Regaining Lost Seconds:
Efficient Page Preloading for SGX Enclaves

Ximing Liu, Wang Lizhi,
Xiaoli Gong*, Ziyi Zhao
Nankai University

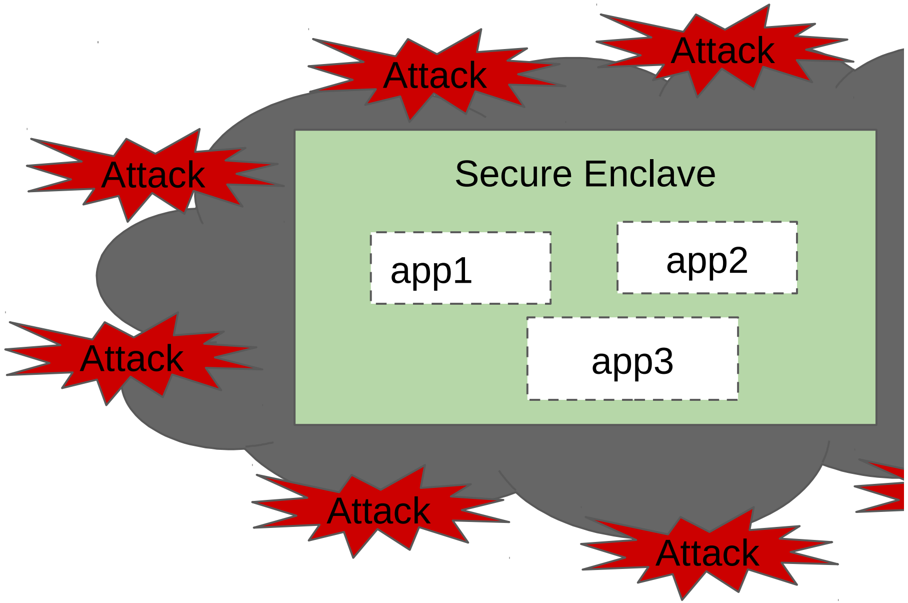
Wenwen Wang

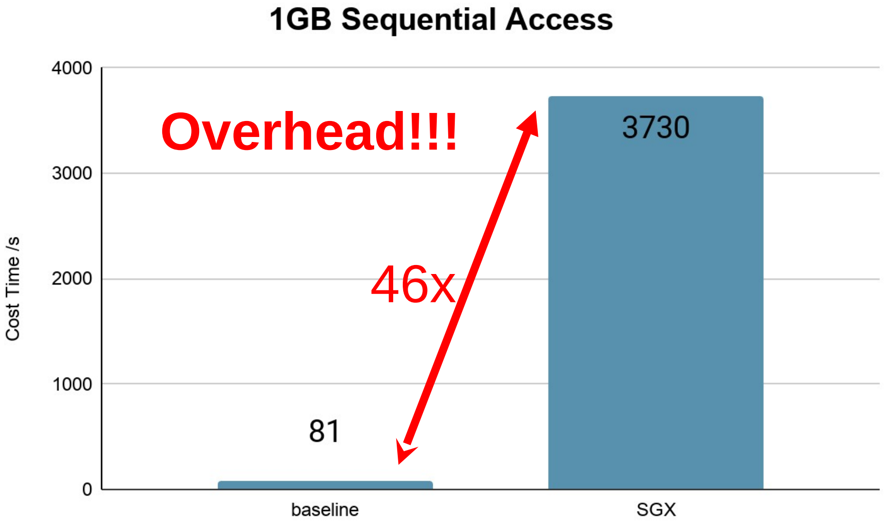
Univeristy of Georgia

Pen-Chung Yew

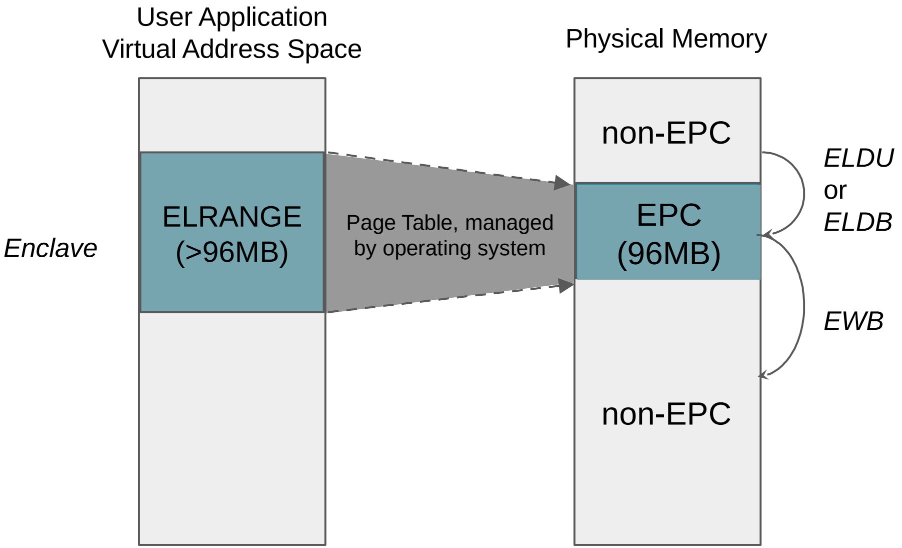
Univeristy of Minnesota

