

SnuMAP

운영체제 수준의 멀티코어 어플리케이션 프로파일러
<https://github.com/SnuMAP/SnuMAP>



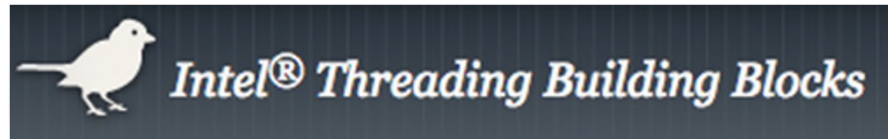
Computer Systems and Platforms Laboratory
School of Computer Science and Engineering
Seoul National University

개발하게 된 배경

멀티 코어 컴퓨팅 시대

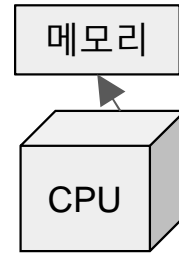
- ❖ 공짜 점심은 없다
 - 멀티 코어 (멀티 스레드) 컴퓨팅이 필수적임
- ❖ 멀티 코어 컴퓨팅이 어디에 쓰이는가?
 - 모바일/데스크탑 컴퓨터
 - 데이터 센터
 - 슈퍼컴퓨팅 센터
 - 지금, 어디에서나
- ❖ 하지만, 엄청나게 어려운 멀티 코어 프로그래밍

멀티 코어 프로그래밍 방법의 많은 발전



플랫폼의 변화

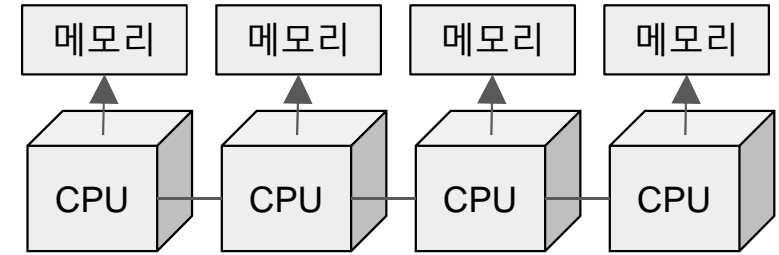
❖ 멀티/매니 코어 플랫폼



UMA 구조

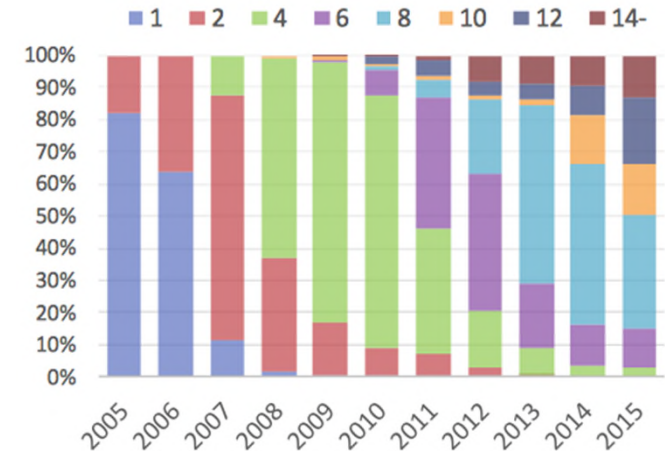
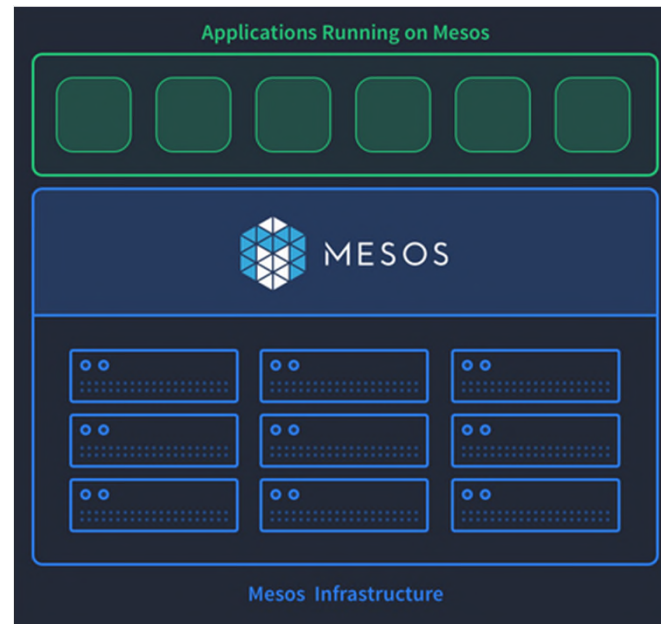
UMA: Uniform Memory Access

NUMA: Non-Uniform Memory Access



NUMA 구조

❖ 데이터 센터 / 슈퍼컴퓨팅 센터



Top 500 슈퍼컴퓨터의
CPU 소켓 당 코어 개수
(출처: 미국 Argonne national lab.)

우리가 작성한 어플리케이션들이 멀티 코어에서 정말로 잘 동작할까?

