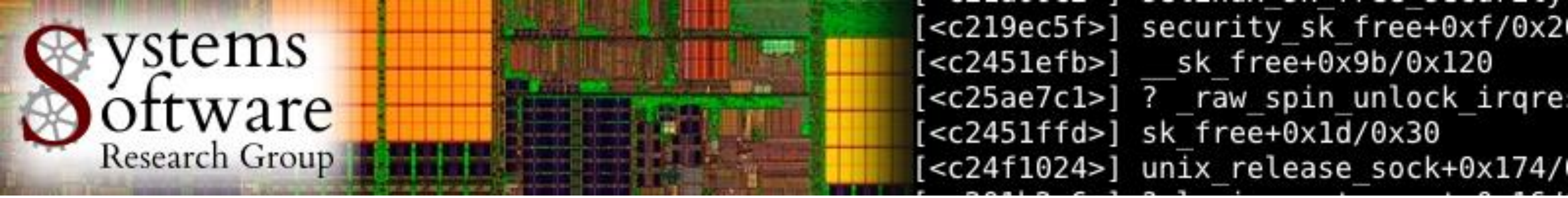


```
[<c219ec5f>] security_sk_free+0xf/0x2
[<c2451efb>] __sk_free+0x9b/0x120
[<c25ae7c1>] ? _raw_spin_unlock_irqre
[<c2451ffd>] sk_free+0x1d/0x30
[<c24f1024>] unix_release_sock+0x174/
```

Popcorn Linux: System Software for Emerging Heterogeneous Platforms

Rob Lyerly (rlyerly@vt.edu)

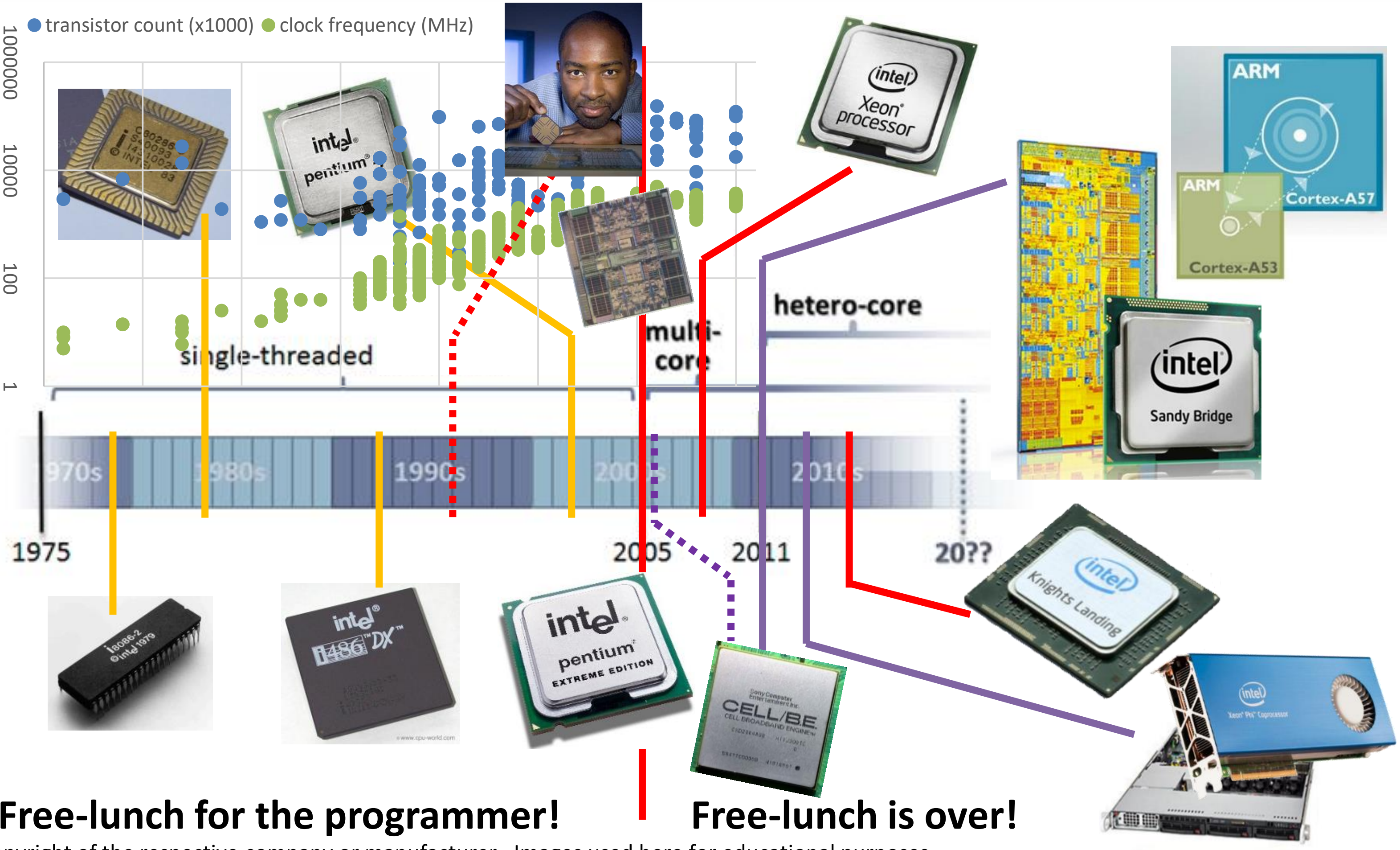
Systems Software Research Group at Virginia Tech

ssrg.ece.vt.edu

ece The BRADLEY DEPARTMENT of ELECTRICAL and COMPUTER ENGINEERING

VT VIRGINIA TECH™

Introduction



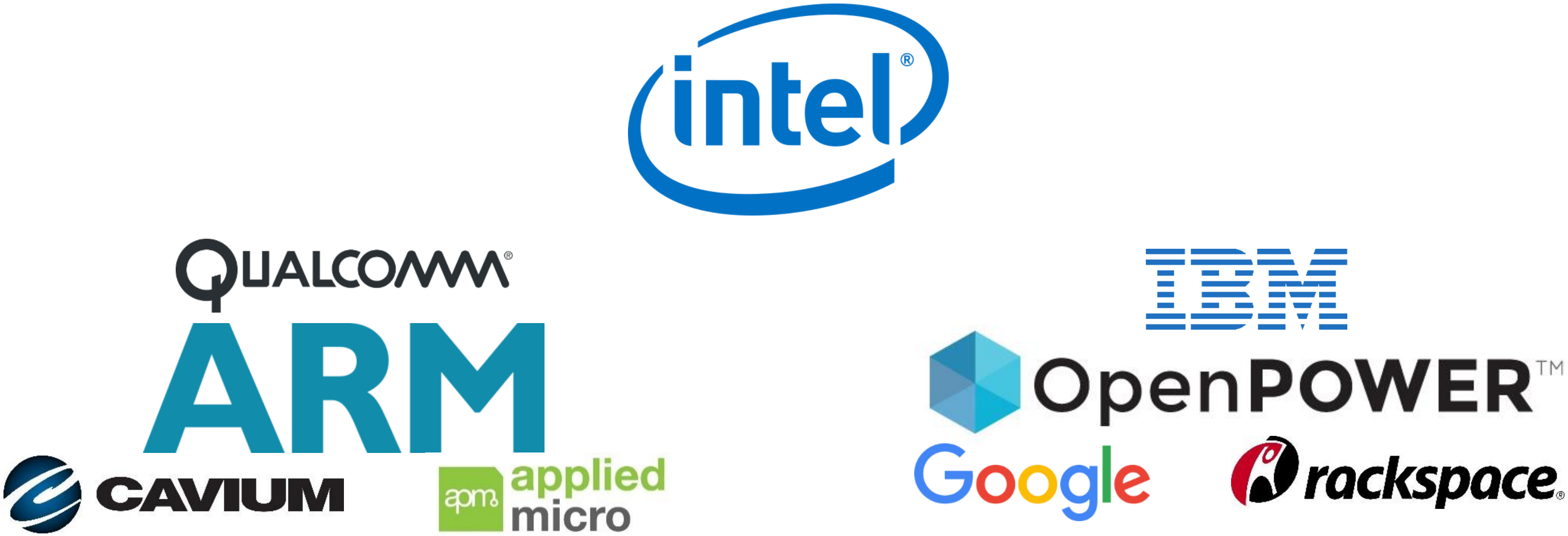
Free-lunch for the programmer!

Free-lunch is over!

Each image is copyright of the respective company or manufacturer. Images used here for educational purposes.

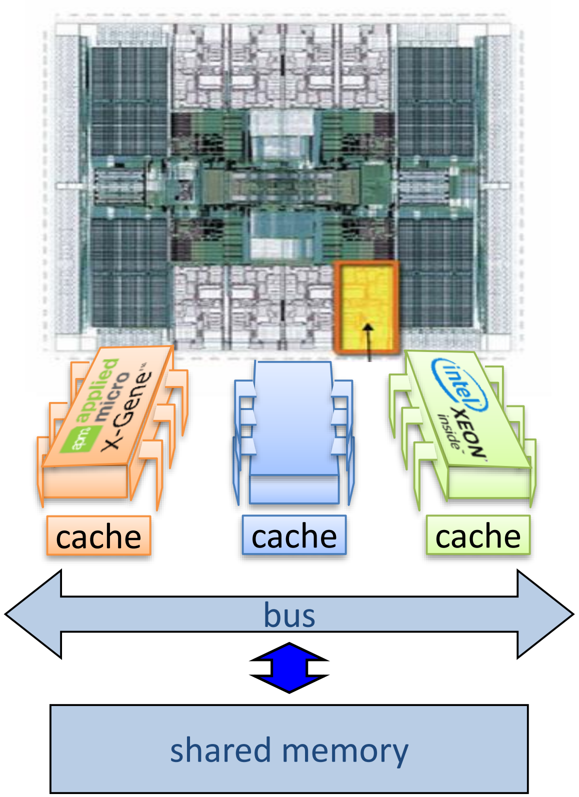
Introduction

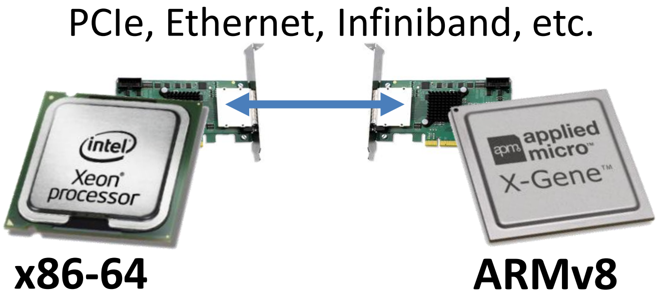
- Not only heterogeneous at the chip level – datacenters incorporating heterogeneous ISAs



Introduction

- On the horizon – fully general purpose, OS-capable, heterogeneous-ISA chip multiprocessors & rack-scale systems





Why Heterogeneous ISAs?

- Rack-level – vastly different performance/efficiency designs

Intel Xeon E5-1650v2

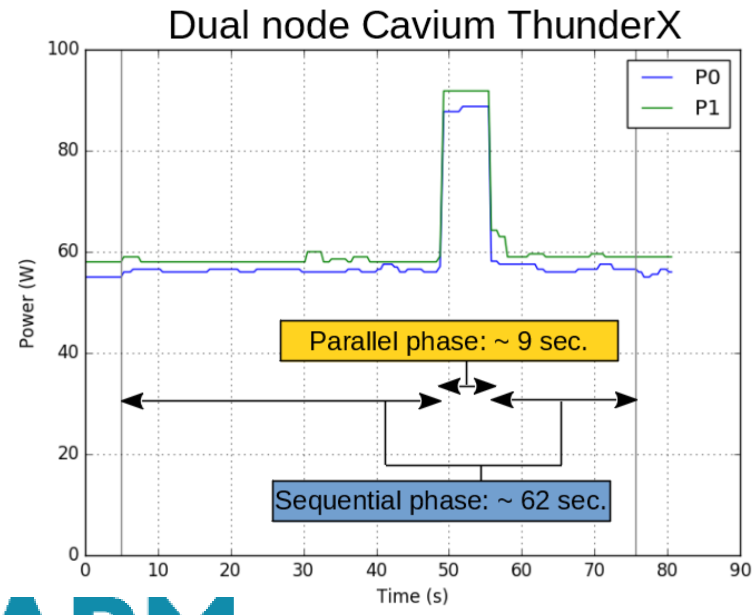
- x86-64, 64-bit
- 6 cores/12 threads per socket
- 3.5GHz base, 3.9GHz turbo boost
- L1: 32KB I$, 32KB D$ per core
- L2: 256KB per-core
- L3: 12MB (shared)
- 130W TDP

Cavium ThunderX

- ARMv8, 64-bit
- 48 cores per socket
- 2GHz per core
- L1: 78KB I$, 32KB D$ per core
- L2: 16MB (shared)
- 120W TDP

Why Heterogeneous ISAs?