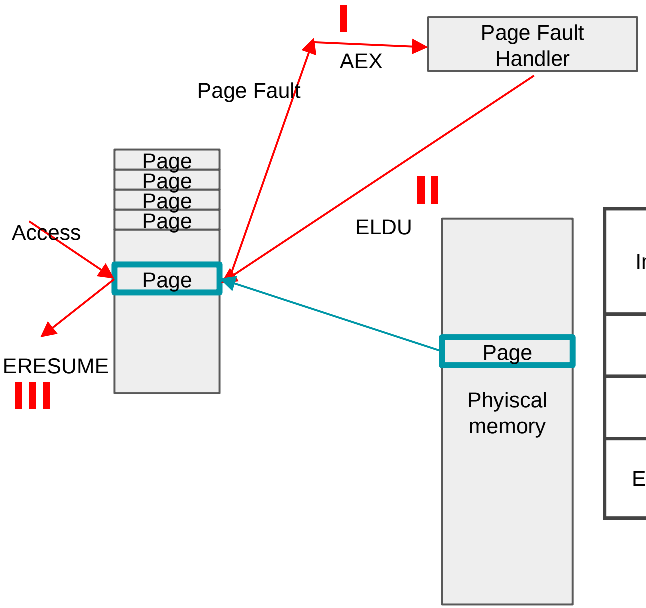
Intel(R) SGX Comes with Significant Overheads





Enclave Page Cache (EPC) Crucial to Performance

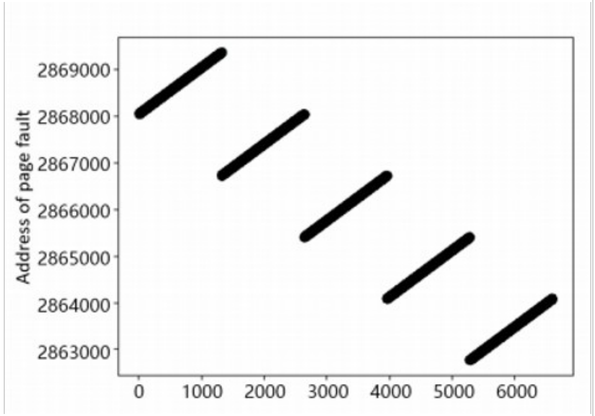




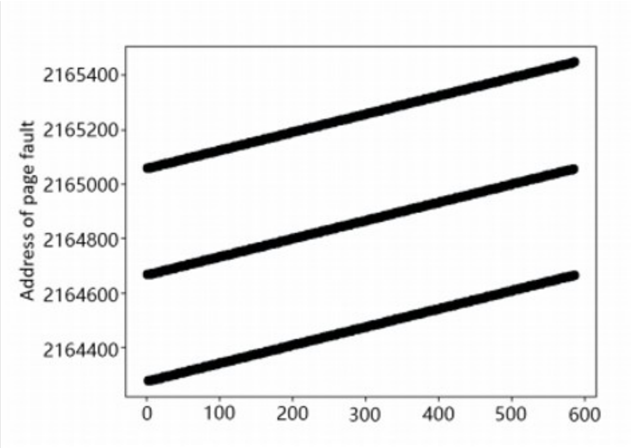
Instruction	Time/ **cycles**
AEX	10000
ELDU	44000
ERESUME	10000

Can We Hide SGX Page Fault Latency Using Data Prefetching?