❖ 성능 버그 / 스케줄링 버그

- 어플리케이션 프로그램 코드를 잘 작성했는가?
 - 병렬 처리를 위해서 작업이 잘 분배가 되었을까?
 - 복잡한 메모리 구조를 잘 활용하였는가?
 - 시스템의 다른 워크로드들과 잘 어우러져 실행되는가?
-
- 성능 버그 및 스케줄링 버그는 컴퓨팅 플랫폼의 효율성에 큰 영향을 미친다

성능 버그를 찾기 위한 도구들



PIN: 인텔에서 만든 어플리케이션 성능 분석기
명령어 수준에서 어플리케이션의 성능을 자세하게 분석하지만 어플리케이션 성능이 크게 떨어진다.



HPCToolkit
Rice 대학에서 시작한 병렬 처리 어플리케이션 프로파일러 [BSD3]
병렬 어플리케이션의 실행 흐름을 샘플링 기술을 통해 수집하여 시각화하는 도구.



리눅스 운영체제에서 동작하는 어플리케이션의 성능 분석기 [GPL2]
리눅스의 수준에서 성능 분석을 제공하지만 어플리케이션 성능이 크게 떨어진다.

```
$ oprofile --exclude-dependent --demangle-smart --symbols `which lyx`
CPU: PIII, speed 863.195 MHz (estimated)
Counted CPU_CLK_UNHALTED events (clocks processor is not halted) with a unit mask of 0x00 (No unit)
vma      samples  %      symbol name
081ec974 5016    8.5096  _Rb_tree<unsigned short, pair<unsigned short const, int>, unsigned
0810c4ec 3323    5.6375  Paragraph::getFontSettings(BufferParams const&, int) const
081319d8 3220    5.4627  LyXText::setFont(Buffer const&, Paragraph*, int) const
080e45d8 3011    5.1082  LyXFont::realize(LyXFont const&)
080e3d78 2623    4.4499  LyXFont::LyXFont()
081255a4 1823    3.0927  LyXText::singleWidth(BufferView*, Paragraph*, int, char) const
080e3cf0 1804    3.0605  operator==(LyXFont::FontBits const&, LyXFont::FontBits const&)
081128e0 1729    2.9332  Paragraph::Pimpl::getChar(int) const
081ed020 1380    2.3412  font_metrics::width(char const&, unsigned, LyXFont const&)
08110d60 1310    2.2224  Paragraph::getChar(int) const
081ebc94 1227    2.0816  qfont_loader::getfontinfo(LyXFont const&)
...
```

기존 도구들의 한계점

어플리케이션 성능의 감소 및 부족한 시각화 기능
단일 어플리케이션만을 고려한 실행 흐름 수집



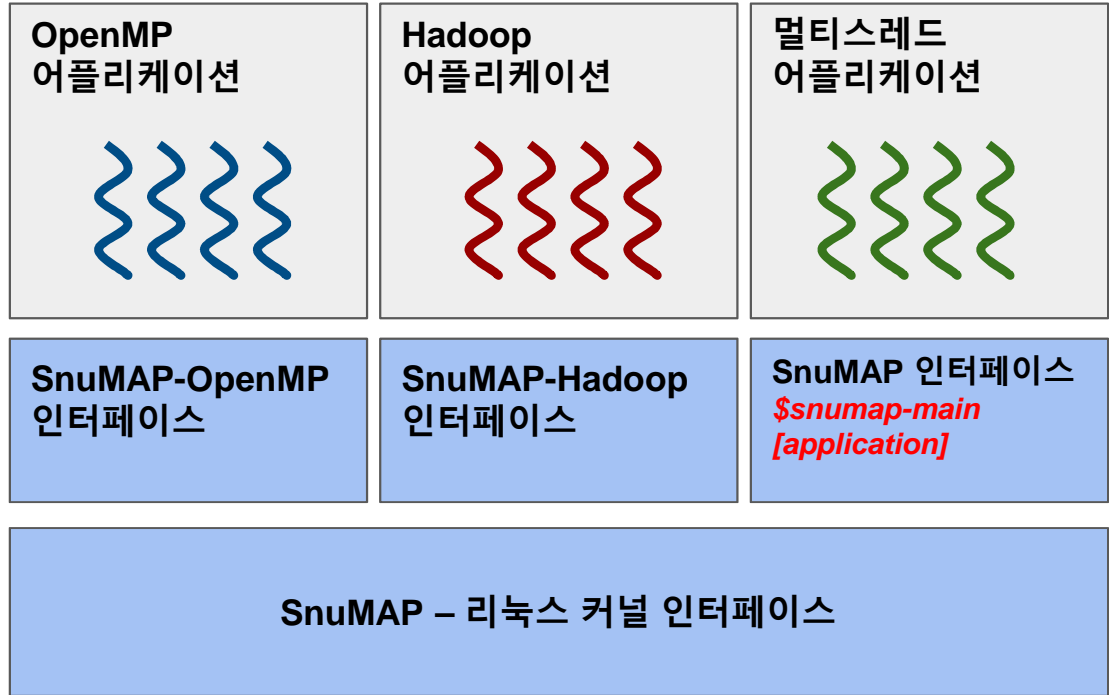
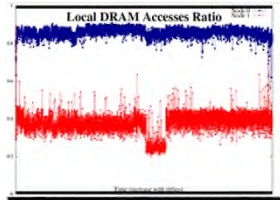
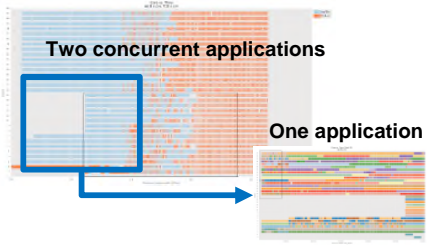
운영체제의 영향까지 고려하면서도 어플리케이션 성능을 떨어뜨리지 않는 실행 흐름 분석 도구가 없다.