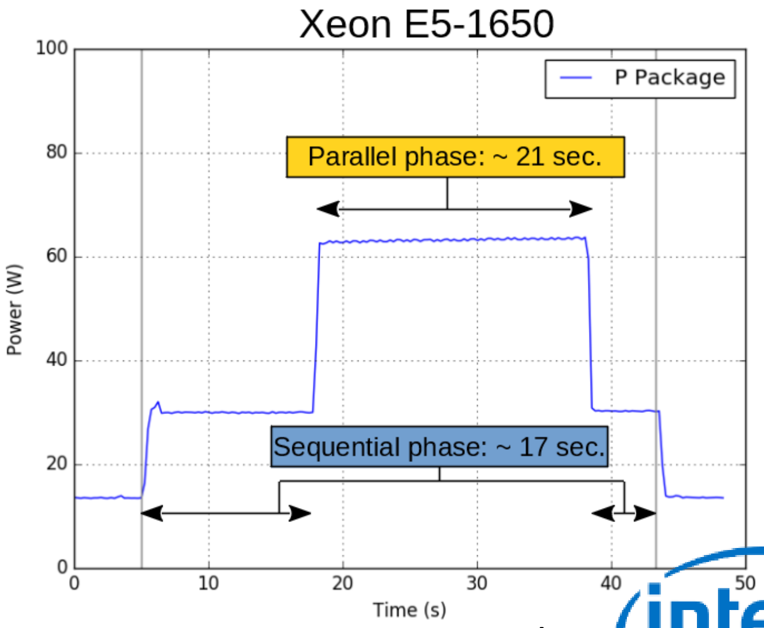
- Smart consolidation, load balancing for performance & energy gains
- ISA/Machine affinity:
  - **Program phases** illustration: PARSEC *blackscholes* native input



Dual node Cavium ThunderX

Parallel phase: ~ 9 sec.

Sequential phase: ~ 62 sec.



Xeon E5-1650

Parallel phase: ~ 21 sec.

Sequential phase: ~ 17 sec.

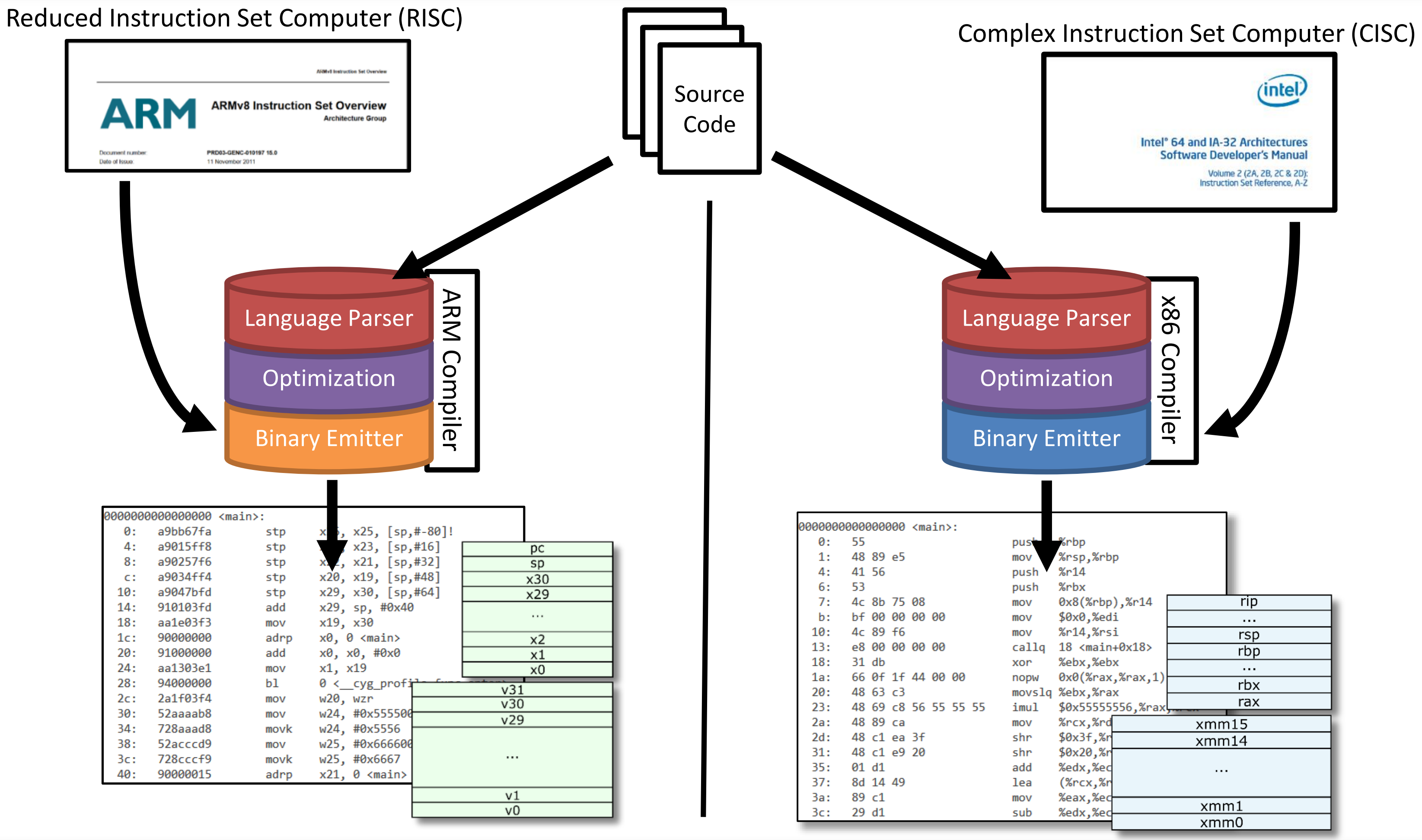
ARM 96 cores @2Ghz

6 cores @3.2Ghz intel

Systems Software Research Group

VIRGINIA TECH

Why Heterogeneous ISAs?



Reduced Instruction Set Computer (RISC)

ARMv8 Instruction Set Overview

ARM

ARMv8 Instruction Set Overview
Architecture Group

Document number: PRD03-GENC-010197 15.0
Date of Issue: 11 November 2011

Complex Instruction Set Computer (CISC)

intel

Intel® 64 and IA-32 Architectures
Software Developer's Manual

Volume 2 (2A, 2B, 2C & 2D):
Instruction Set Reference, A-Z

```
0000000000000000 <main>:
   0:   a9bb67fa        stp     x[illegible]5, x25, [sp,#-80]!
   4:   a9015ff8        stp     [illegible] x23, [sp,#16]
   8:   a90257f6        stp     x[illegible], x21, [sp,#32]
   c:   a9034ff4        stp     x20, x19, [sp,#48]
  10:   a9047bfd        stp     x29, x30, [sp,#64]
  14:   910103fd        add     x29, sp, #0x40
  18:   aa1e03f3        mov     x19, x30
  1c:   90000000        adrp    x0, 0 <main>
  20:   91000000        add     x0, x0, #0x0
  24:   aa1303e1        mov     x1, x19
  28:   94000000        bl      0 <__cyg_profi
  2c:   2a1f03f4        mov     w20, wzr
  30:   52aaaab8        mov     w24, #0x555500
  34:   728aaad8        movk    w24, #0x5556
  38:   52acccd9        mov     w25, #0x666600
  3c:   728cccf9        movk    w25, #0x6667
  40:   90000015        adrp    x21, 0 <main>
```

```
0000000000000000 <main>:
   0:   55                      pus     %rbp
   1:   48 89 e5                mov     %rsp,%rbp
   4:   41 56                   push    %r14
   6:   53                      push    %rbx
   7:   4c 8b 75 08             mov     0x8(%rbp),%r14
   b:   bf 00 00 00 00          mov     $0x0,%edi
  10:   4c 89 f6                mov     %r14,%rsi
  13:   e8 00 00 00 00          callq   18 <main+0x18>
  18:   31 db                   xor     %ebx,%ebx
  1a:   66 0f 1f 44 00 00       nopw    0x0(%rax,%rax,1)
  20:   48 63 c3                movslq  %ebx,%rax
  23:   48 69 c8 56 55 55 55    imul    $0x55555556,%rax
  2a:   48 89 ca                mov     %rcx,%rd
  2d:   48 c1 ea 3f             shr     $0x3f,%r
  31:   48 c1 e9 20             shr     $0x20,%r
  35:   01 d1                   add     %edx,%ec
  37:   8d 14 49                lea     (%rcx,%r
  3a:   89 c1                   mov     %eax,%ec
  3c:   29 d1                   sub     %edx,%ec
```

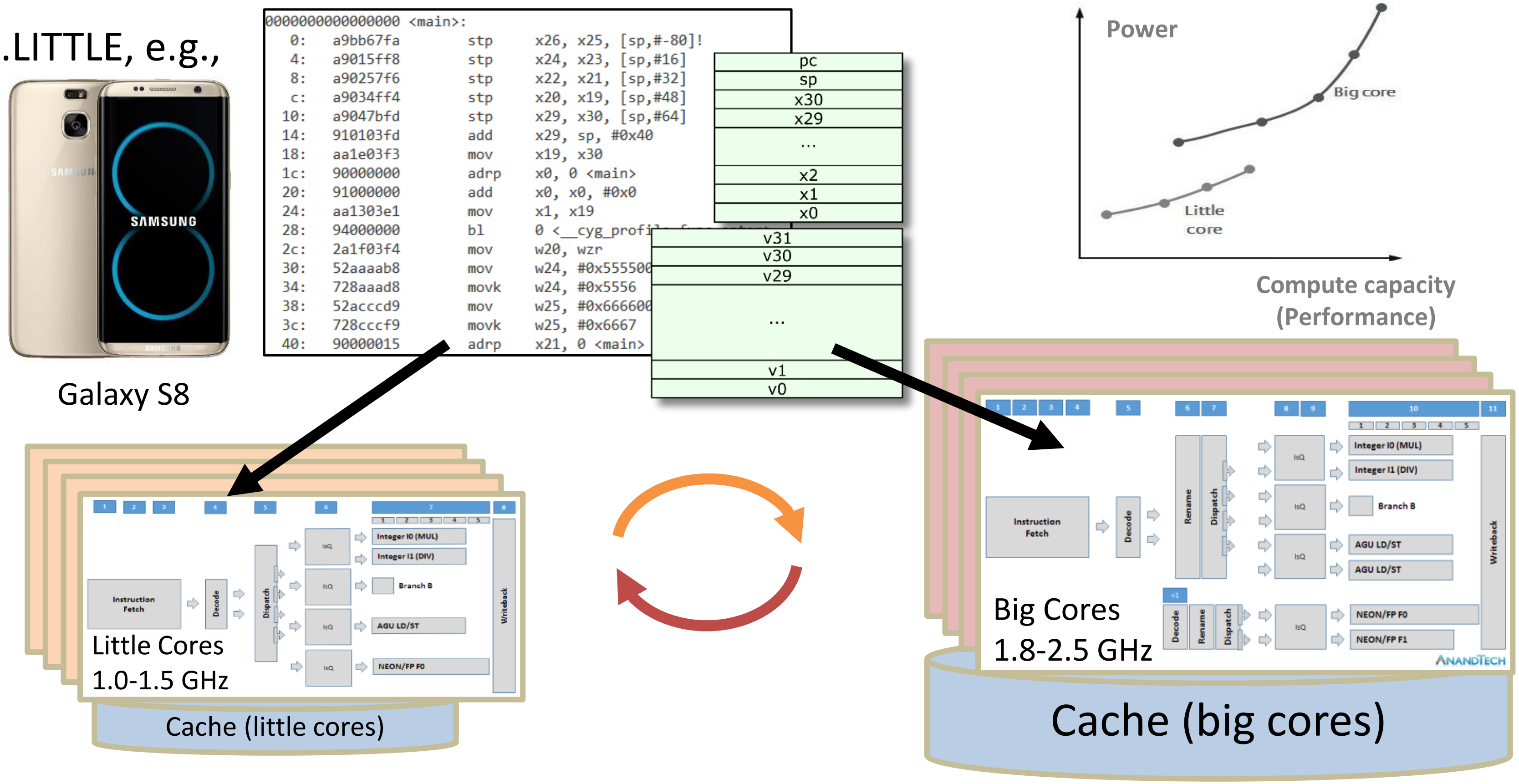
Why Heterogeneous ISAs?

