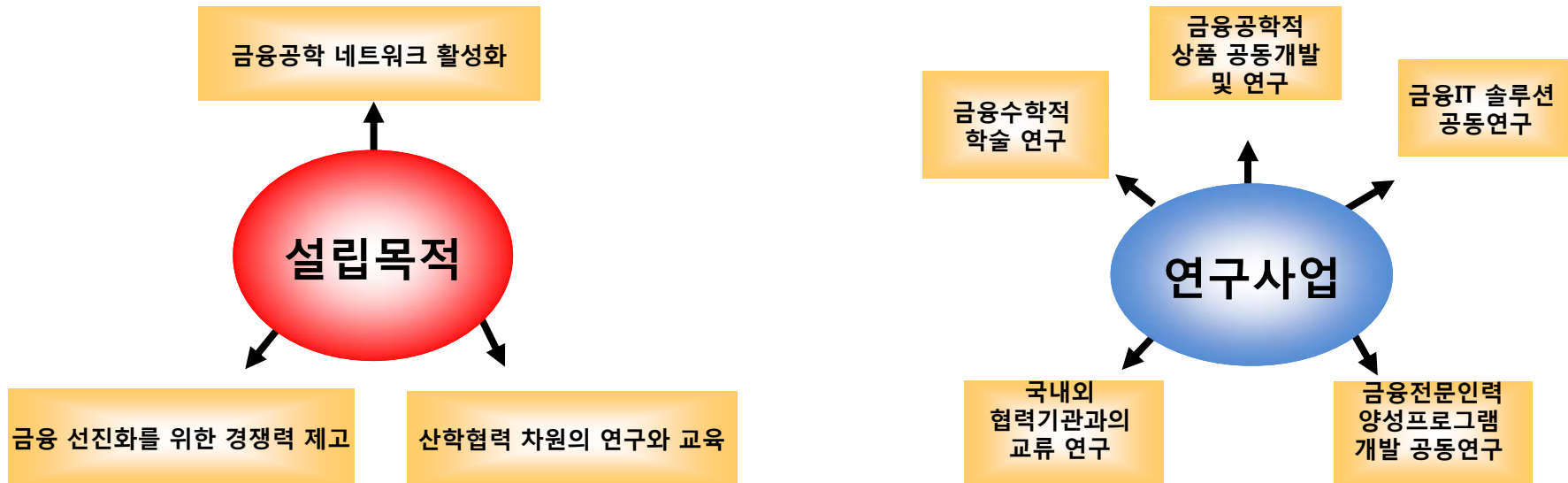
차 례

- 연세 금융퀀트 연구센터 소개
- 연구분야 (옵션 가치 평가)
- 병렬화 (병렬 몬테카를로 시뮬레이션)
- CUDA 적용 사례
- 결과 및 가능성
- 향후 전망

연세대학교 금융퀀트연구센터

<http://quant.yonsei.ac.kr/>

교수님 3분, 박사과정 : 4명, 석사과정 : 12명, 학부생 : 15명



해외 석학 초청 세미나
산학협력 세미나
병렬 컴퓨팅 자문
산학협력 공동 연구 프로젝트 등

연세 금융퀀트 연구센터 CUDA 연구 현황

수퍼컴퓨터를 이용한 병렬처리 (Monte Carlo Simulation) - 2006년부터

Clearspeed를 이용한 병렬처리 (Monte Carlo Simulation) - 2007년부터

CUDA를 이용한 병렬처리 (Monte Carlo Simulation) - 2008년 초부터

세미나 개최 - 2008년 6월 12일, 7월 4일, 7월 11일

연구 결과 발표 - KMS 확률론 워크숍 6월 13일(서울), 16th ICFIDCAA 7월 30일(경주, 예정)