SnuMAP

SnuMAP

- ❖ **오픈 소스 멀티 코어 어플리케이션 성능 프로파일러**
 - 성능 버그 및 스케줄링 버그를 탐지할 수 있는 실행 정보 제공
 - 어플리케이션 개발자, 멀티코어 플랫폼 관리자 모두에게 유용함
- ❖ **어플리케이션 성능에 영향을 미치지 않는 경량 프로파일러**
 - 운영체제 수준에서 어플리케이션 실행 흐름 및 메모리 접근 패턴 분석
 - 리눅스 커널 패치가 필요하나 하드웨어 플랫폼에 무관하게 동작
- ❖ **다양한 어플리케이션 프로그래밍 방법을 확장적으로 지원**
 - 현재 Pthread / OpenMP / Hadoop 어플리케이션을 지원
 - SnuMAP 을 사용하기 위해서 SnuMAP API 의 포팅을 필요로 함

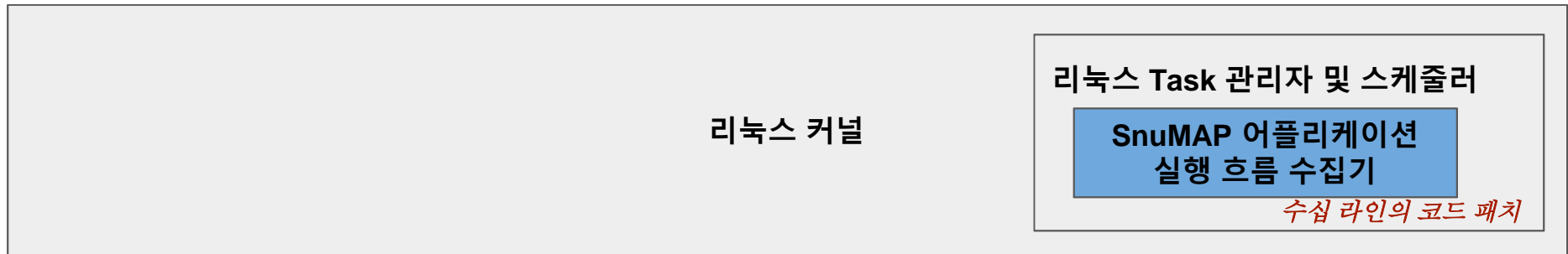
SnuMAP 구조



SnuMAP
어플리케이션
실행 흐름 분석기
\$snumap-plot [log1]
[log2] ...

SnuMAP
어플리케이션
메모리 접근 분석기
\$snumap-numa [log1]
[log2] ...