- Chip-level – micro-architectural heterogeneity is already here!

ARM big.LITTLE, e.g.,

iPhone 8/X

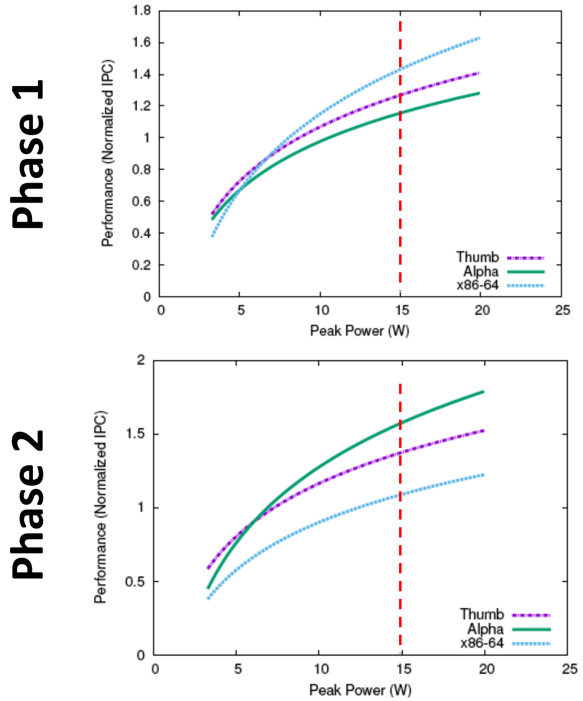
Galaxy S8



```
0000000000000000 <main>:
   0:   a9bb67fa        stp     x26, x25, [sp,#-80]!
   4:   a9015ff8        stp     x24, x23, [sp,#16]
   8:   a90257f6        stp     x22, x21, [sp,#32]
   c:   a9034ff4        stp     x20, x19, [sp,#48]
  10:   a9047bfd        stp     x29, x30, [sp,#64]
  14:   910103fd        add     x29, sp, #0x40
  18:   aa1e03f3        mov     x19, x30
  1c:   90000000        adrp    x0, 0 <main>
  20:   91000000        add     x0, x0, #0x0
  24:   aa1303e1        mov     x1, x19
  28:   94000000        bl      0 <__cyg_profi
  2c:   2a1f03f4        mov     w20, wzr
  30:   52aaaab8        mov     w24, #0x555500
  34:   728aaad8        movk    w24, #0x5556
  38:   52acccd9        mov     w25, #0x666600
  3c:   728cccf9        movk    w25, #0x6667
  40:   90000015        adrp    x21, 0 <main>
```

Why Heterogeneous ISAs?

- The case for heterogeneous-ISA multicores



Performance of bzip2's two different phases for different peak power budgets

🟦 Homogeneous

- Alpha

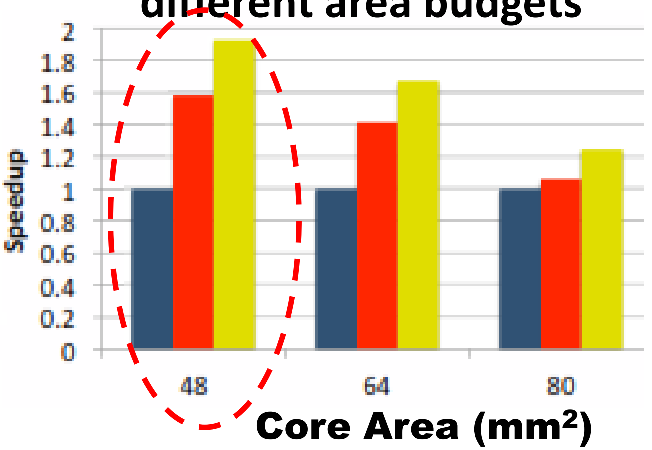
🟥 Single-ISA

- big Alpha
- medium Alpha
- little Alpha

🟨 Heterogeneous-ISA

- ARM's thumb
- x86_64
- Alpha

Single thread performance for different area budgets



Smaller the core area, greater # of cores can be placed on the silicon

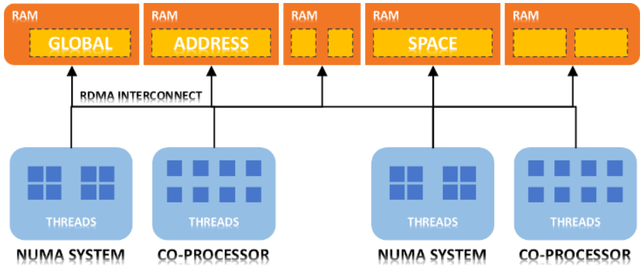
"Harnessing ISA Diversity: Design of a Heterogeneous-ISA Chip Multiprocessor," A. Venkat and D. M. Tullsen, ISCA 2014.

Heterogeneous-ISA Execution

- Big questions for programming heterogeneous-ISA systems
  1. What is the programming language/model?
  2. How do I abstract away ISA differences, e.g., code, data layout?
  3. How is memory accessed/shared across distinct memory regions?

THE C PROGRAMMING LANGUAGE

OPEN MPI



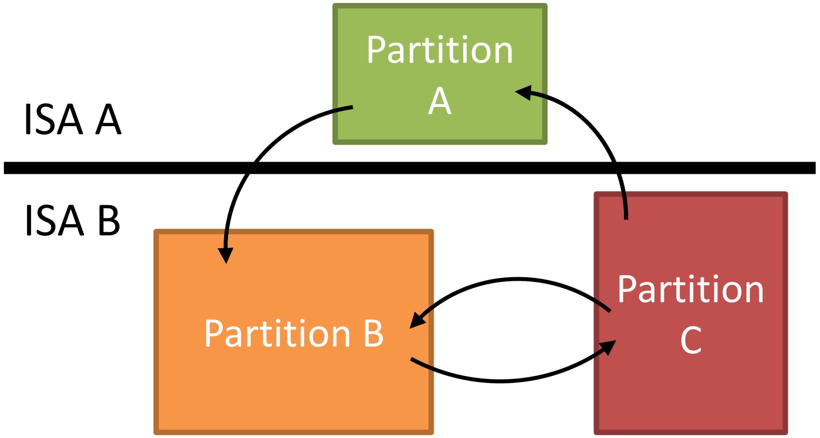
Heterogeneous-ISA Execution

Message Passing Interface (MPI)

+ **High performance**

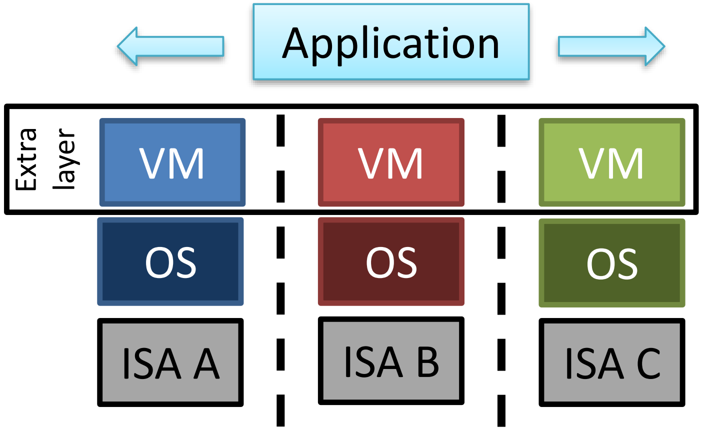
– **Complex code development/refactoring**

– **Hardcoded application partitions**



ISA Virtualization

- Managed languages, e.g., Java
  - – **Rewrite application from scratch**
  - – **Performance overheads**
- Dynamic binary translation, e.g., QEMU
  - + **Run unmodified binaries**
  - – **Order of magnitude slowdown**



Heterogeneous-ISA Execution

- Using **shared memory** gives sanity back to the developer!
  - Well understood programming model – POSIX shared memory (30 years), OpenMP (20 years)
  - One common memory region – no data marshaling!
  - Higher programmability versus offloading!

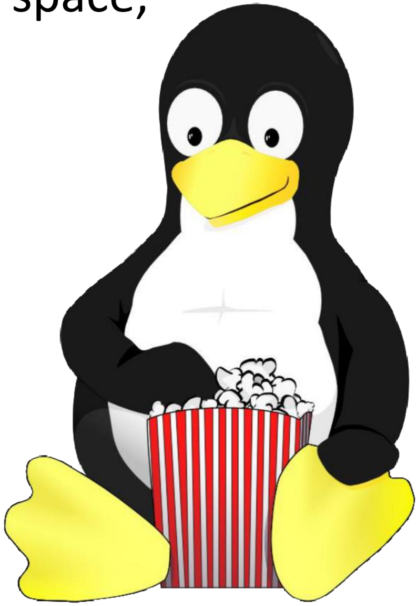
Benchmark	CG	EP	FT	IS	MG
OpenMP LOC	1150	297	1106	1108	1481
MPI modified	**98%**	**44%**	**98%**	**46%**	**97%**

OpenMP and MPI version of NASA NPB

  - High performance – no language or system VMs necessary!
  - Flexible platform-wide resource management!

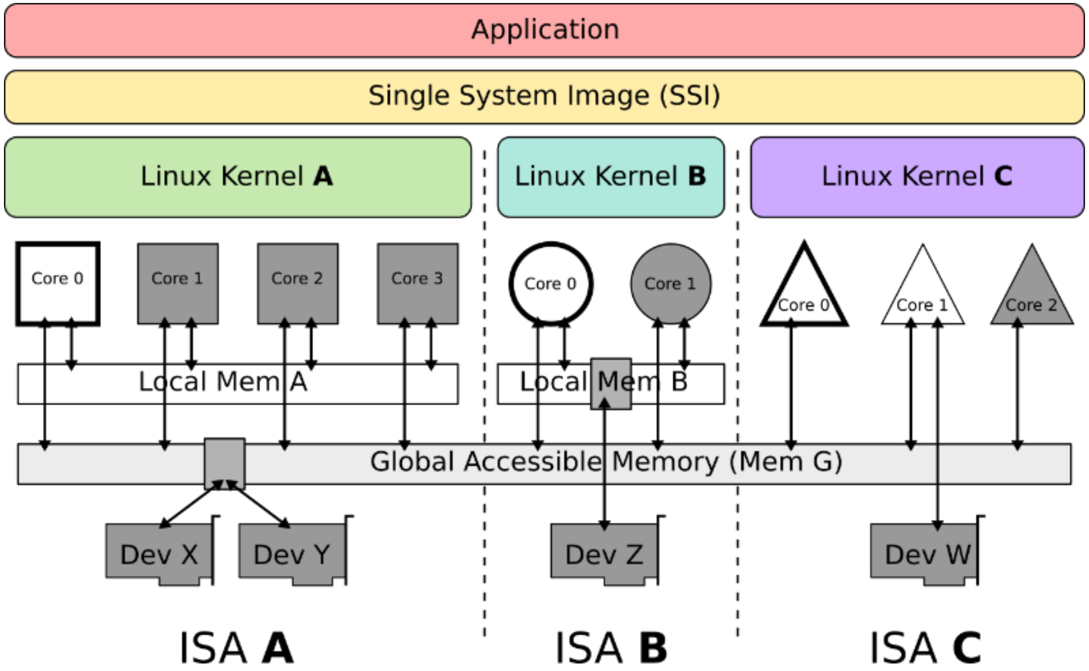
Popcorn Linux

- System software stack for **migrating** compiled applications between heterogeneous-ISA servers
  - **Replicated-kernel OS** for thread and data migration
  - **Compiler** for creating a mostly-common virtual address space, generating metadata about ISA-specific execution state
  - **Runtime** for transforming ISA-specific execution state
- Allow developers to write **shared memory compiled applications** and leverage heterogeneity
  - Legacy code works too!