Taking Advantage of Application Access Patterns

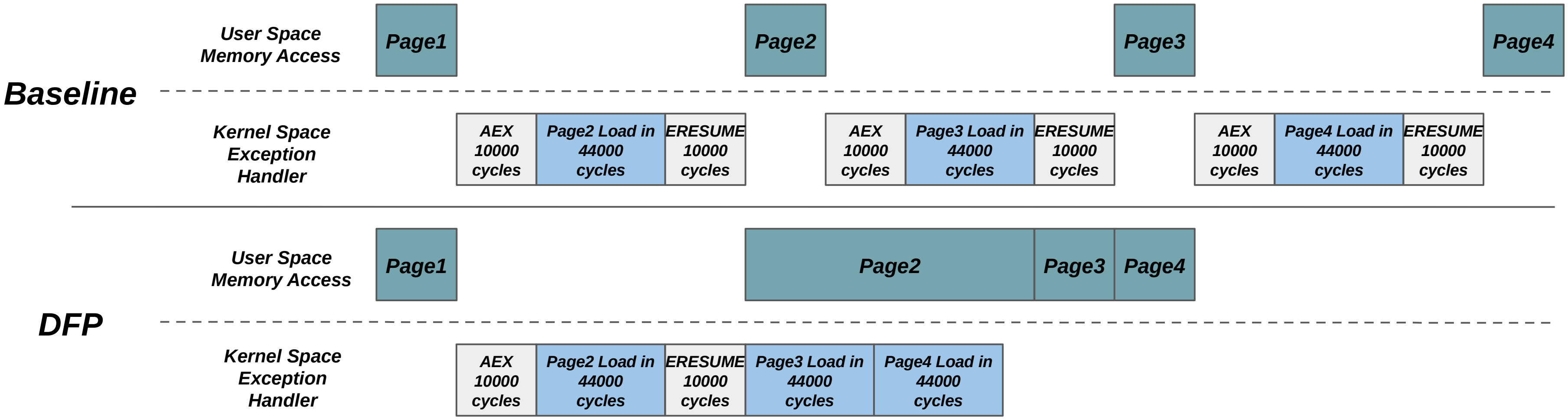


bwaves



lbm

DFP - Dynamic Fault History-Based Preloading



Baseline

User Space Memory Access

Page1 Page2 Page3 Page4

Kernel Space Exception Handler

AEX 10000 cycles | Page2 Load in 44000 cycles | ERESUME 10000 cycles

AEX 10000 cycles | Page3 Load in 44000 cycles | ERESUME 10000 cycles

AEX 10000 cycles | Page4 Load in 44000 cycles | ERESUME 10000 cycles

DFP

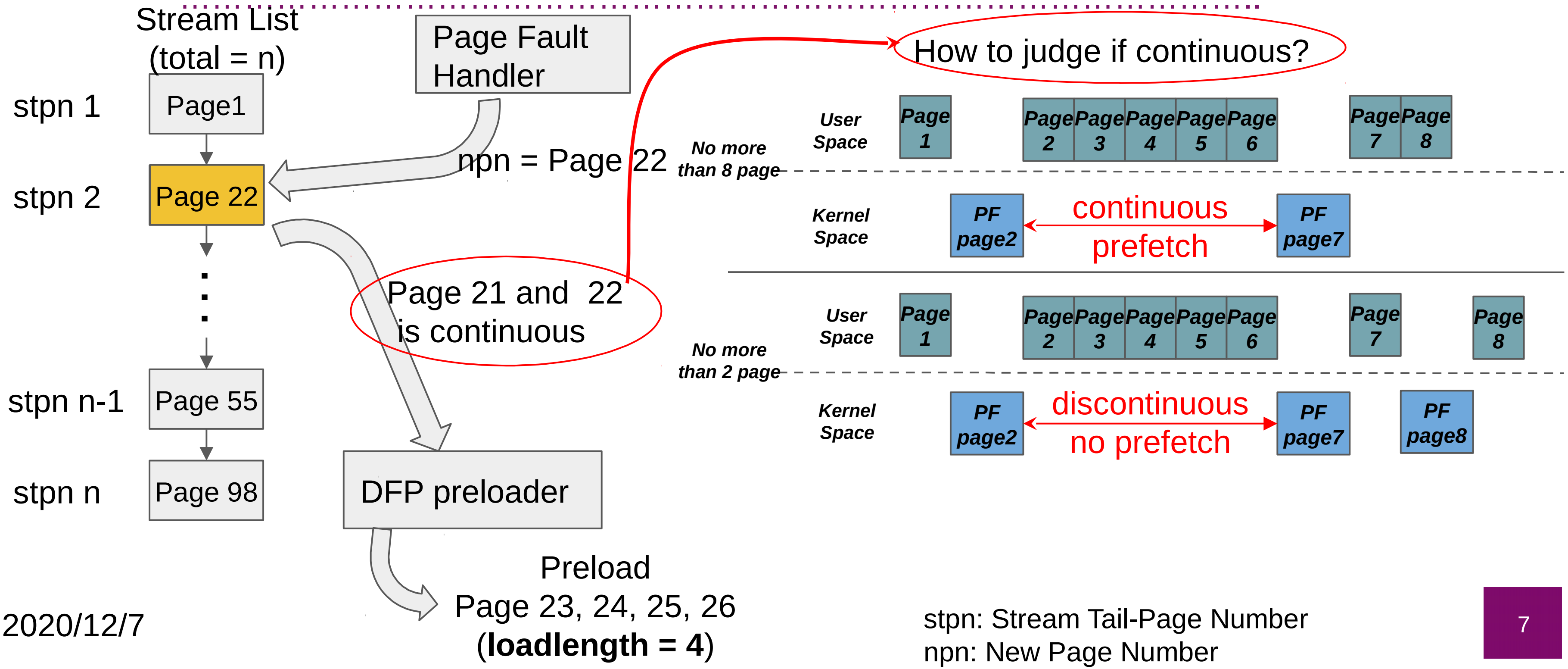
User Space Memory Access

Page1 Page2 Page3 Page4

Kernel Space Exception Handler

AEX 10000 cycles | Page2 Load in 44000 cycles | ERESUME 10000 cycles | Page3 Load in 44000 cycles | Page4 Load in 44000 cycles