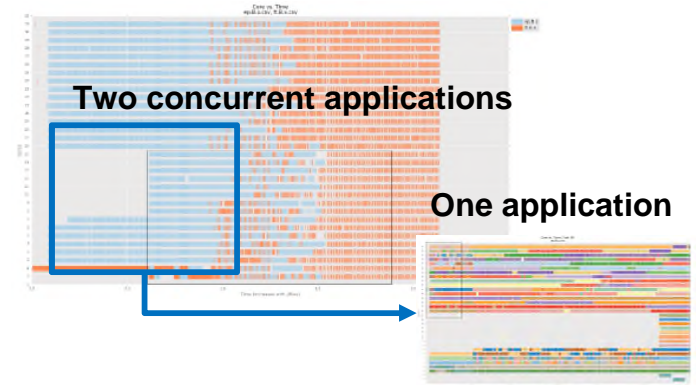
User-space
Kernel-space



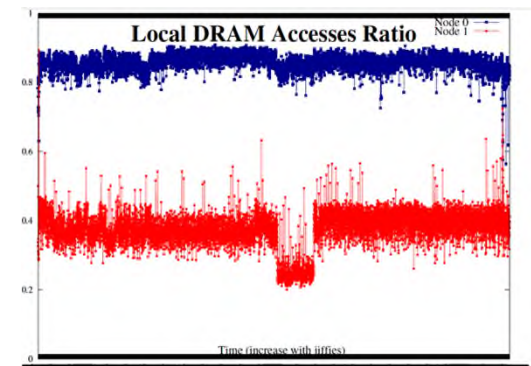
멀티/매니 코어 하드웨어

SnuMAP 이 제공하는 성능 정보

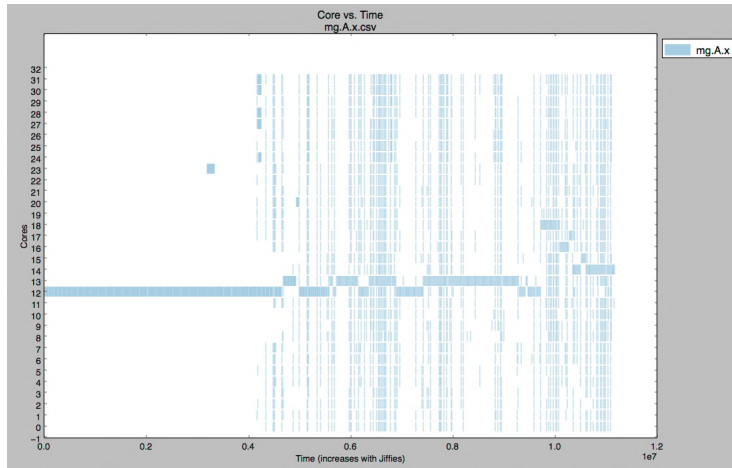
❖ 멀티 코어 플랫폼에서 실행되는 어플리케이션들의 실행 흐름 분석



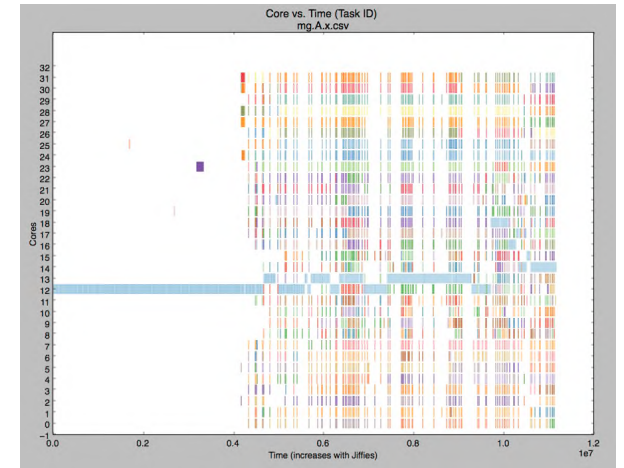
❖ 어플리케이션의 메모리 접근 패턴 분석



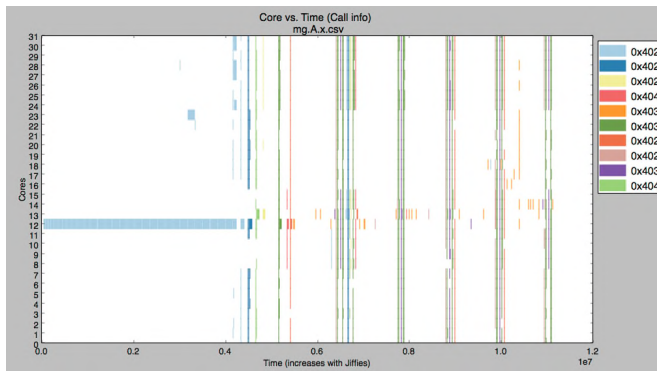
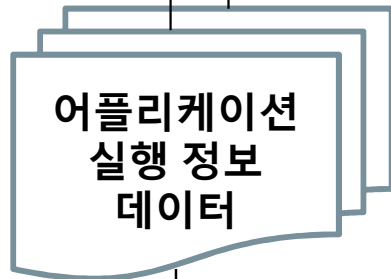
어플리케이션 실행 흐름 분석