Popcorn Linux: Operating System

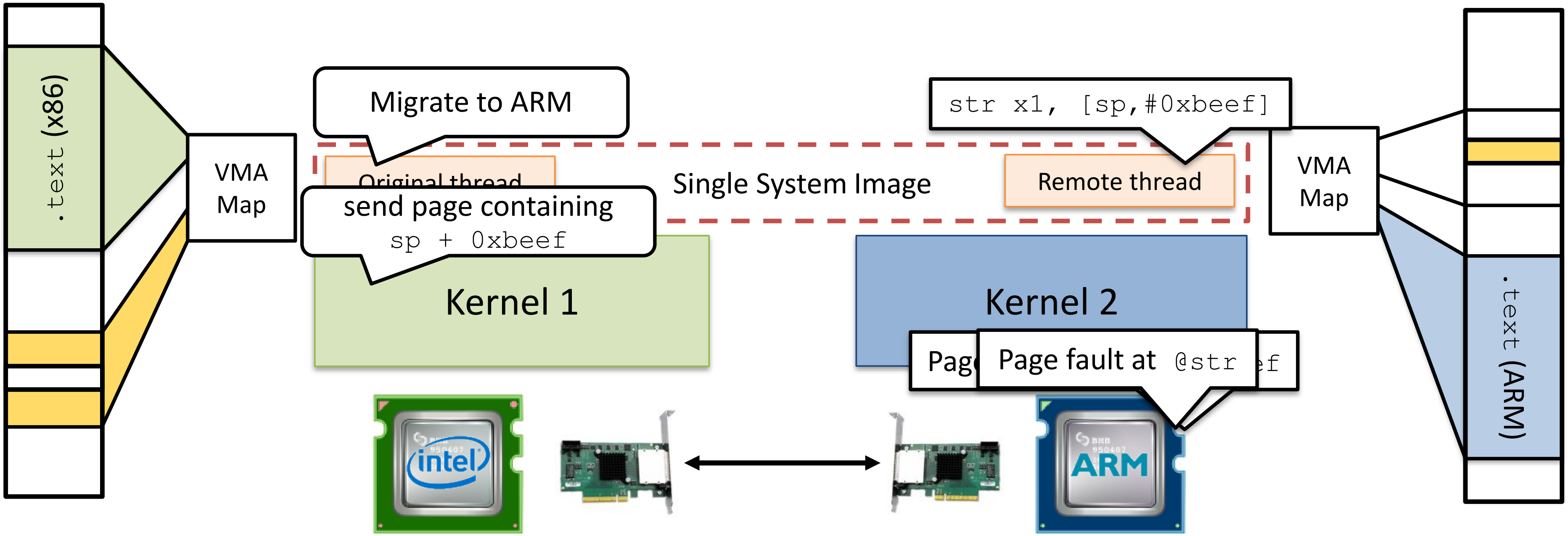
- Multiple kernels provides ***single system image*** (SSI) allowing threads to migrate freely between nodes



Popcorn Linux: Operating System

- Thread migration & heterogeneous continuations
  - Threads invoke migration via syscall
  - Kernels cooperate to migrate user-space thread contexts between ISAs
  - Kernel maps user-space PC, SP and FBP registers between ISAs
- On-demand page migration
  - Migrate memory pages between kernels when accessed by application
    - Intercept & redirect the page fault handler
    - Kept coherent using MSI-like protocol
  - Memory region aliasing for ISA-specific sections (e.g., `.text`)

Popcorn Linux: Operating System



.text (x86)

VMA Map

Migrate to ARM

Original thread

send page containing
`sp + 0xbeef`

Single System Image

`str x1, [sp,#0xbeef]`

Remote thread

VMA Map

Kernel 1

Kernel 2

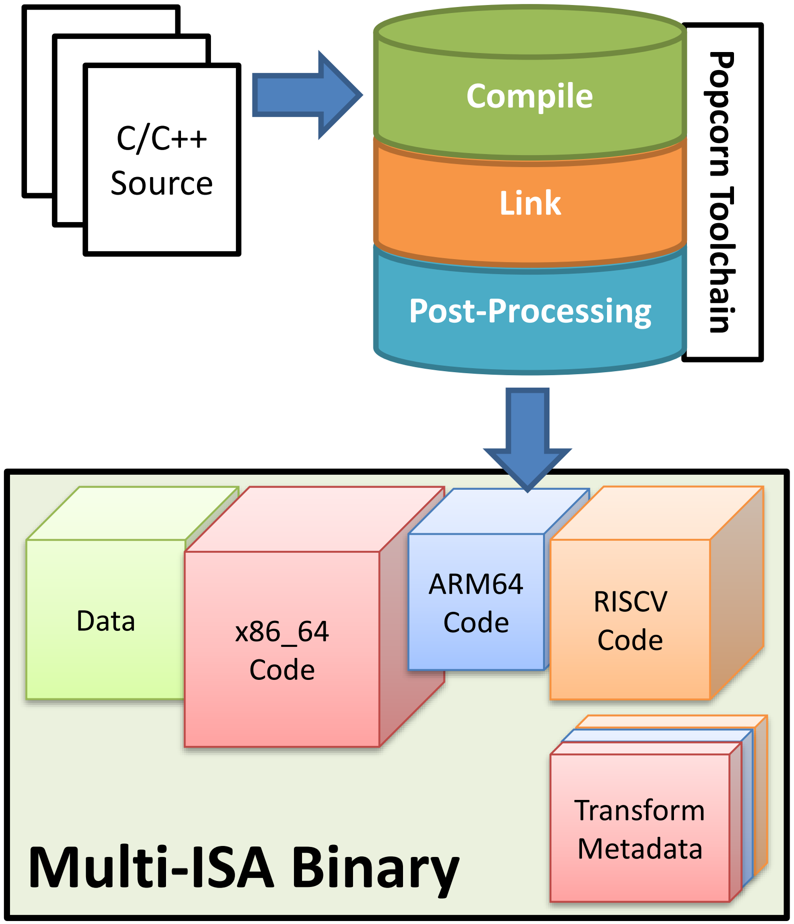
Pag

Page fault at `@str` ef

.text (ARM)

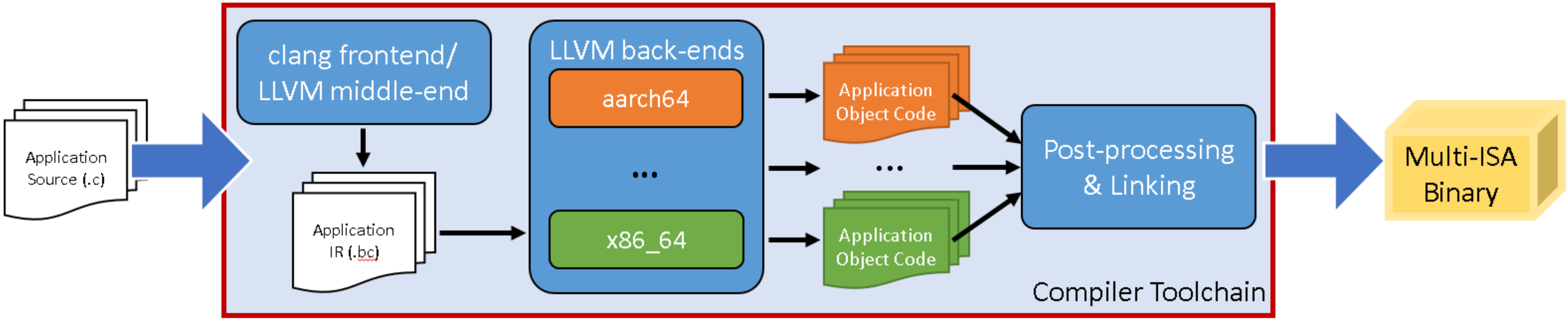
Popcorn Linux: Compiler

- Compiler toolchain builds ***multi-ISA binaries***
  - Create mostly-common virtual address space (data, code, heap)
    - Pointers are valid across all ISAs
  - Dynamically transform thread execution state (stack, registers) between ISA-specific formats at migration time
  - Instrument generated code with migration points



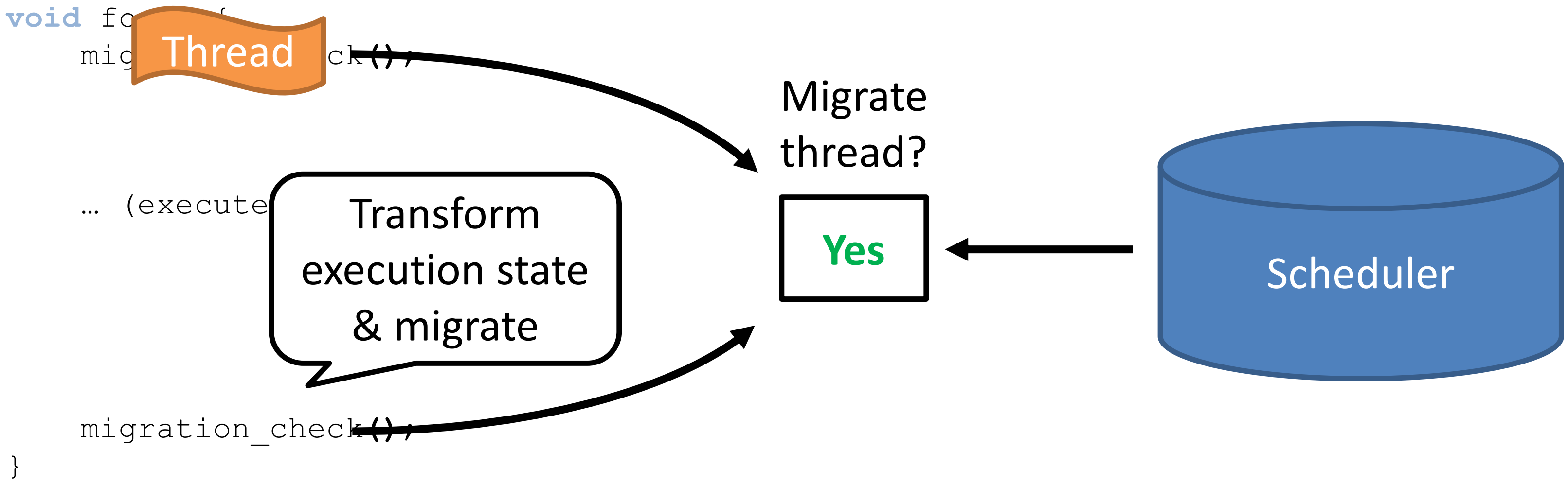
Popcorn Linux: Compiler

- Built on top of clang/LLVM
  - clang/LLVM 3.7.1, GNU gold 2.27, musl-libc 1.1.18
  - Custom address space alignment, post-processing tools
  - State transformation/migration libraries



Popcorn Linux: Compiler

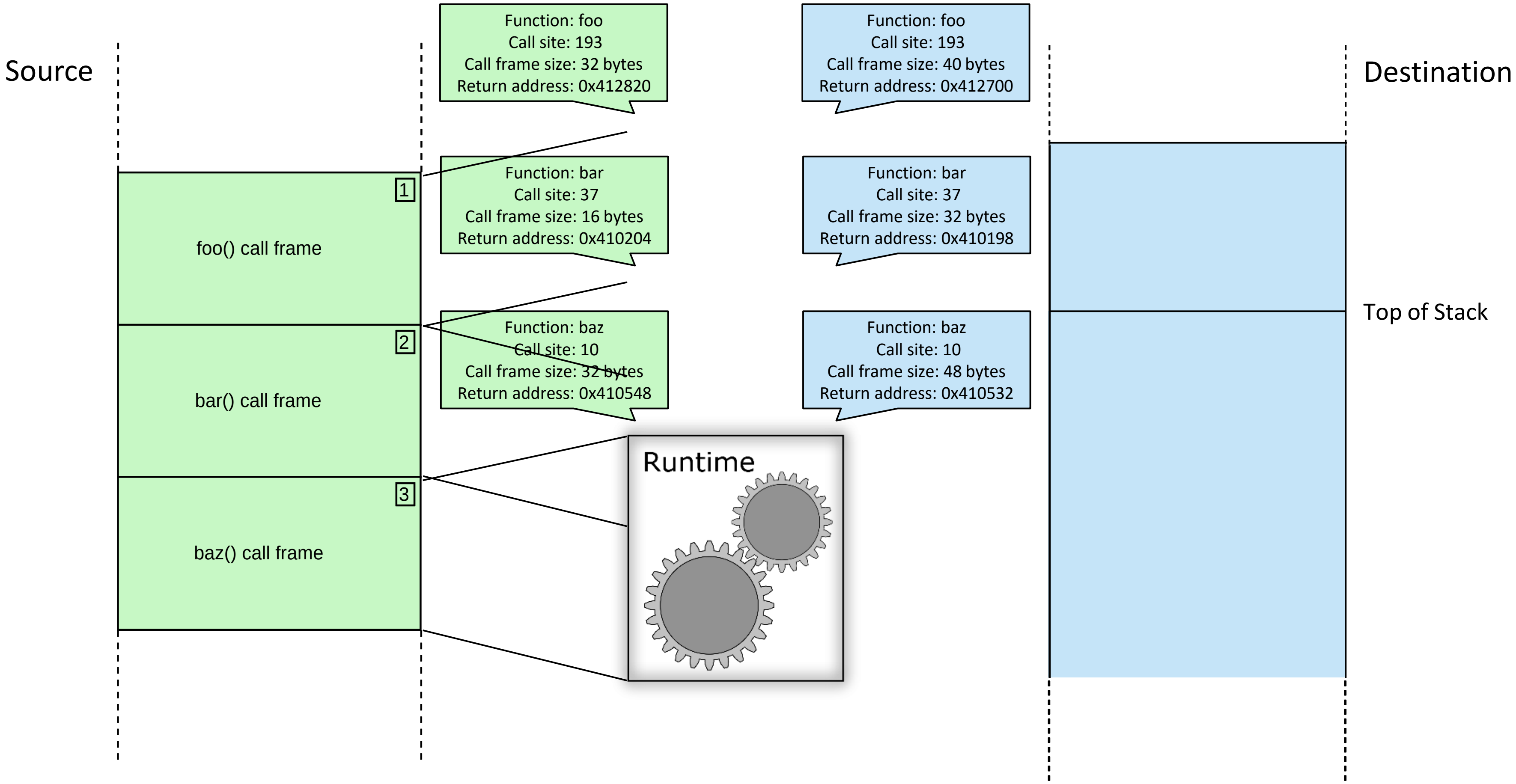
- Insert **migration points** into code
  - Can only transform stack at *equivalence points*
    - Direct mapping of execution state between ISA-specific formats
  - Scheduler cannot migrate threads at arbitrary points, must signal threads to initiate migration process