Multiple-Stream Predictor Algorithm in DFP



Stream List (total = n)

stpn 1 Page1

stpn 2 Page 22

stpn n-1 Page 55

stpn n Page 98

Page Fault Handler

npn = Page 22

Page 21 and 22 is continuous

DFP preloader

Preload Page 23, 24, 25, 26 (**loadlength = 4**)

How to judge if continuous?

No more than 8 page

User Space — *Page 1*, *Page 2*, *Page 3*, *Page 4*, *Page 5*, *Page 6*, *Page 7*, *Page 8*

Kernel Space — *PF page2* continuous prefetch *PF page7*

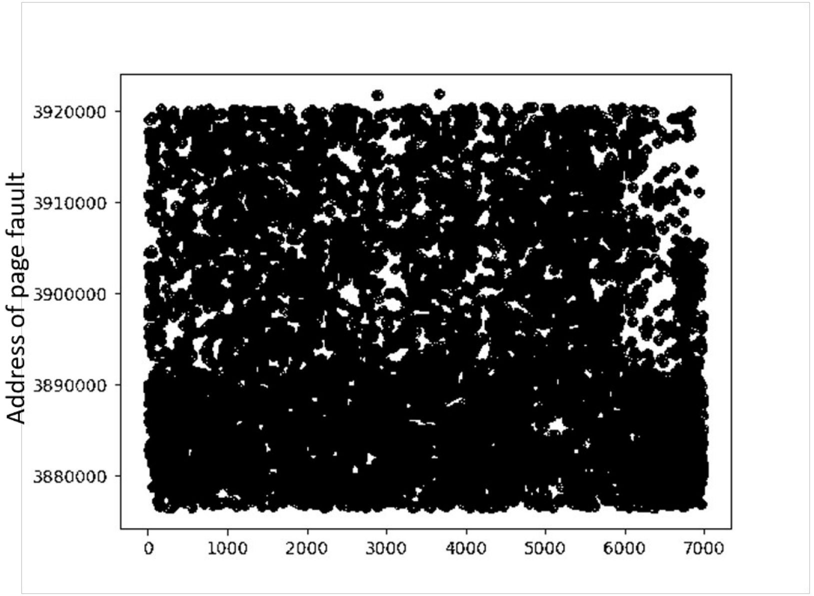
No more than 2 page

User Space — *Page 1*, *Page 2*, *Page 3*, *Page 4*, *Page 5*, *Page 6*, *Page 7*, *Page 8*