CUDA 등 병렬처리 관련 대학 연구팀, 연구기관 및 업체들과
산학연 협력체제 및 네트워크 구축 중

병렬컴퓨팅 (CUDA) 연구: 국내 대학 금융관련 학과 중 유일함

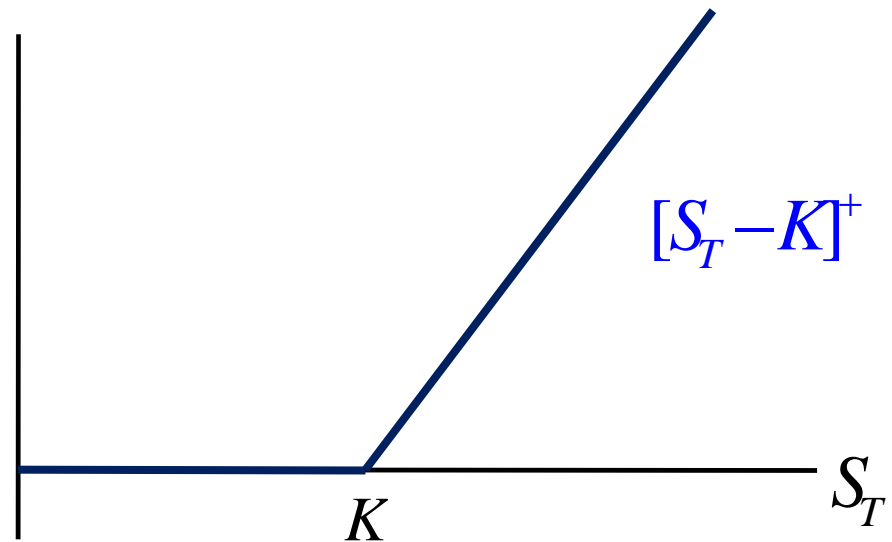
연구 분야 소개 (옵션가치 평가)



ELS 가치 평가, 헷징, 위험관리

옵션 가치 평가 모델을 이용

Payoff



$$C_t = E[e^{-r\tau} [S_T - K]^+]$$

옵션의 현재가치

$$C_t = E[e^{-r\tau} [S_T - K]^+]$$

직접 풀이 Closed form solution

$$C_t = S_0 \Phi(d_1) - Ke^{-r\tau} \Phi(d_2)$$
$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-z^2/2} dz \quad d_1 = \frac{\ln(S_0/K) + (r - \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \quad d_2 = d_1 - \sigma\sqrt{\tau}$$

PDE방법 in FDM, FEM, Meshless

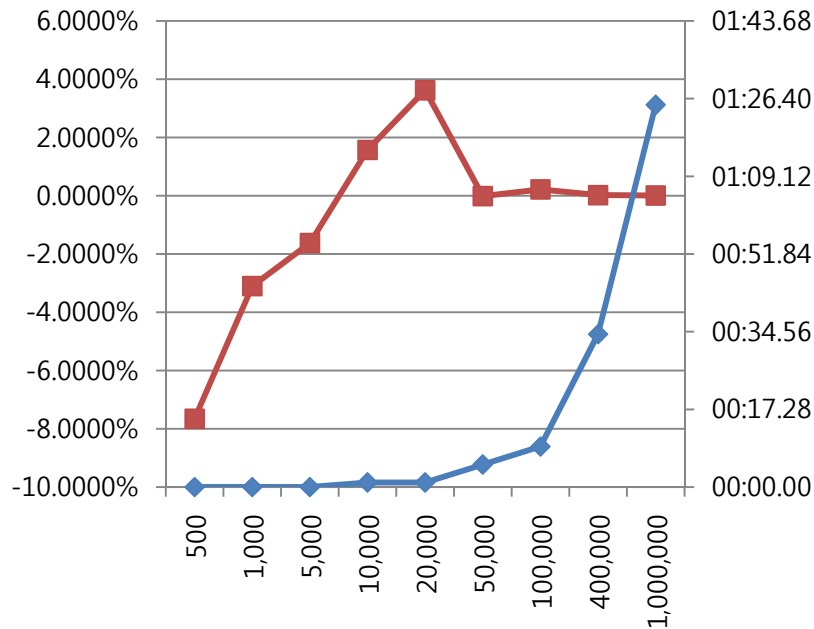
$$\frac{\partial f}{\partial t} + rx \frac{\partial f}{\partial x} + ry \frac{\partial f}{\partial y} + \frac{1}{2} \sigma_x^2 \frac{\partial^2 f}{\partial x^2} + \rho \sigma_x \sigma_y \frac{\partial^2 f}{\partial x \partial y} + \frac{1}{2} \sigma_y^2 \frac{\partial^2 f}{\partial y^2} = rf$$

시뮬레이션 방법 Monte Carlo

$$E[e^{-r\tau} [S_T - K]^+] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_i^N e^{-r\tau} [S_{T_i} - K]^+$$