Popcorn Linux: Runtime

- Transform registers & stack between ISA-specific formats
- Runtime transforms state before migration
  - Attaches to a thread's registers/stack
  - Reads compiler metadata describing function activation layouts
  - Rewrites stack in its entirety from source to destination ISA format
- After transformation, runtime invokes migration
  - Passes destination ISA's register state and stack to OS's thread migration service

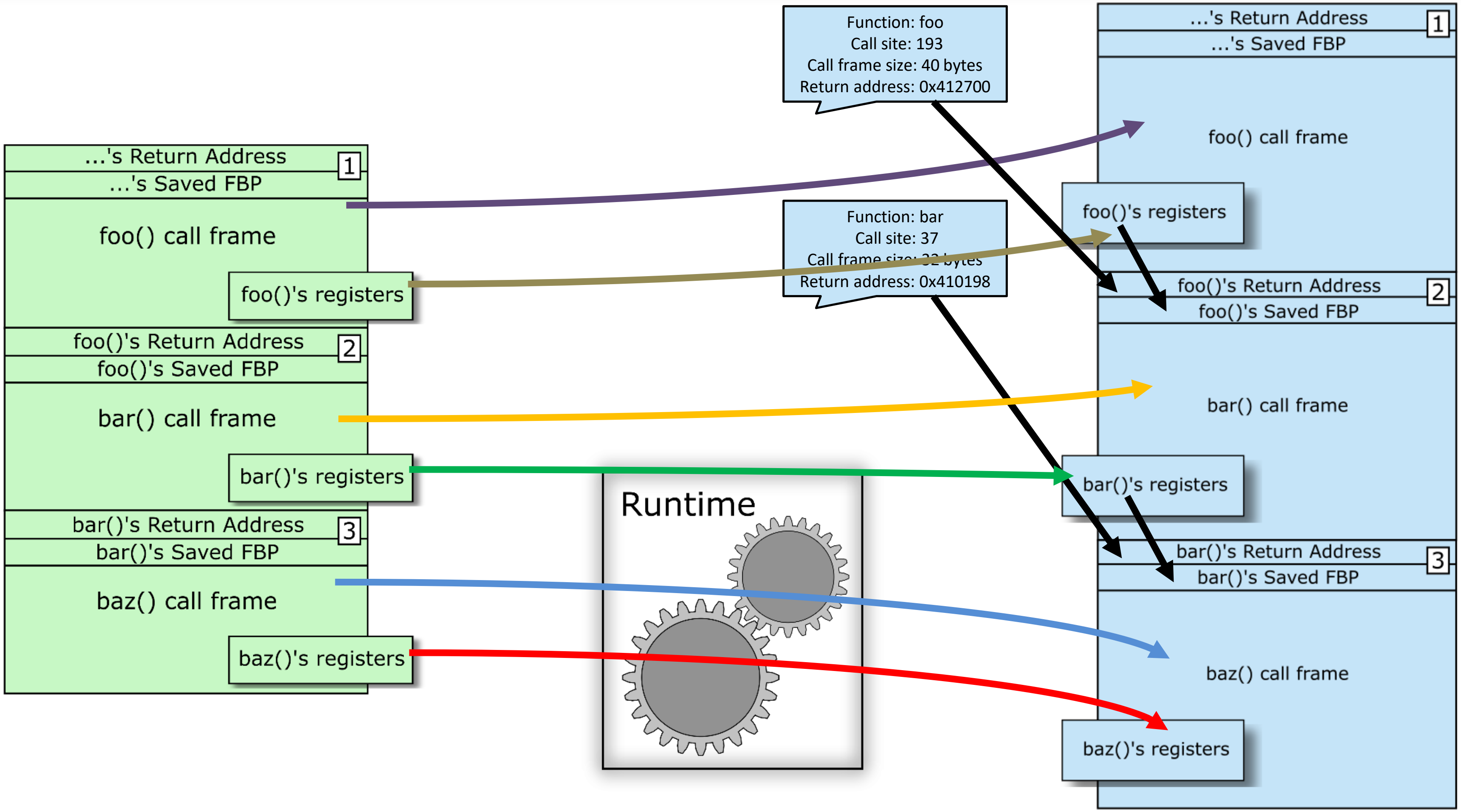
Popcorn Linux: Runtime



Systems Software Research Group

VIRGINIA TECH™

Popcorn Linux: Runtime



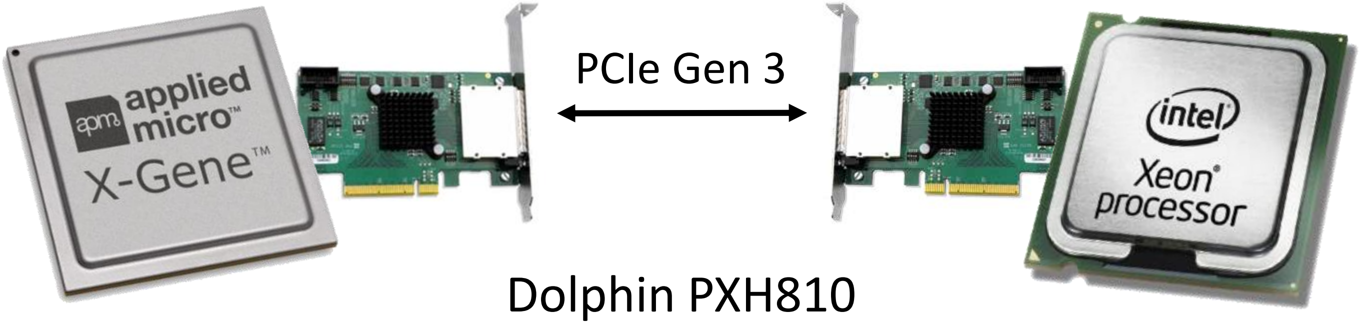
Function: foo
Call site: 193
Call frame size: 40 bytes
Return address: 0x412700

Function: bar
Call site: 37
Call frame size: 32 bytes
Return address: 0x410198

Evaluation

- APM X-Gene 1
  - 8 cores @ 2.4GHz
  - 8MB LLC, 32GB RAM
  - 40nm process, 50W TDP
    - Measured via on-board sensor
    - Estimated power consumption scaled to 22nm using McPAT

- Intel Xeon E5-1650v2
  - 6 cores @ 3.5GHz (3.9GHz turbo)
    - Hyperthreading disabled
  - 12MB LLC, 16GB RAM
  - 22nm process, 130W TDP
    - Measured via RAPL



PCIe Gen 3

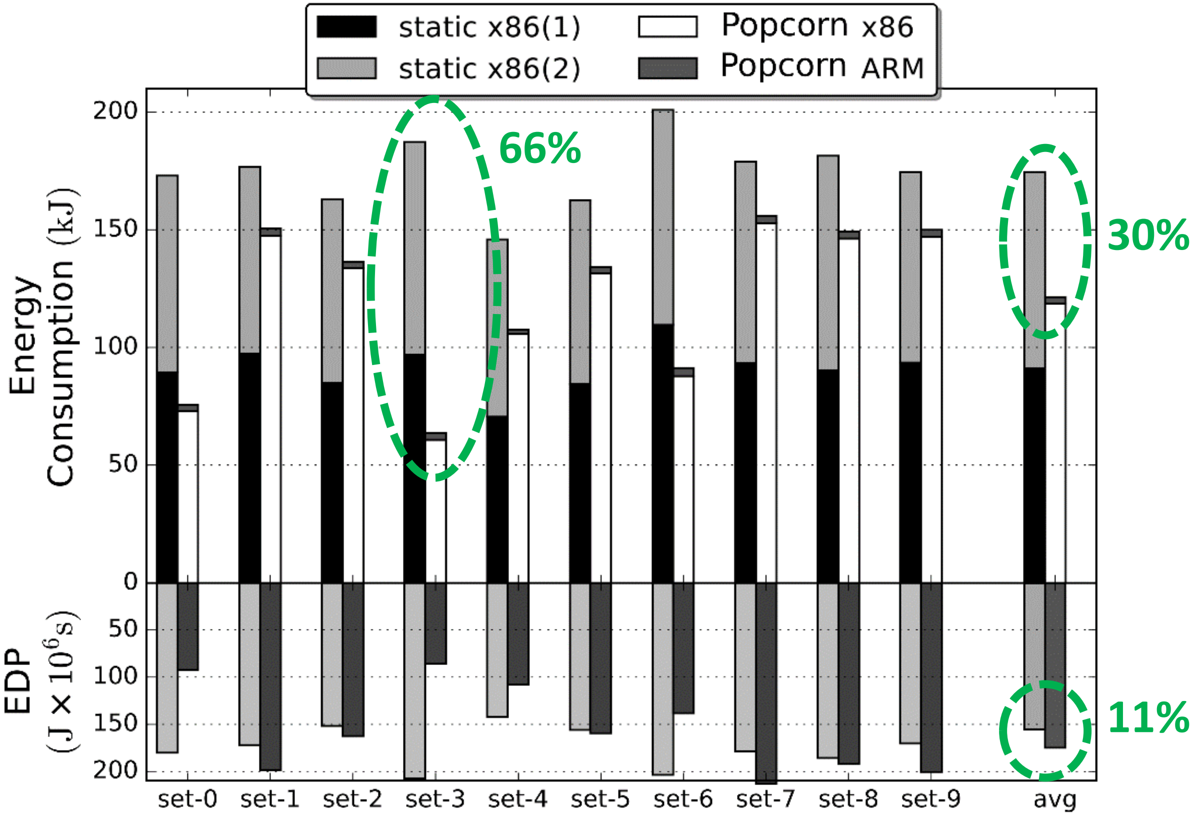
Dolphin PXH810

PCIe point-to-point connection, 64Gbps

Evaluation

- Load balance across architectures
  - Periodic workload – each set consists of 5 waves of up to 14 jobs
    - Uniformly sampled from NAS Parallel Benchmarks (NPB), class A, B & C
    - Waves arrive every 60-240 seconds
  - Comparison against 2 x Intel Xeon E5-1650v2 w/o migration

Results


static x86(1)
static x86(2)
Popcorn x86
Popcorn ARM
Energy Consumption (kJ)
0
50
100
150
200
EDP $(J \times 10^6 s)$
50
100
150
200
set-0
set-1
set-2
set-3
set-4
set-5
set-6
set-7
set-8
set-9
avg
66%
30%
11%

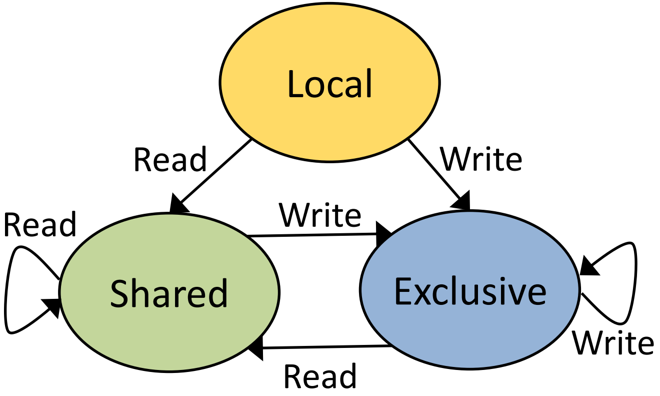
Summary

- Initial working prototype on 2 node ARM64 & x86-64
- Load balancing across high-performance/energy-efficient processors enabled energy savings versus homogeneous setup