Kernel Space — *PF page2* discontinuous no prefetch *PF page7* *PF page8*

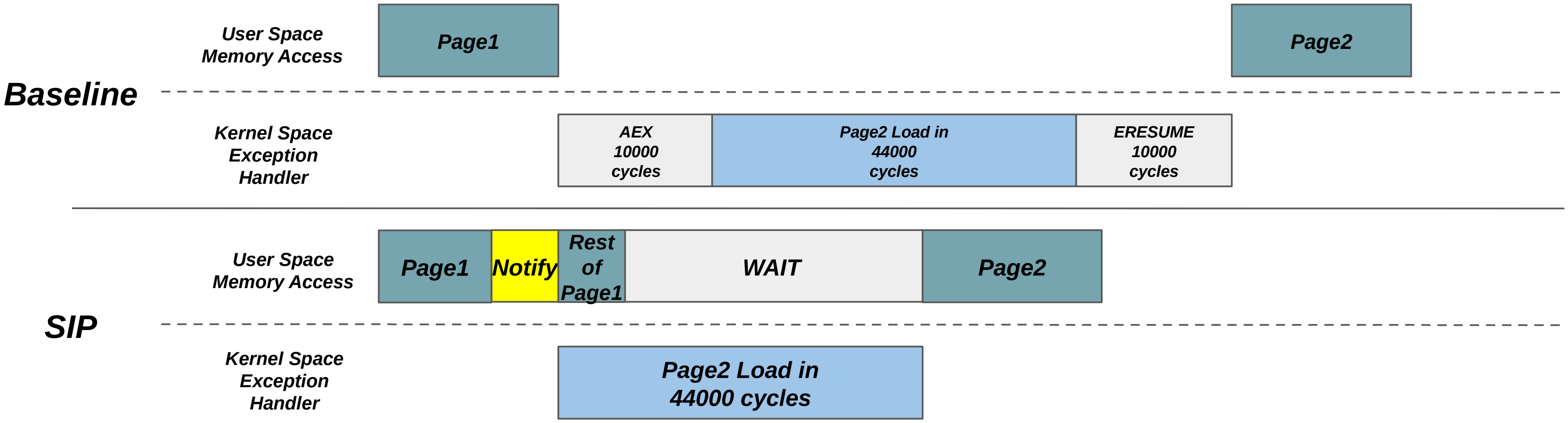
stpn: Stream Tail-Page Number
npn: New Page Number