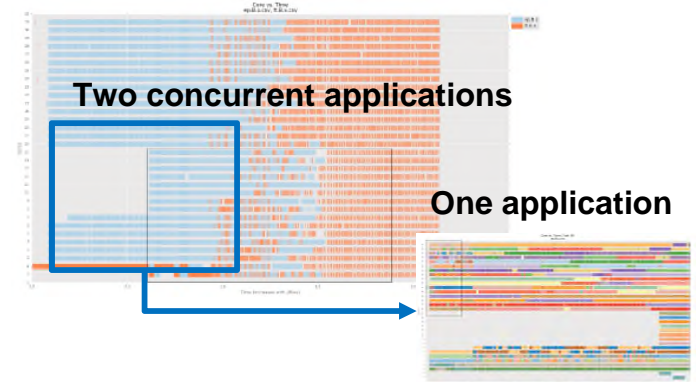
어플리케이션 단위의 실행 흐름



어플리케이션 스레드 별 실행 흐름



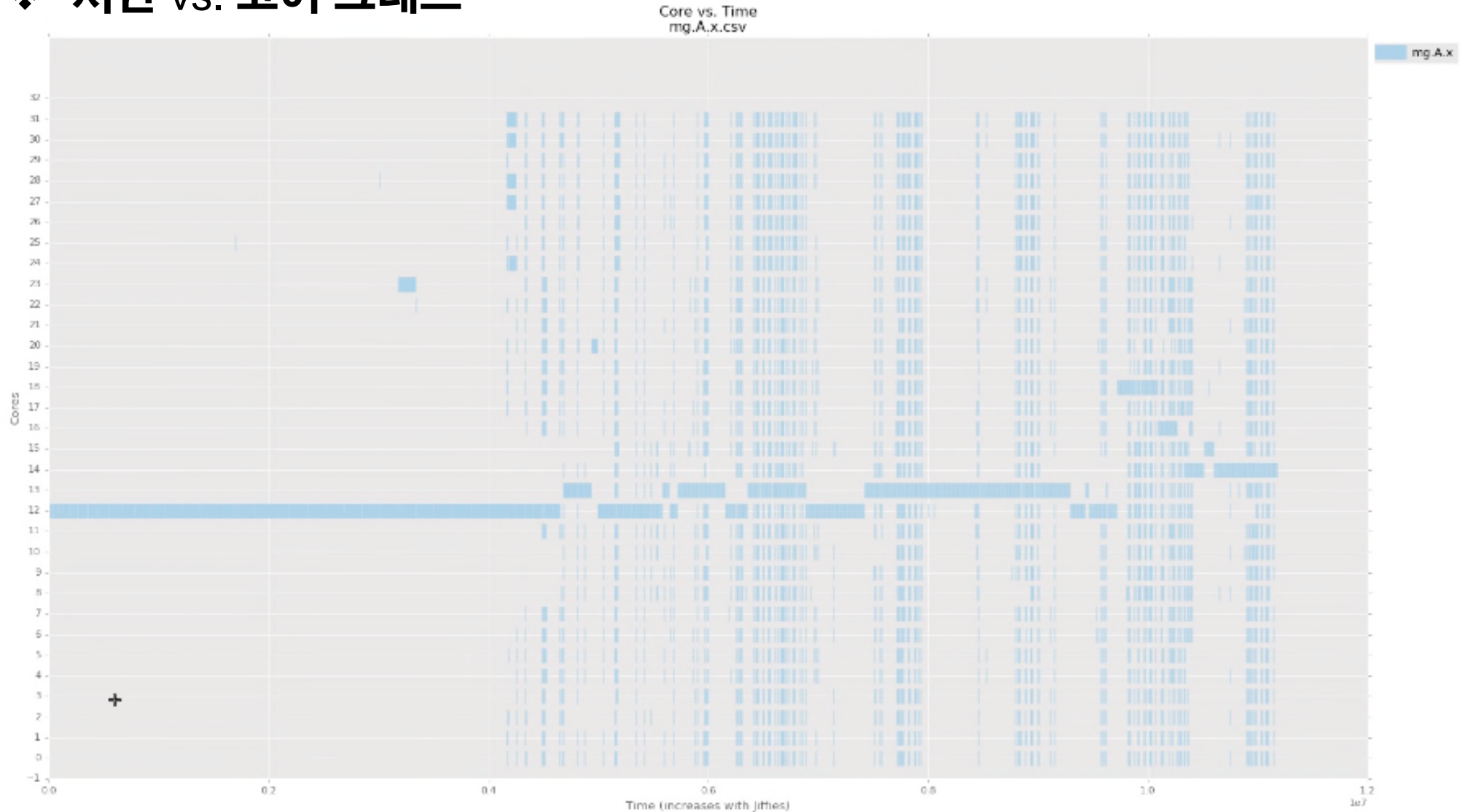
어플리케이션 코드 영역별 흐름



다중 어플리케이션들의 실행 흐름