Monte Carlo Simulation

$$E[e^{-r\tau} [S_T - K]^+] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_i^N e^{-r\tau} [S_{T_i} - K]^+$$



N : 10만번 이상 시행해 줘야함

실제 거래되는 ELS의 계산시
10만 번 실행에 1분 정도 계산시간 소요

100개의 상품이라면 2 시간 이상 소요

현재 국내업계의 구현방법

1. 5만번 정도만 돌림
2. FDM 등의 방법 이용

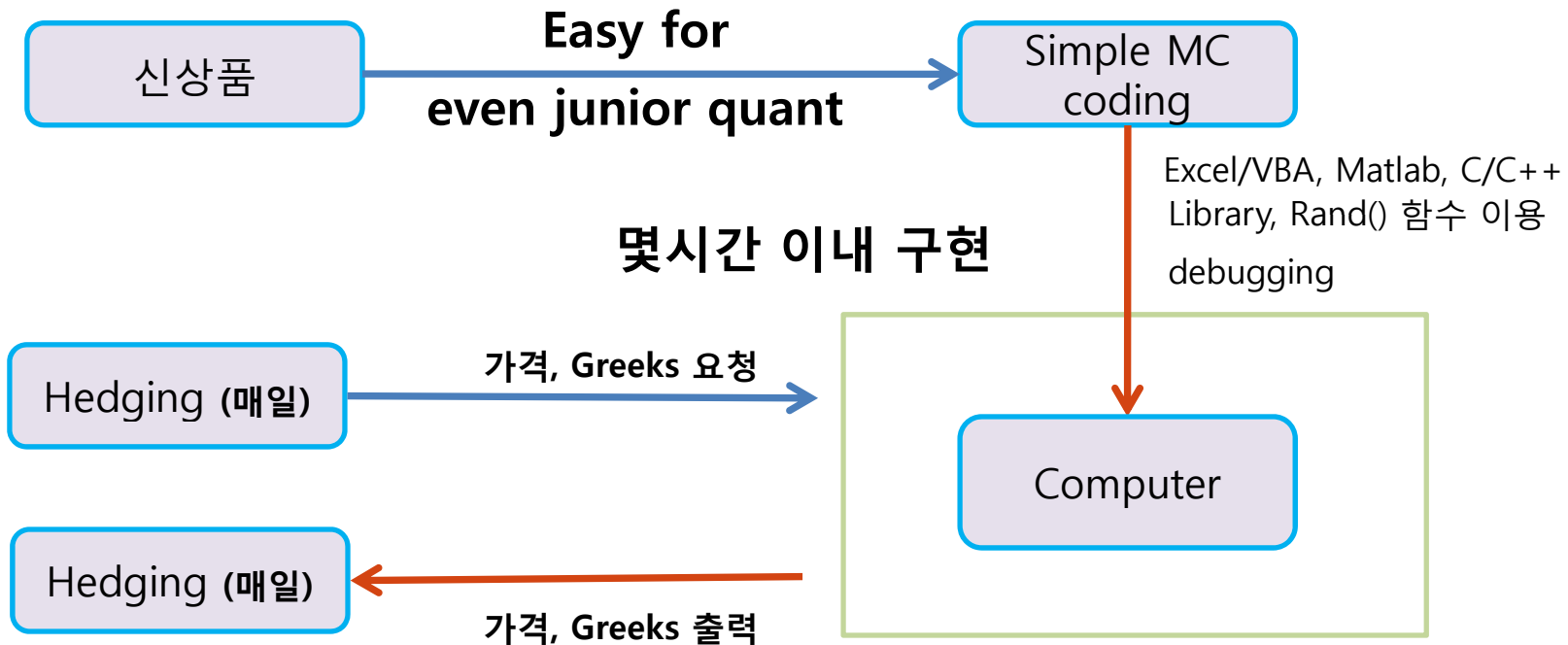


CUDA를 이용한 병렬화

몬테카를로 시뮬레이션

상품 발행시
 Pricing
 Overview of Greek
 Hedge Simulation
 조기상환 확률계산

상품거래시
 Pricing
 Greeks Search
 VaR계산, 위험관리

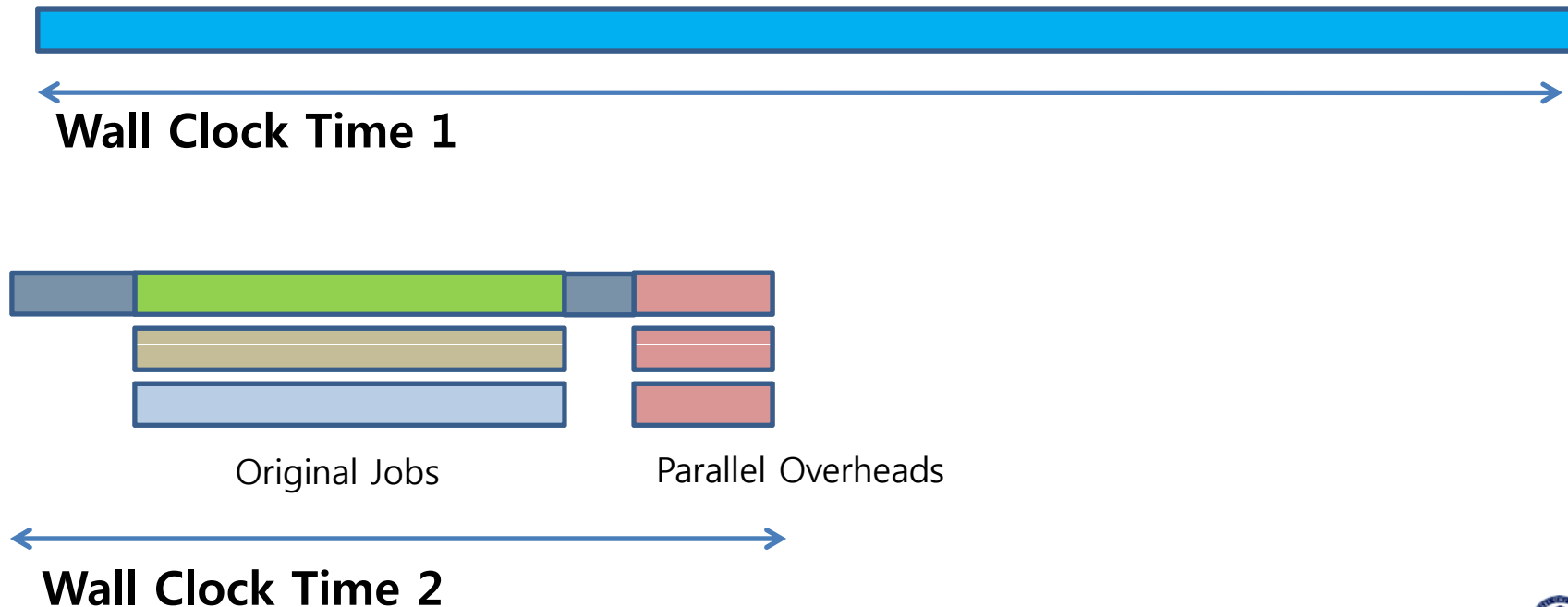