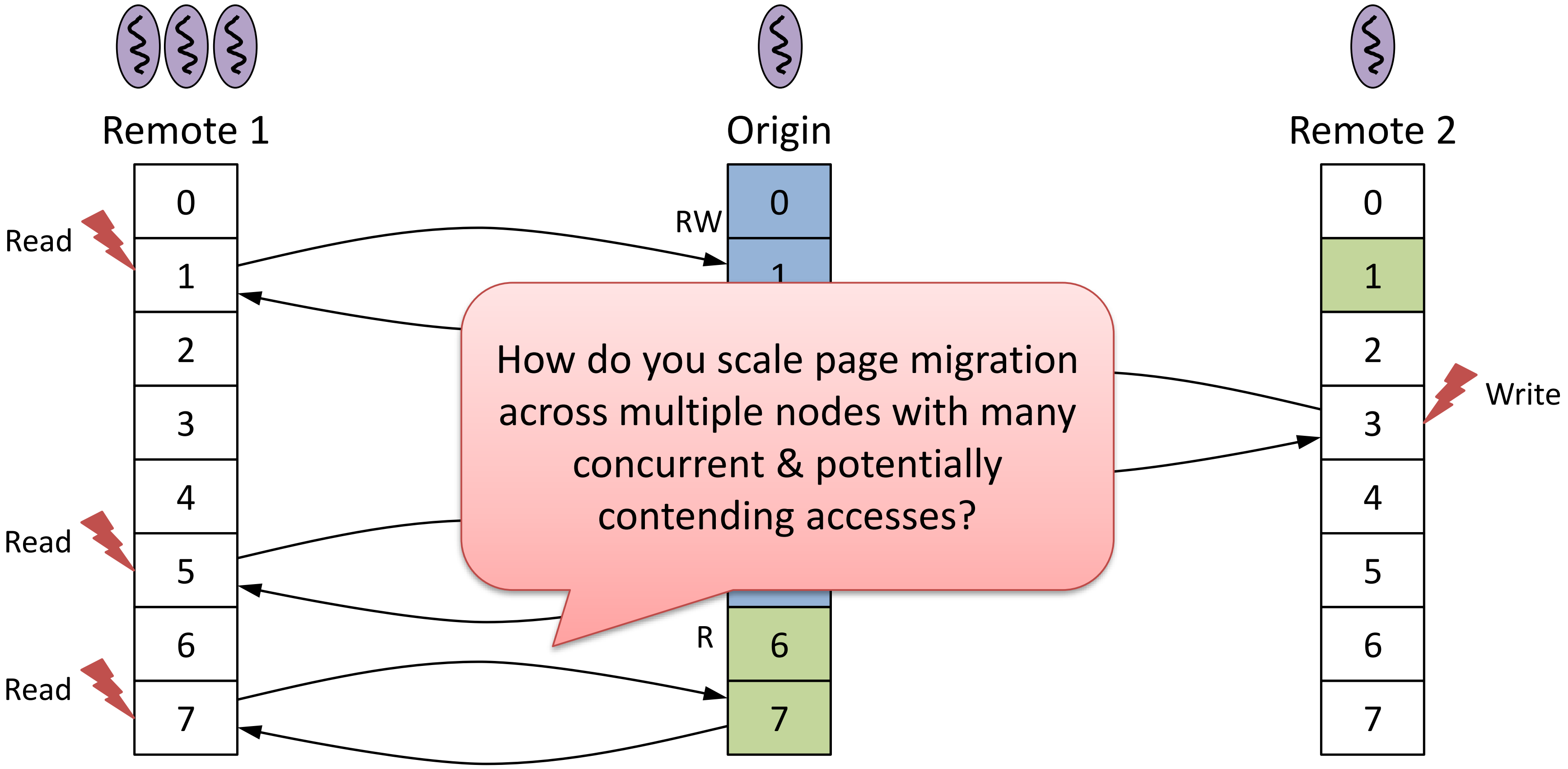
Ongoing Research: Rack-Scale

- Centralized page management for implementation feasibility
  - Track the ownership for each page at the origin
  - Remote threads request missing pages to the origin
  - The origin handles the requests accordingly

- Model the simple RMI cache coherent protocol
  - Read-Modify Invalidate
  - Read replicates a page
  - Write gets exclusive access to the page
    - Invalidates the page from other nodes



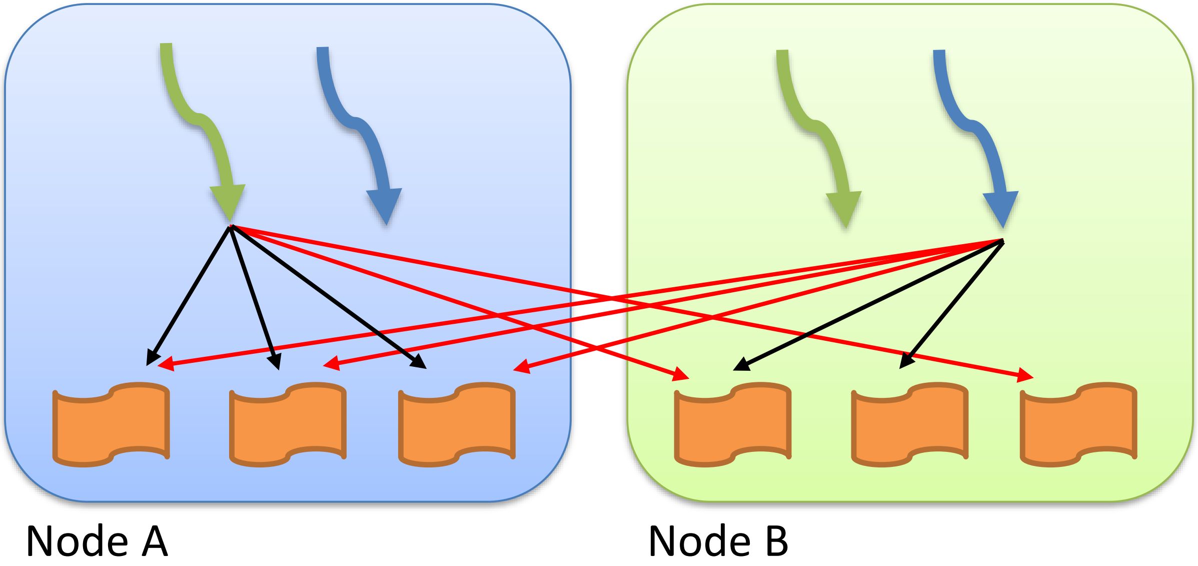
Ongoing Research: Rack-Scale



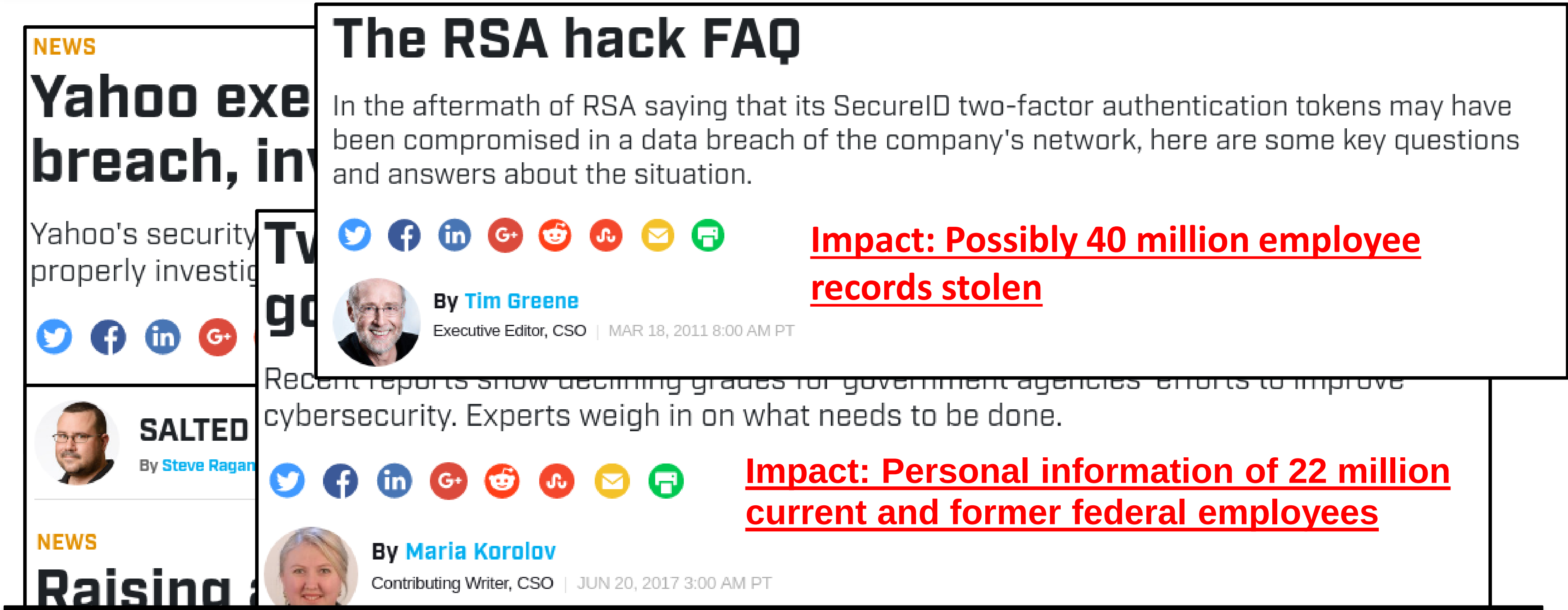
How do you scale page migration across multiple nodes with many concurrent & potentially contending accesses?

Ongoing Research: Rack-Scale

- Reduce number of page migrations
  - Co-locate threads with needed data – graph partitioning problem
  - Invert mechanism – migrate threads to data
  - Change work ↔ data mapping



New Project: Secure Popcorn



NEWS

Yahoo exe

breach, in

Yahoo's security

properly investig

SALTED

By Steve Ragan

NEWS

Raising

Tw

go

Recent reports show declining grades for government agencies' efforts to improve cybersecurity. Experts weigh in on what needs to be done.

By Maria Korolov

Contributing Writer, CSO | JUN 20, 2017 3:00 AM PT

Impact: Personal information of 22 million current and former federal employees

The RSA hack FAQ

In the aftermath of RSA saying that its SecureID two-factor authentication tokens may have been compromised in a data breach of the company's network, here are some key questions and answers about the situation.