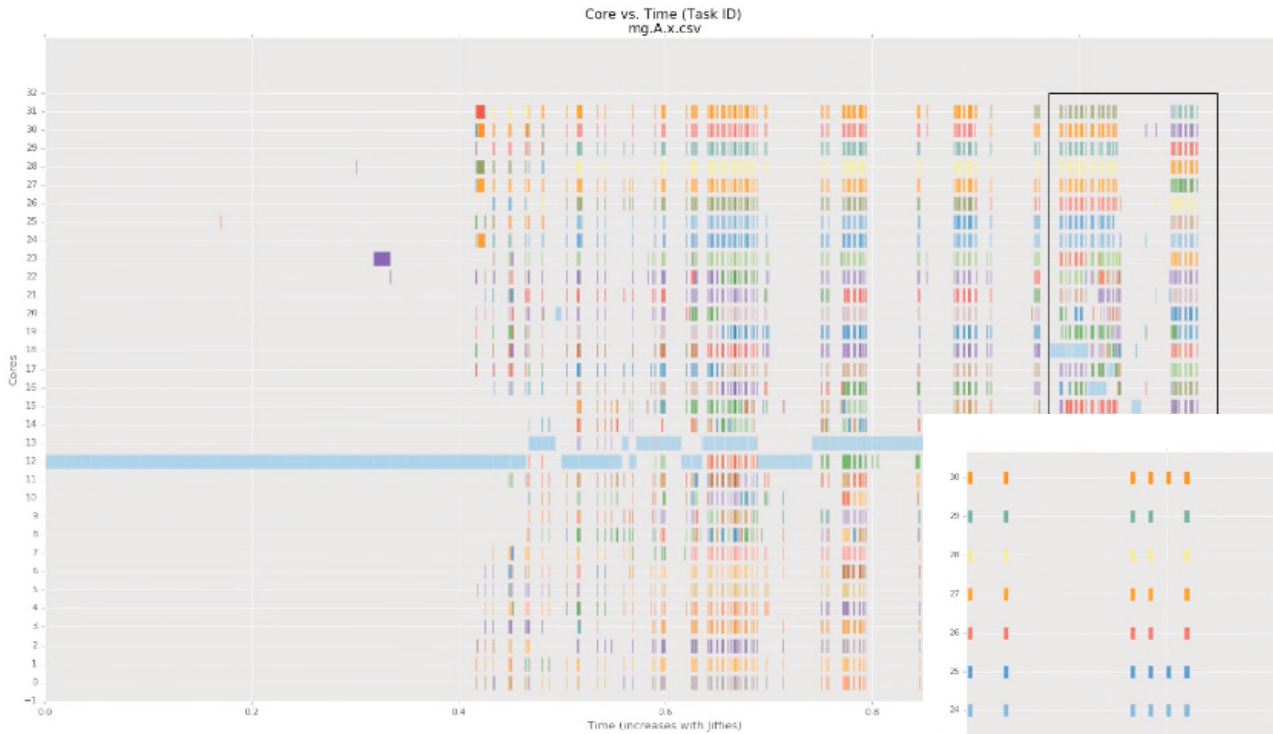
어플리케이션 단위의 실행 흐름

❖ 시간 vs. 코어 그래프

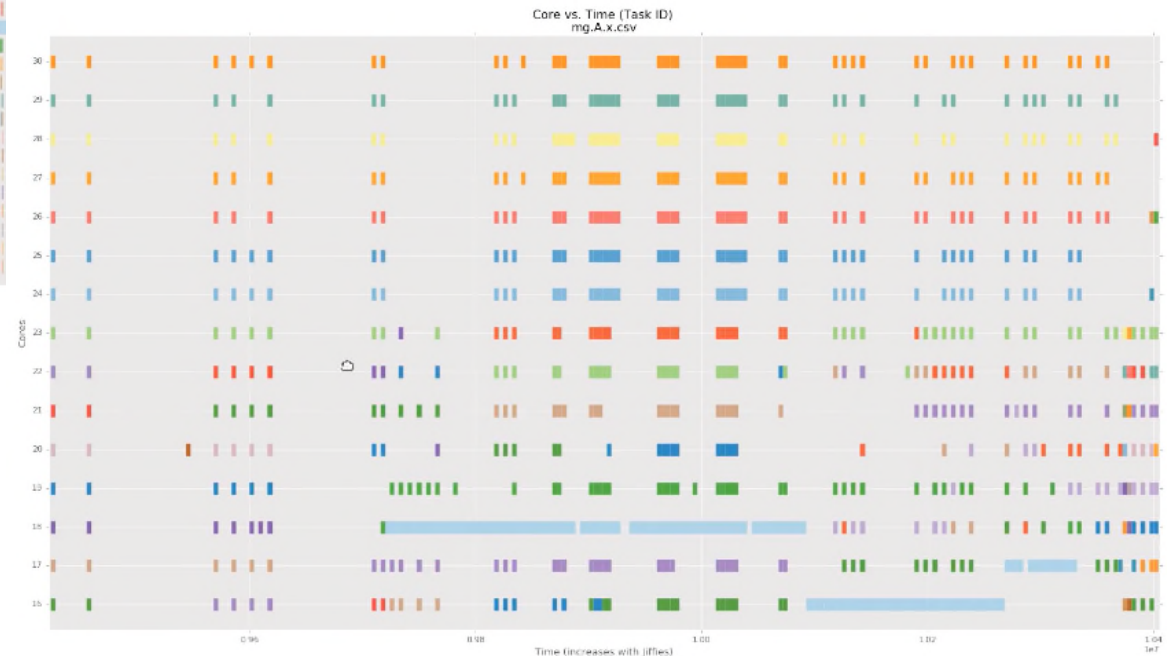
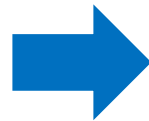