병렬처리란 (HPC – High performance Computing)

하나의 작업을 여러 개의 CPU(Cores)로
작업을 분할하여 실행시켜 계산시간을 줄이는 방법

Wall Clock Time

Task를 병렬처리에 의해 처리하는데 걸린 전체 시간



KISTI 슈퍼컴퓨터 4호기 **Tachyon**



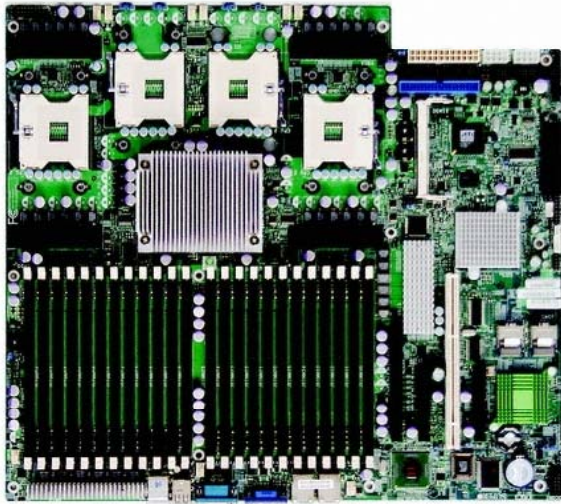
| 구분 | 내용 |
|-----------|--|
| 모델명 | SUN Blade 6048 |
| 블레이드 노드 | 성능 : 24TFlops(Rpeak) 노드 : 188개(컴퓨팅) |
| CPU | AMD Opteron 2.0GHz 4개 노드당 : 16 Core (Quad Core) 총 : 3,008 개 (전체) |
| 메모리 | 32GB(노드) |
| 스토리지 | 207TB(디스크), 422TB(테이프) |
| 노드 간 네트워크 | Infiniband 4X DDR |