By Tim Greene

Executive Editor, CSO | MAR 18, 2011 8:00 AM PT

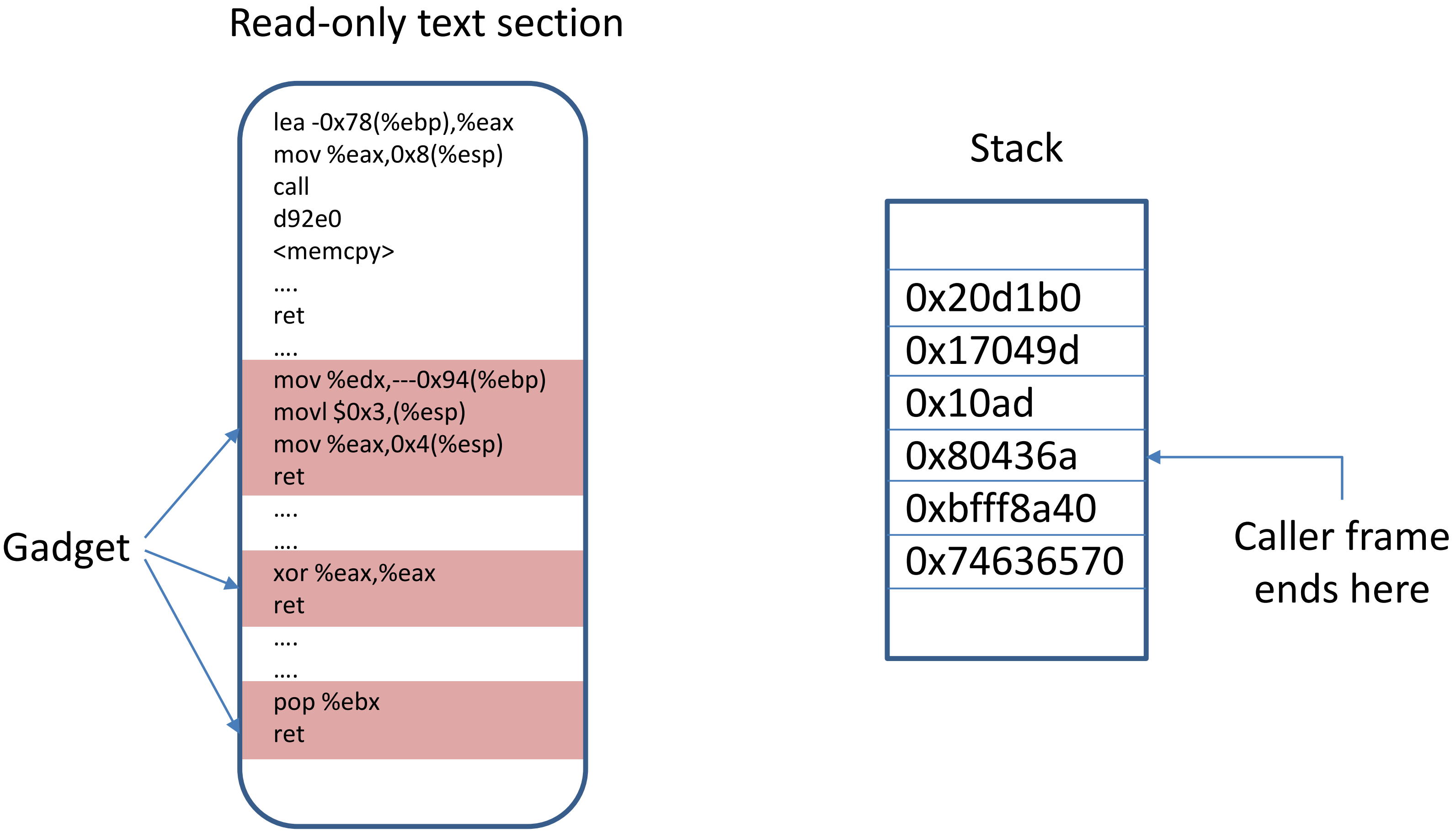
Impact: Possibly 40 million employee records stolen

90% of today's attacks utilize ROP

Systems Software Research Group

VIRGINIA TECH

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
ret
….
….
pop %ebx
ret
```

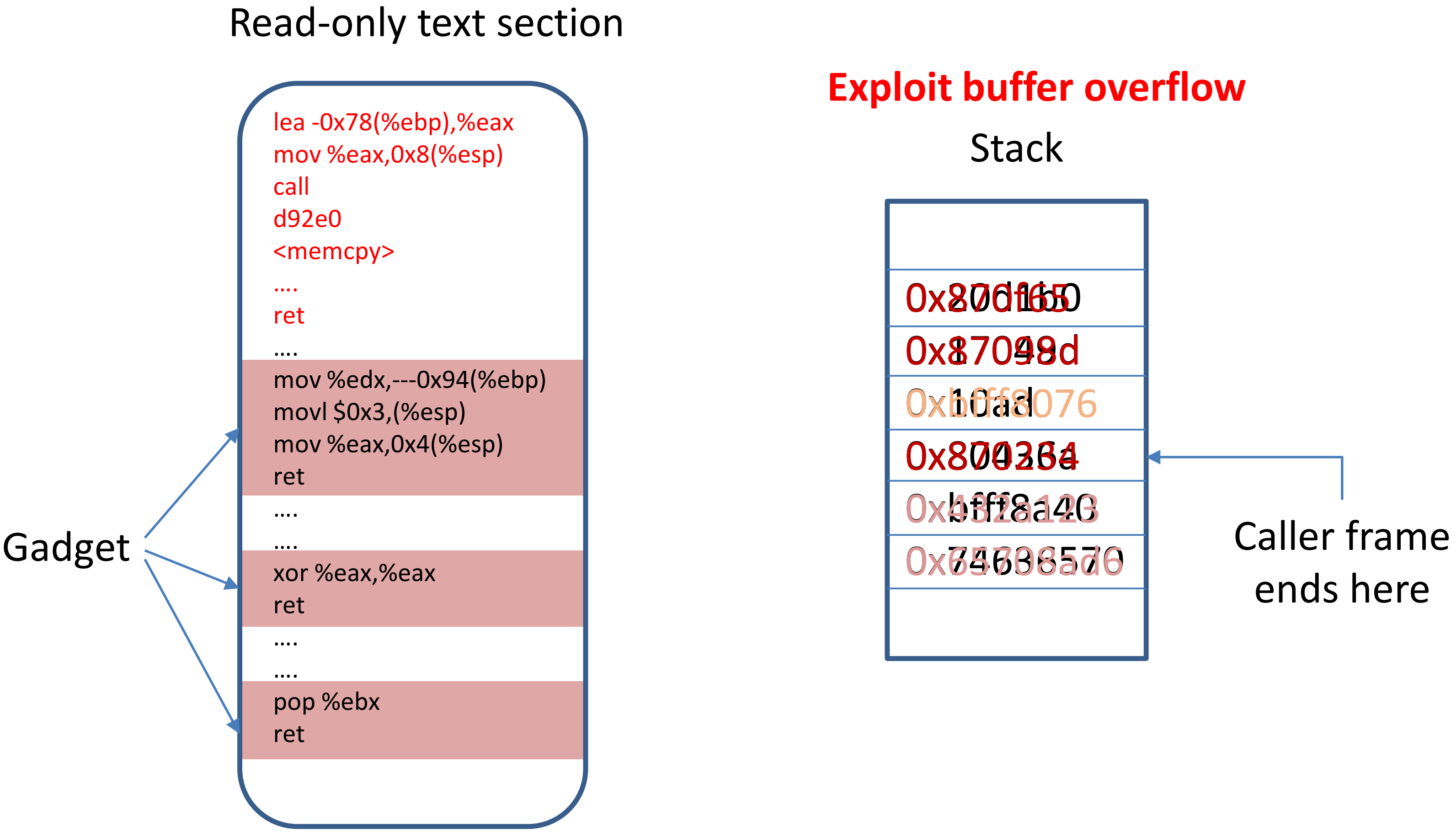
Gadget

Stack

0x20d1b0
0x17049d
0x10ad
0x80436a
0xbfff8a40
0x74636570

Caller frame ends here

New Project: Secure Popcorn

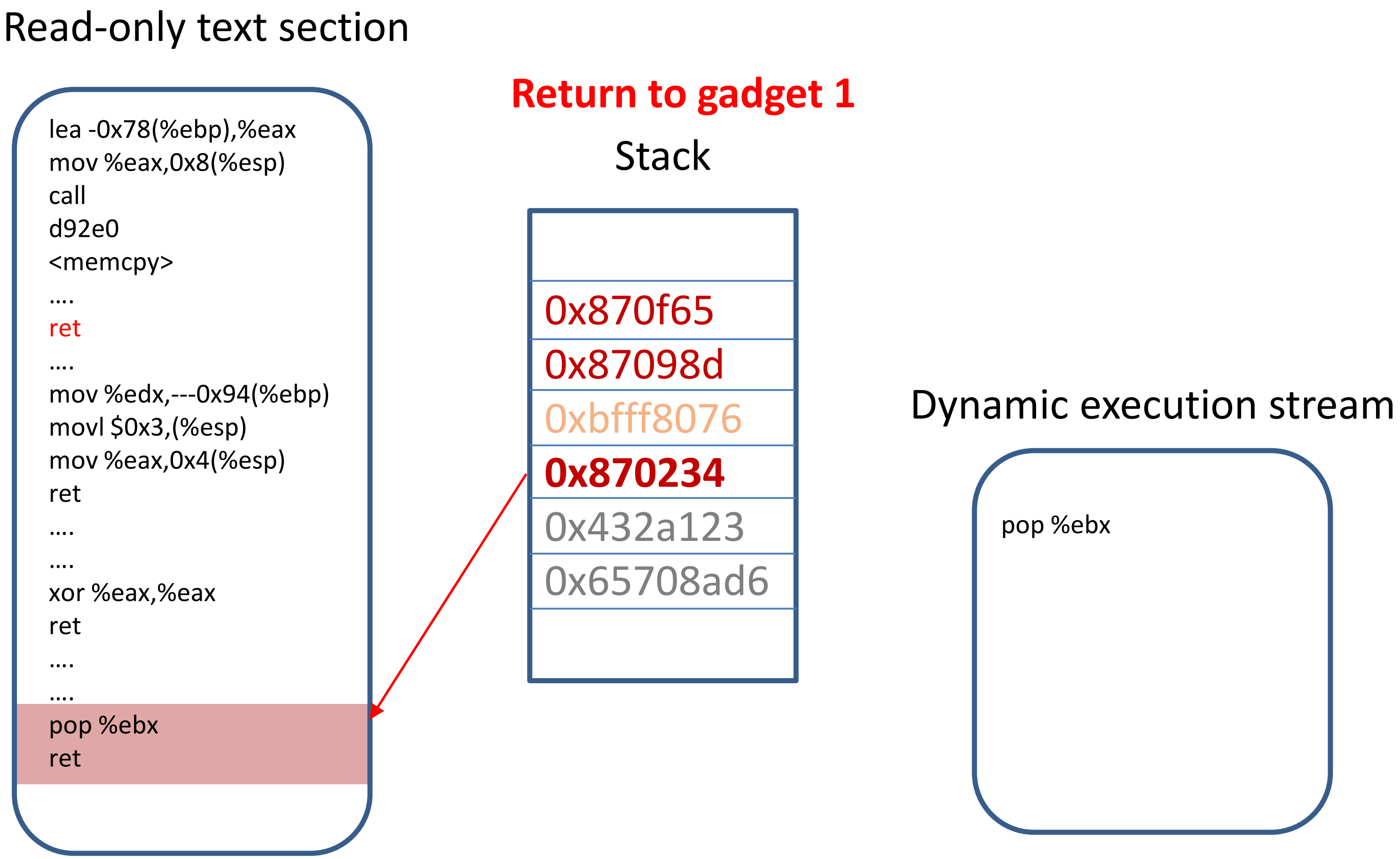


```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
ret
….
….
pop %ebx
ret
```

Exploit buffer overflow

```
0x870f165
0x87098d
0xbfff8076
0x870234
0x432a123
0x65708ad6
```

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
ret
….
….
pop %ebx
ret
```

Return to gadget 1

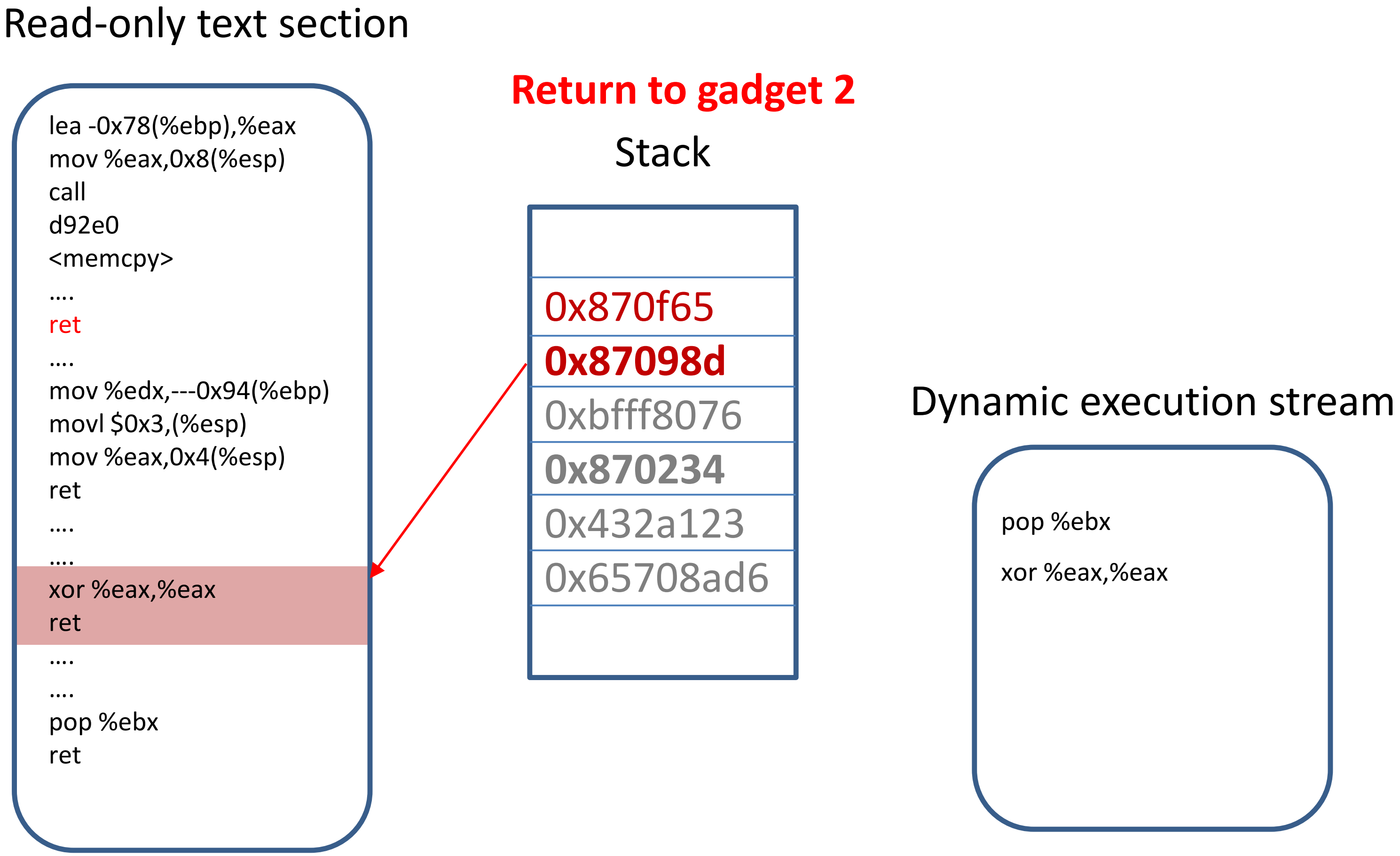
Stack

```
0x870f65
0x87098d
0xbfff8076
0x870234
0x432a123
0x65708ad6
```