Apps with Irregular/Unpredictable Access Patterns

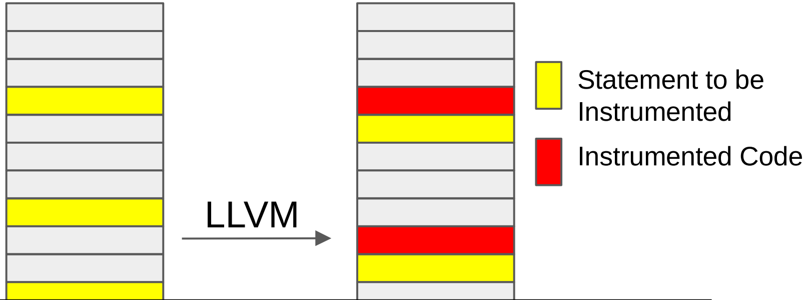


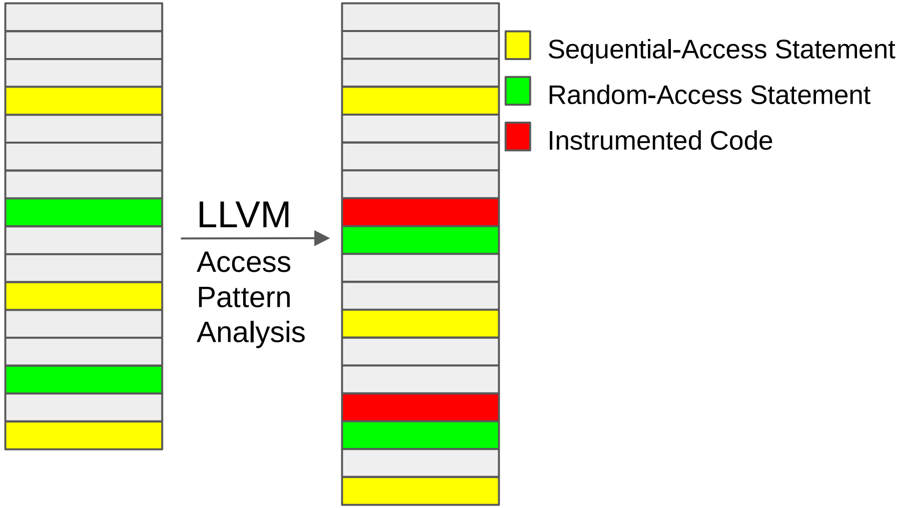
deepsjeng

SIP - Source-Level Instrumentation-Based Preloading



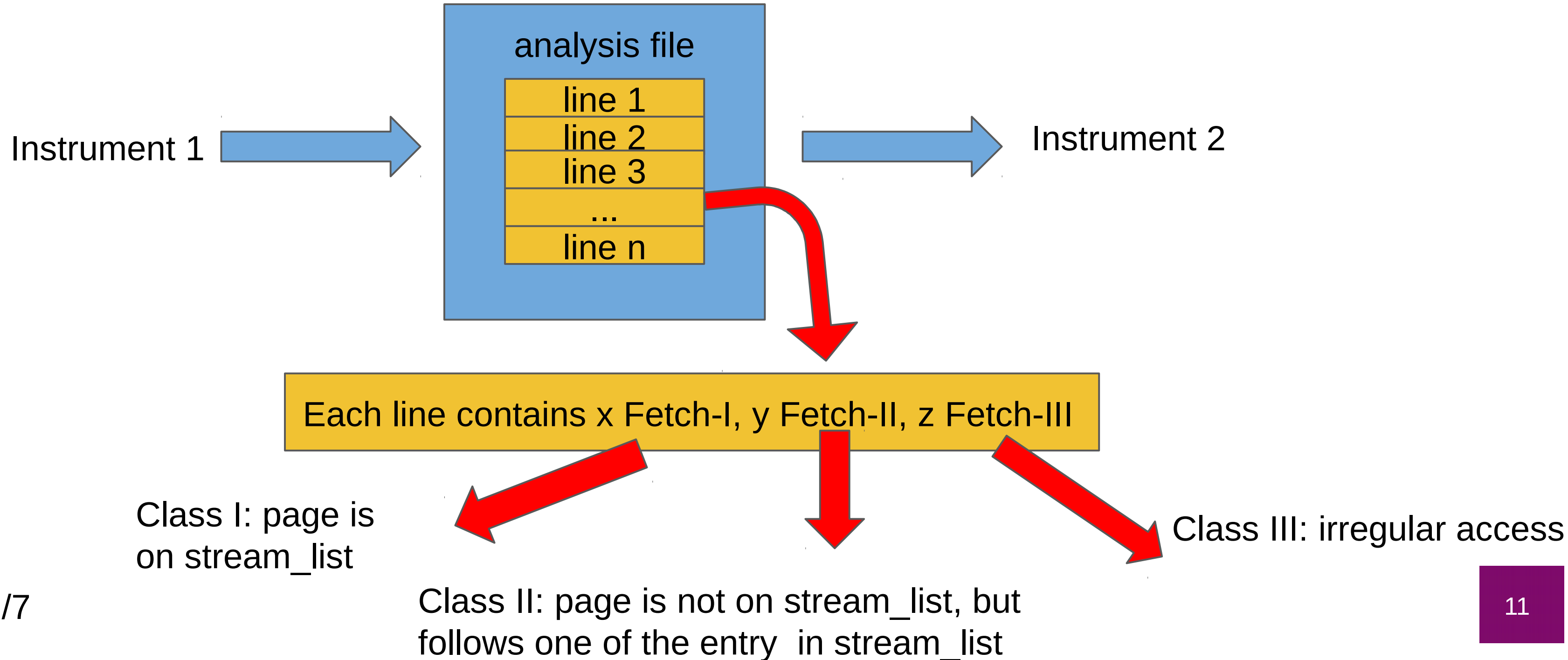
Profile-Guided Program Instrumention





- During analysis phase:
  - Mark all the address accessed
  - Mark all locations of instrumentation

How To Integrate DFP And SIP In An Application?