금융권 HPC용 16코어 SMP머신



도입비용 약 2000만원 가량

HP ProLiant DL580G5 Rack Type
4U case 사용
인텔 제온 E7340 CPU 4개 : **16코어**
메모리 8GB



Node당 100Gflops 유지
Tachyon 1 노드와 유사한 성능

8노드 도입시 약 3억원 정도 예상됨

성능 : 64배 향상

국내 금융권 도입 : 비용대비 성능향상의 미약으로 HPC 도입이 잘 안됨

CUDAp workstation at Yonsei Math-Finance Lab



계산용 서버 : Fedora Linux + Intel Compiler + CUDA
Tesla C870 3개 장착 : 이론상 SP 1.5Tflops in SP

: 500만원 이하

보조용 워크스테이션 Windows XP + Visual Studio 2003 + CUDA
Tesla C870 1개 장착 : 이론상 SP 0.5Tflops in SP

비교

Kisti 슈퍼컴퓨터 4호기 : 전체 : DP 24Tflops : (40억원 이상),
하나의 노드 : DP 0.1Tflops

: 2천만원 이상

CUDA 병렬화

Single Node

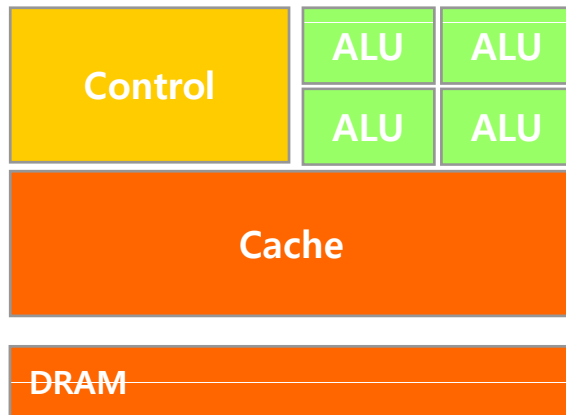
Intel CPU : 4 Cores X 4CUP = 16개 Core

Geforce 8800GTX, Tesla C870 : 128개 SP

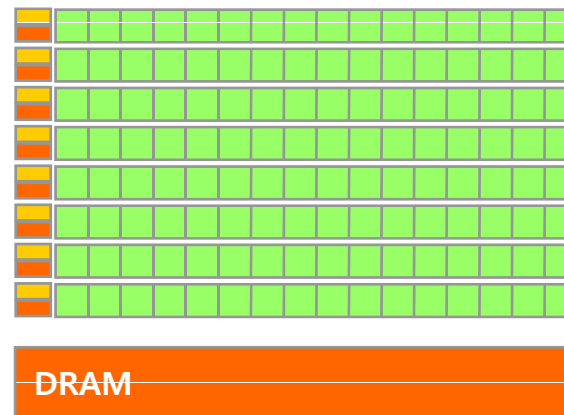
Tesla S870 : $128 * 4 = 512$ 개 SP

Multi Node

Single Node X N개 SP



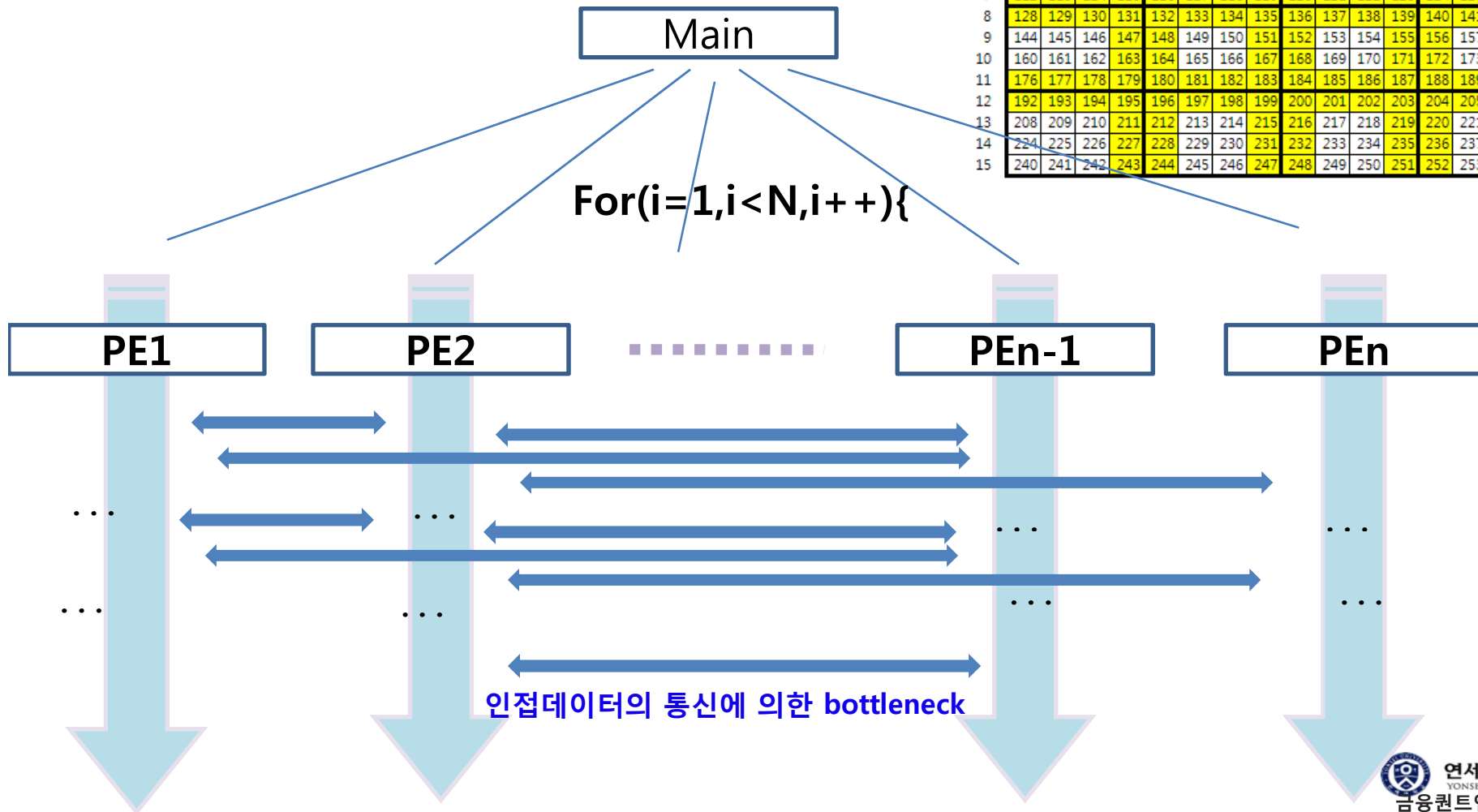
CPU



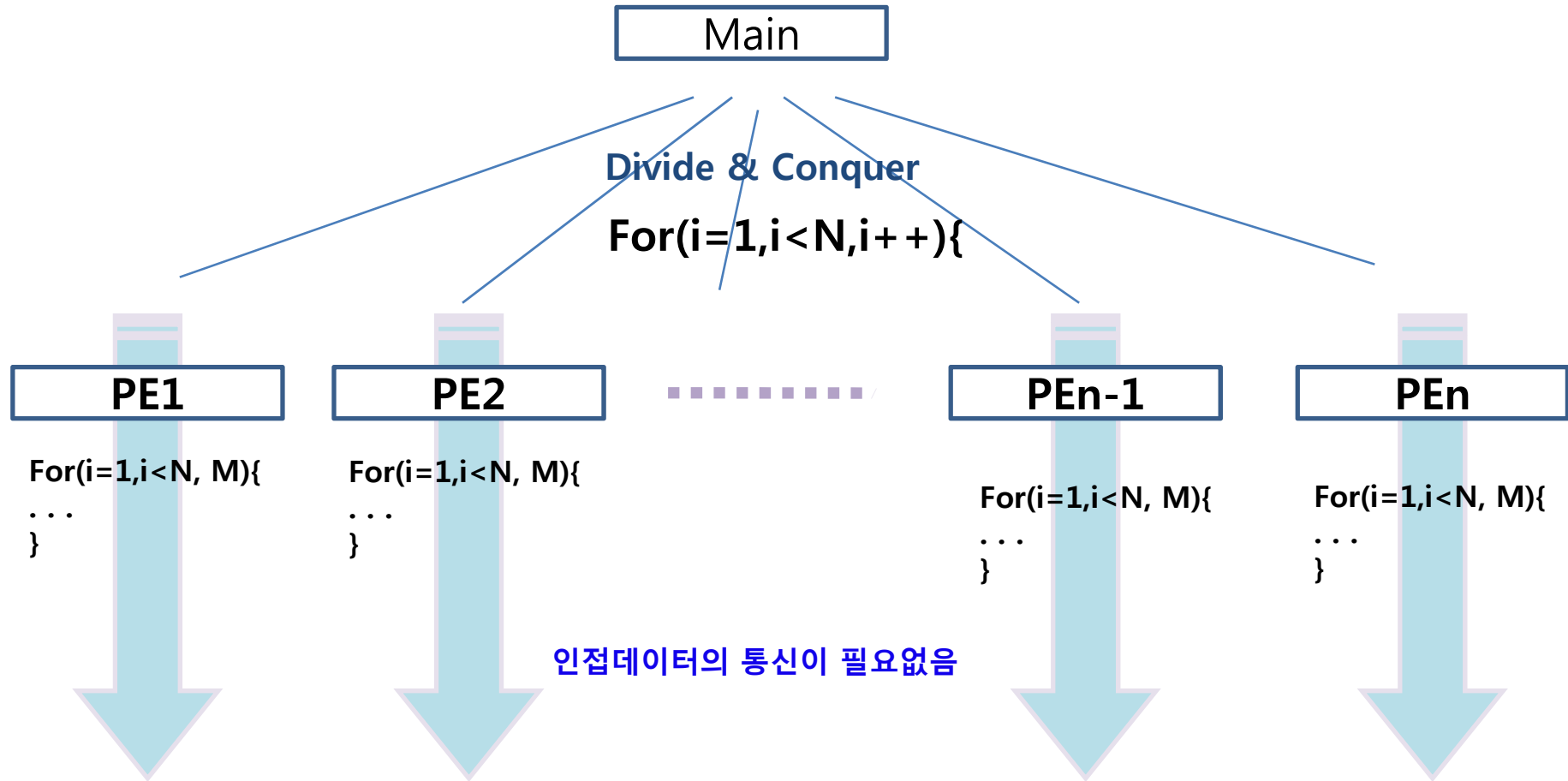
GPU

Parallel FDM, FEM (matrix)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 2 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 3 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 4 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 5 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 6 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 7 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 8 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 9 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 10 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 11 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| 12 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| 13 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 14 | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 15 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |



Parallel MC simulation

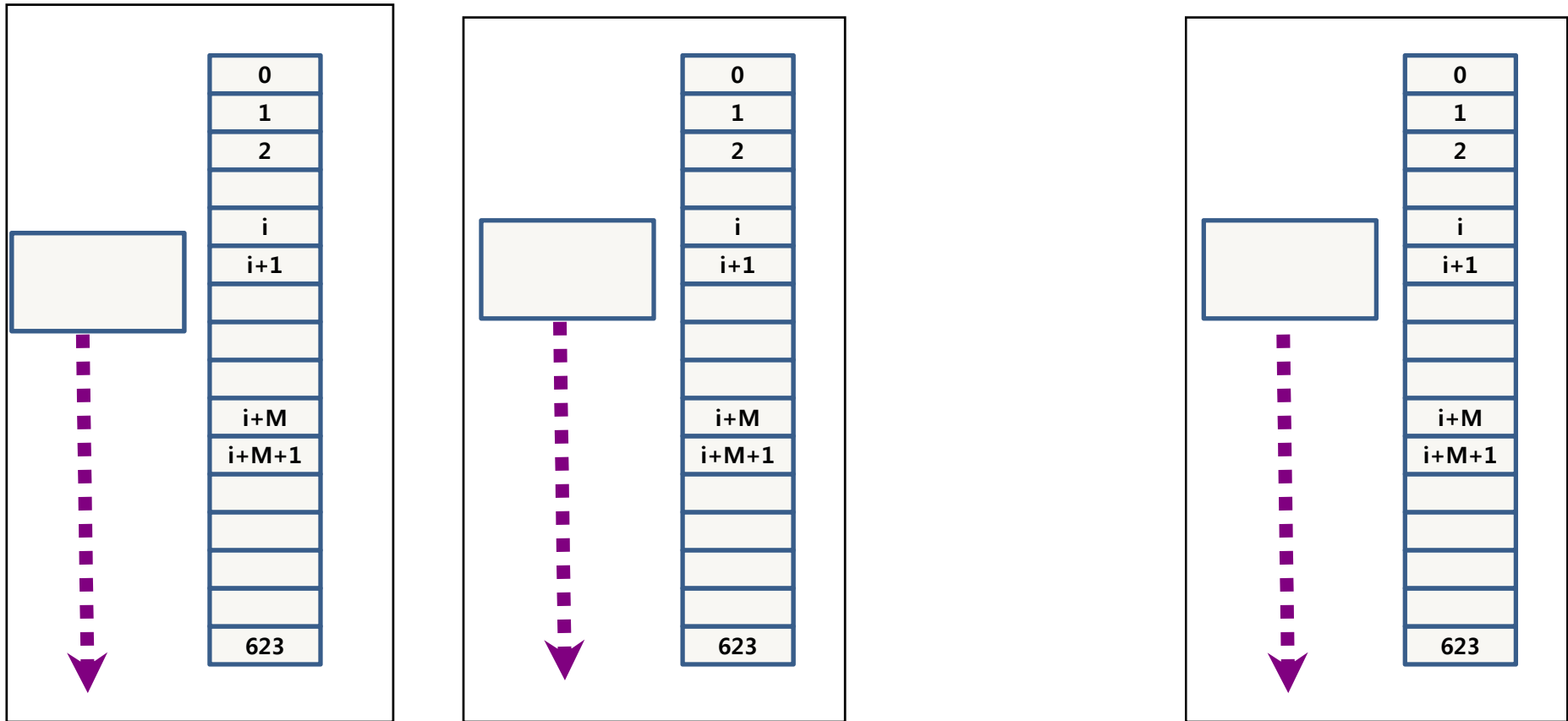


: MC 병렬 scalability가 좋다

Parallel sMT MT19937

Each Cores execute sMT independently

We will use this model



Single & CUDA ELS 실행속도

삼성증권 제1909회 ELS

MT19937 이용 (10만회)

| | CPU | 병렬화 | 1차 최적화 | 현재 최적화 |
|------------|----------|-------------|-----------|-----------|
| 중 | | | | |
| RNG | CPU 53초, | GPU1 7.3초, | GPU2 2.1초 | |
| BM 변환 | CPU 20초, | GPU1 3.6초, | GPU2 1.1초 | |
| Copy | CPU 0초 , | GPU1 18.3초, | GPU2 6.2초 | |
| MC | CPU 18초, | GPU1 1.2초, | GPU2 0.4초 | |
| 총 (12배) | 90초, | 30초(2.5배) | 9.8초(9배) | 7.1초 |

C언어 : AMD 페놈 2.5Ghz, GPU vs. Tesla C870
MT607, LCG(rand48) 이용시 40X~80X 성능 향상 예상됨

향후 전망 (CUDA를 이용한 금융)

1X
90.0sec

CPU 사용

최신 3GHz CPU 사용

80X
1.1sec

총 128 SP 사용

C870 1개 사용

← CUDA적용 (최적화 필요, MT607)

300X
0.3sec

총 512 SP 사용

C870 4개 사용 = S870 1대 사용

← multiGPU (pthread, openMP) 적용

500X
0.2sec

총 960 SP 사용

C1060 4개 사용 = S1070 1대 사용

← 차세대 보드 적용

1000X
0.09 sec

총 1920 SP 사용

C1060 8개 사용 = S1070 2대 사용

← multiGPU (MPI) 적용

1500X
0.06 sec

총 3840 SP 사용

C1060 16개 사용 = S1070 4대 사용

← MT19937 필요 : 속도 감소 예상됨

3000X
0.03 sec

총 3840 SP 사용

C1060 32개 사용 = S1070 8대 사용 (구축시 H/W 2억원 정도 예상됨)

6X 8 node CPU cluster 비용보다 저렴 성능 : 60배 차이

현재 : 100개 상품을 9000sec = 2.5시간에 계산함
 곧 실현될 미래 : 100개 상품을 2sec 만에 계산함

The END

경청해 주셔서 감사합니다.

유현곤

연세대학교 수학과 대학원 박사과정

yhgon@yonsei.ac.kr