어플리케이션 스레드 실행 흐름

❖ 어플리케이션 내부 스레드들의 실행 흐름 분석

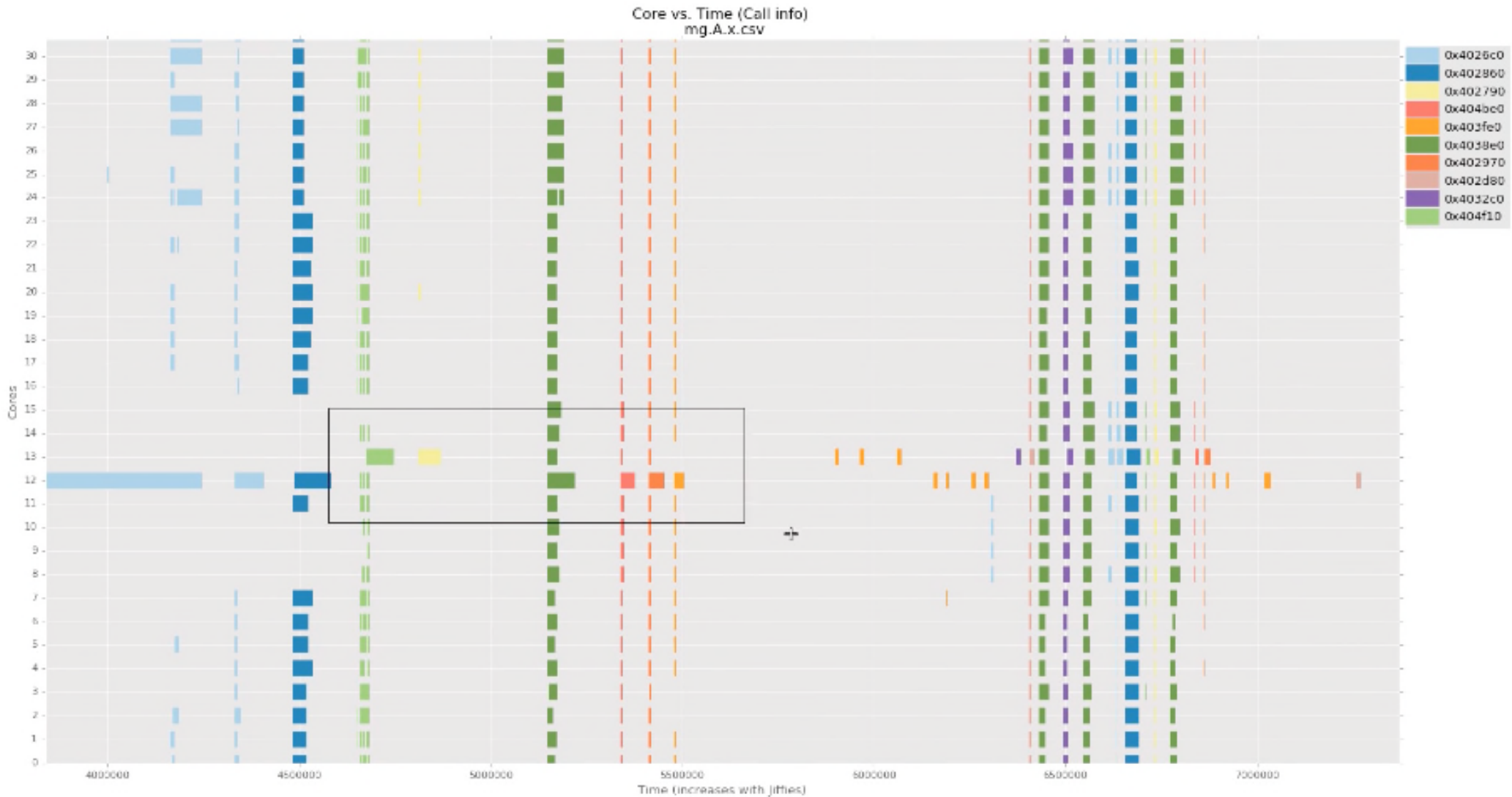


Zoom in 기능 제공



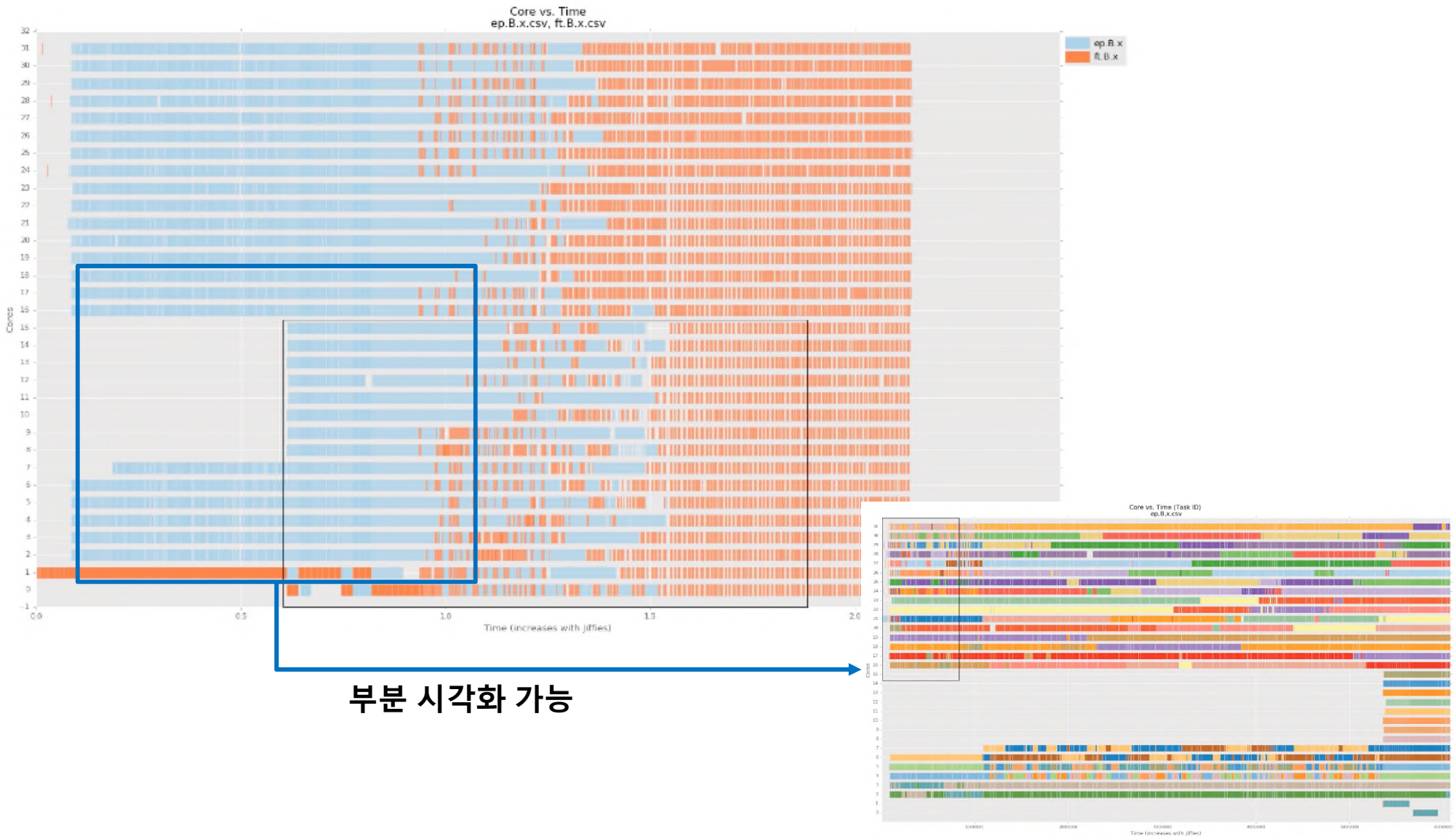
코드 영역 별 실행 흐름

❖ 어플리케이션 내부의 코드 영역별 실행 흐름 분석



다중 어플리케이션 실행 흐름

❖ 여러 워크로드들이 효과적으로 잘 실행될 수 있는가?



SnuMAP 의 기대 효과

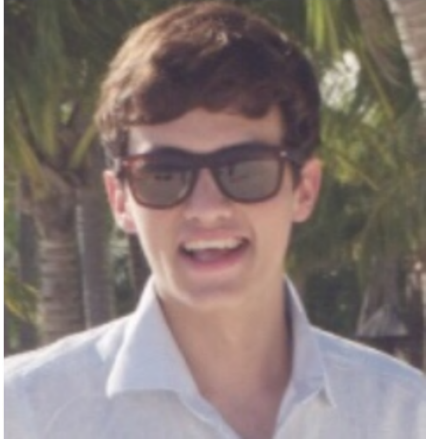
SnuMAP 을 통해서 할 수 있는 것

1. 어플리케이션의 성능 버그 및 스케줄링 버그를 탐지해서
어플리케이션의 성능 및 멀티 코어 자원 관리 방법 튜닝
2. 여러 어플리케이션의 효과적인 co-scheduling 을 통한
플랫폼의 효율 상승

SnuMAP 테스트 환경

- ❖ 일반적인 멀티 코어에서 동작 가능