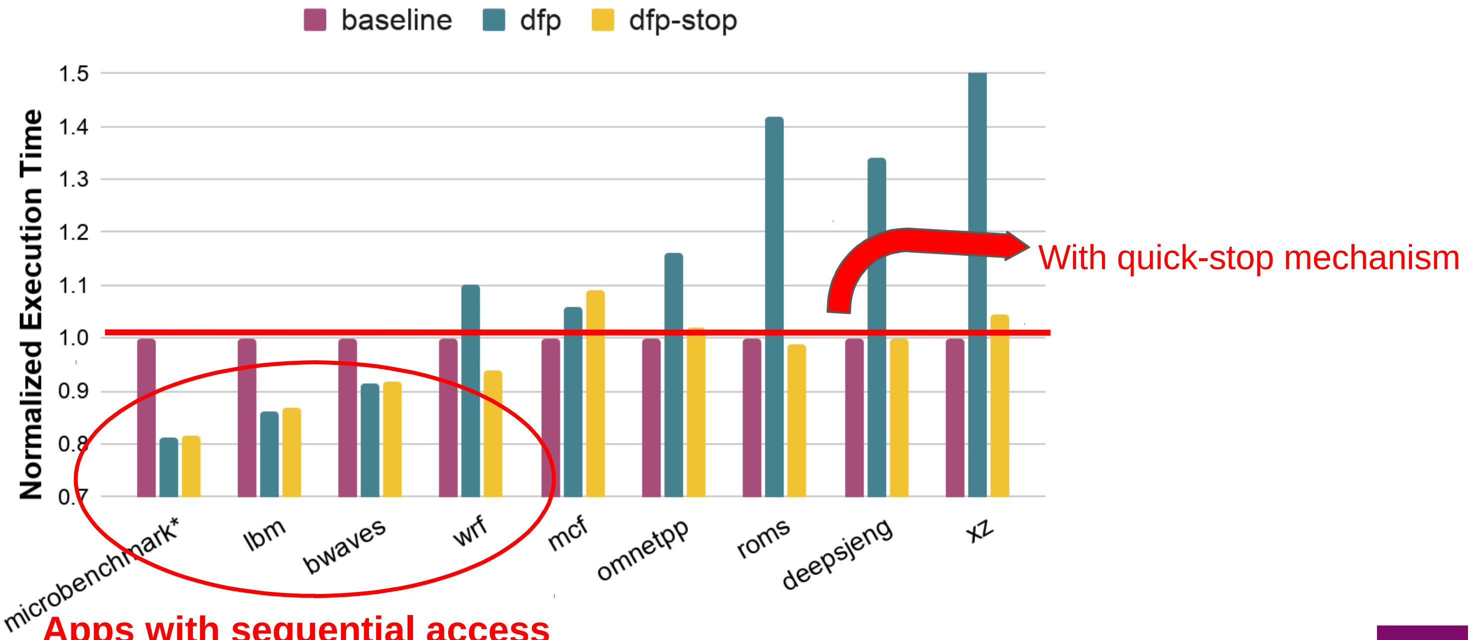
Each line contains x Fetch-I, y Fetch-II, z Fetch-III

Class I: page is on stream_list

Class II: page is not on stream_list, but follows one of the entry in stream_list