 - 64코어 및 32코어 AMD Opteron 서버, 36코어 타일레라 시스템 등에서 테스트 완료
- ❖ 일반적인 멀티 스레드 어플리케이션의 프로파일링 가능
 - **멀티 스레드 어플리케이션, OpenMP 어플리케이션 및 Hadoop 어플리케이션에서 테스트 완료**
- ❖ 커널 패치
 - **리눅스 2.x 및 3.x 에서 테스트 완료**

Contributors



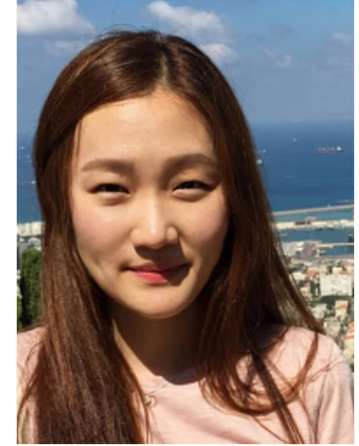
카밀로
(서울대학교)



신희식
(SAP Korea)



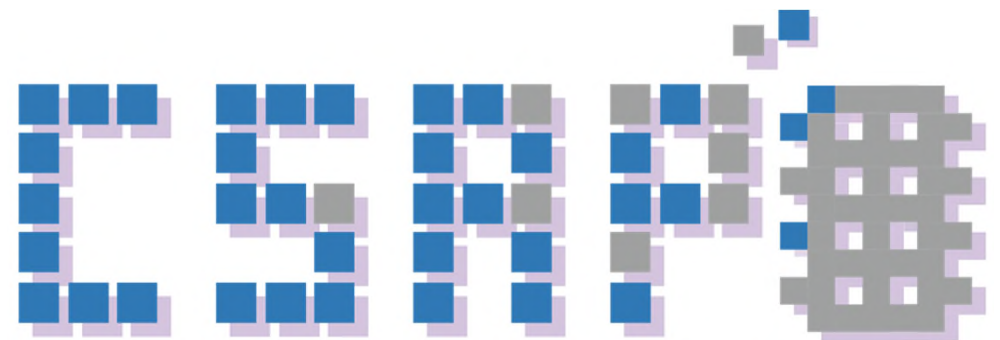
조영현
(서울대학교)



오수림
(서울대학교)



Bernhard Egger
(서울대학교)



Computer Systems and Platforms Laboratory
School of Computer Science and Engineering
Seoul National University