Class III: irregular access

DFP Performance on SPEC2017



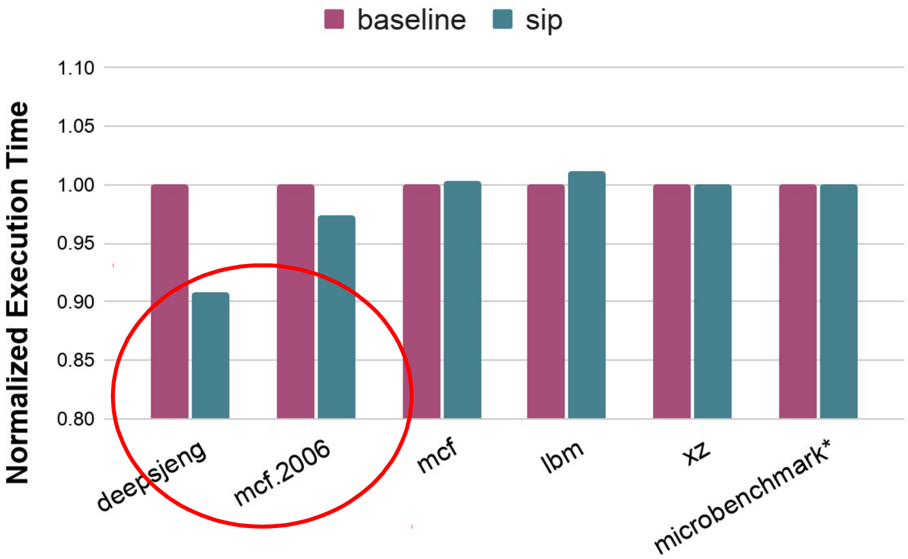
With quick-stop mechanism

Apps with sequential access patterns

*microbenchmark: 1GB sequential access app

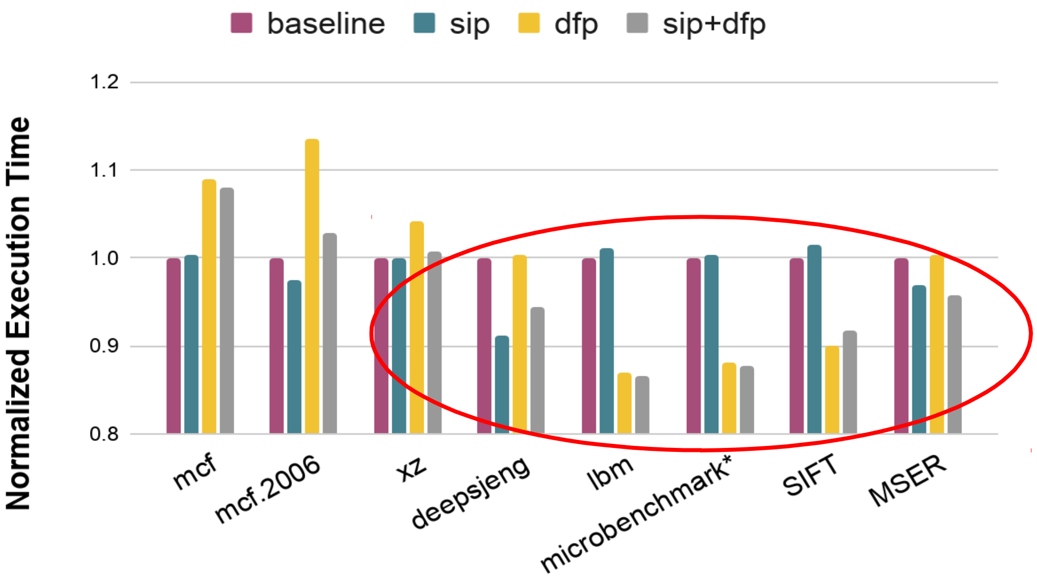
SIP Performance on SPEC2017

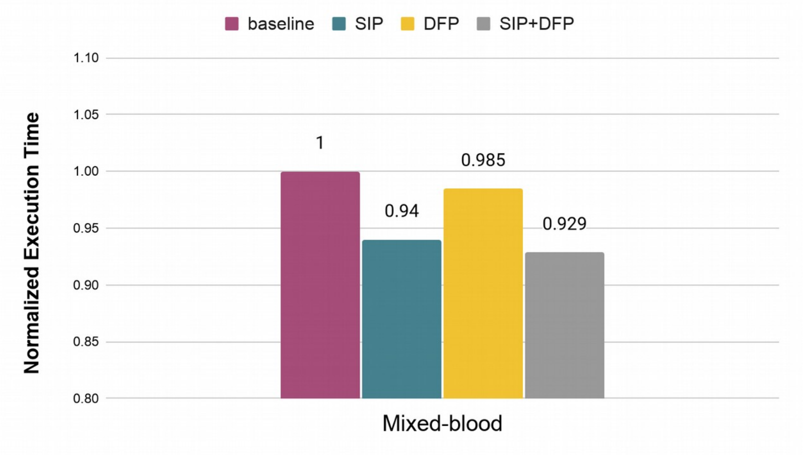
Benchmark	Instrumentation Points
mcf.2006	114
mcf	99
xz	46
deepsjeng	35
lbm	0
microbenchmark	0



*microbenchmark: 1GB sequential access app

Performance Using SIP + DFP





*microbenchmark: 1GB sequential access app

Conclusion

- Intel SGX offers **security** but also **overhead**s caused by **page faults**.
- We propose two page-preloading mechanisms **DFP** and **SIP** to improve **sequential** and **random** memory accesses in applications.
- Evaluation on SPEC2017, some real-world applications and a micro-benchmark program shows these two preloading mechanisms achieve an average of **11.4%** and **7.0%** performance improvement, respectively

Thanks !
Q&A