Dynamic execution stream

```
pop %ebx
```

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
ret
….
….
pop %ebx
ret
```

Return to gadget 2

Stack

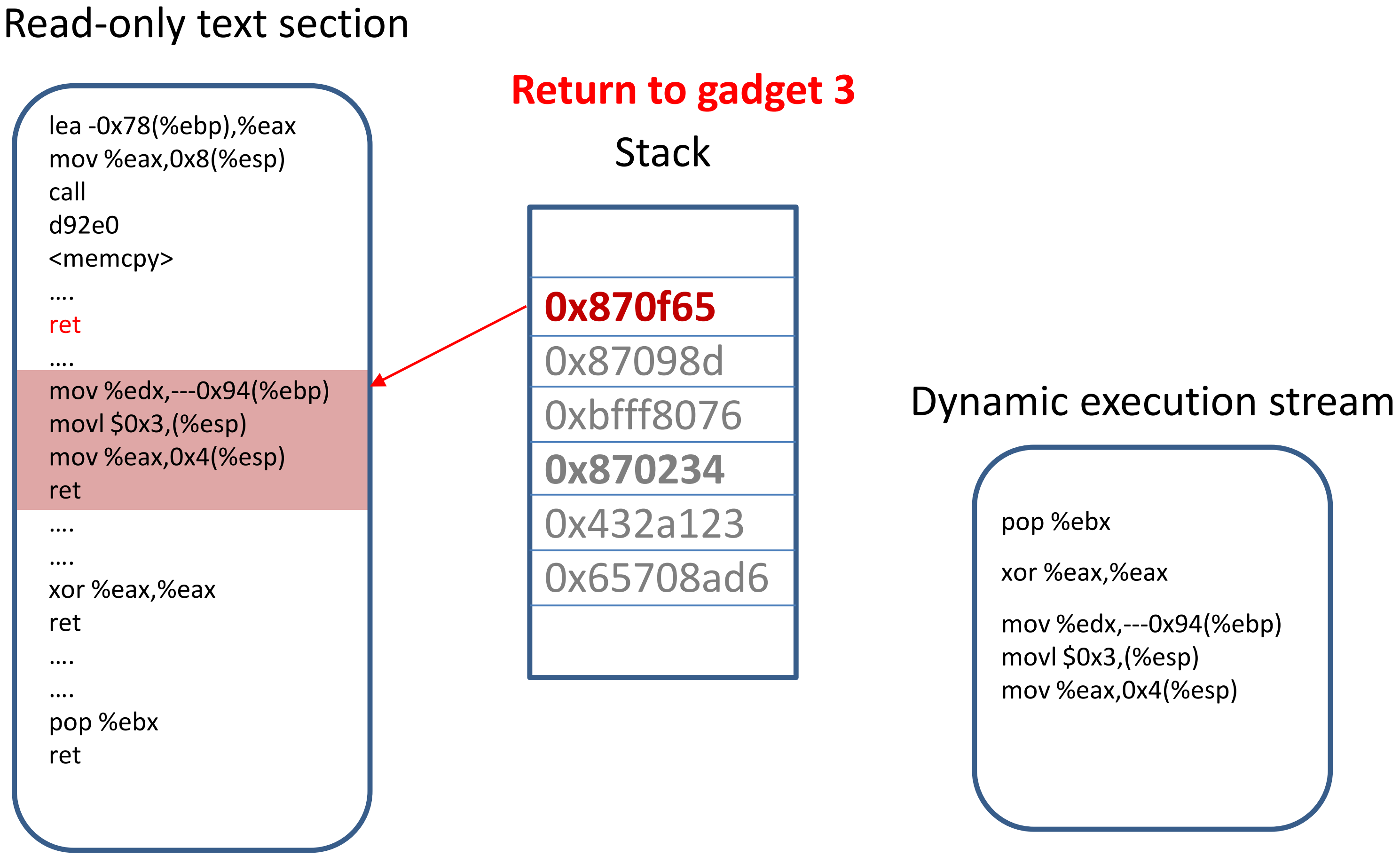
```
0x870f65
0x87098d
0xbfff8076
0x870234
0x432a123
0x65708ad6
```

Dynamic execution stream

```
pop %ebx

xor %eax,%eax
```

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
ret
….
….
pop %ebx
ret
```

Return to gadget 3

Stack

```
0x870f65
0x87098d
0xbfff8076
0x870234
0x432a123
0x65708ad6
```

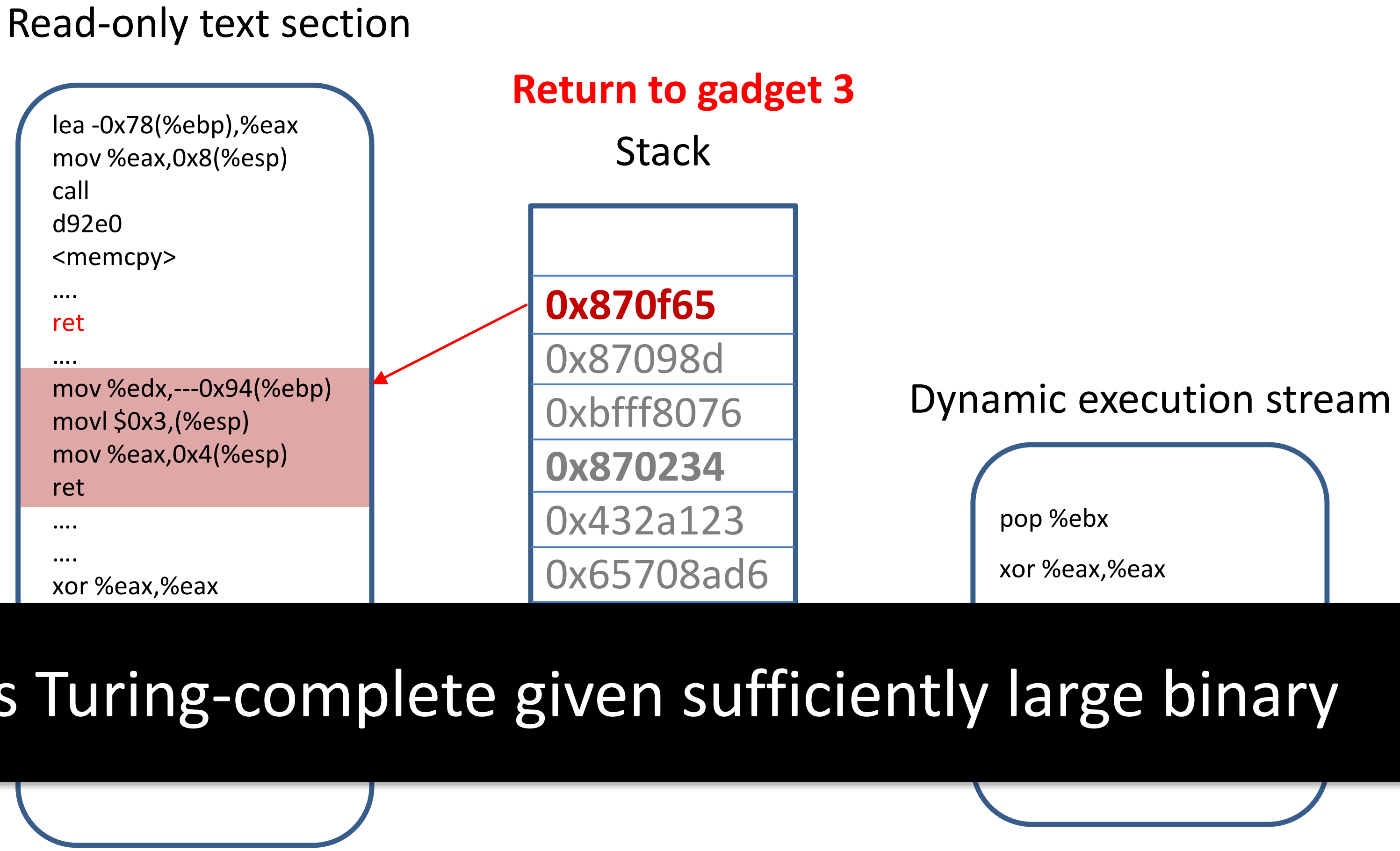
Dynamic execution stream

```
pop %ebx

xor %eax,%eax

mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
```

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
….
ret
….
mov %edx,---0x94(%ebp)
movl $0x3,(%esp)
mov %eax,0x4(%esp)
ret
….
….
xor %eax,%eax
```

Return to gadget 3

Stack

0x870f65
0x87098d
0xbfff8076
0x870234
0x432a123
0x65708ad6

Dynamic execution stream

```
pop %ebx
xor %eax,%eax
```

ROP is Turing-complete given sufficiently large binary

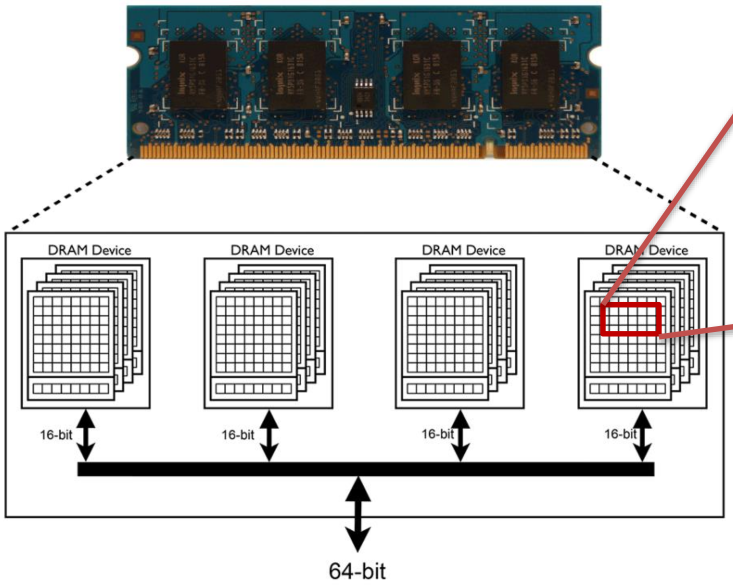
New Project: Secure Popcorn

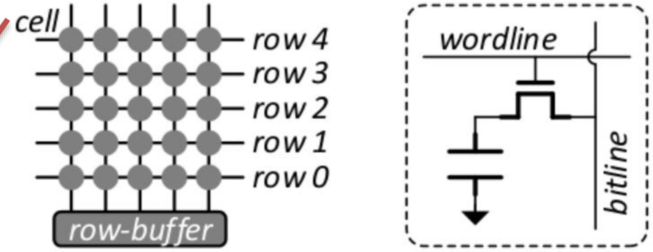
Exploiting the DRAM rowhammer bug to gain kernel privileges

Row hammer attack

Mark Seaborn and Thomas Dullien

New Project: Secure Popcorn





a. Rows of cells **b.** A single cell

Figure 1. DRAM consists of cells

Activating a adjacent row too often causes “disturbance errors”

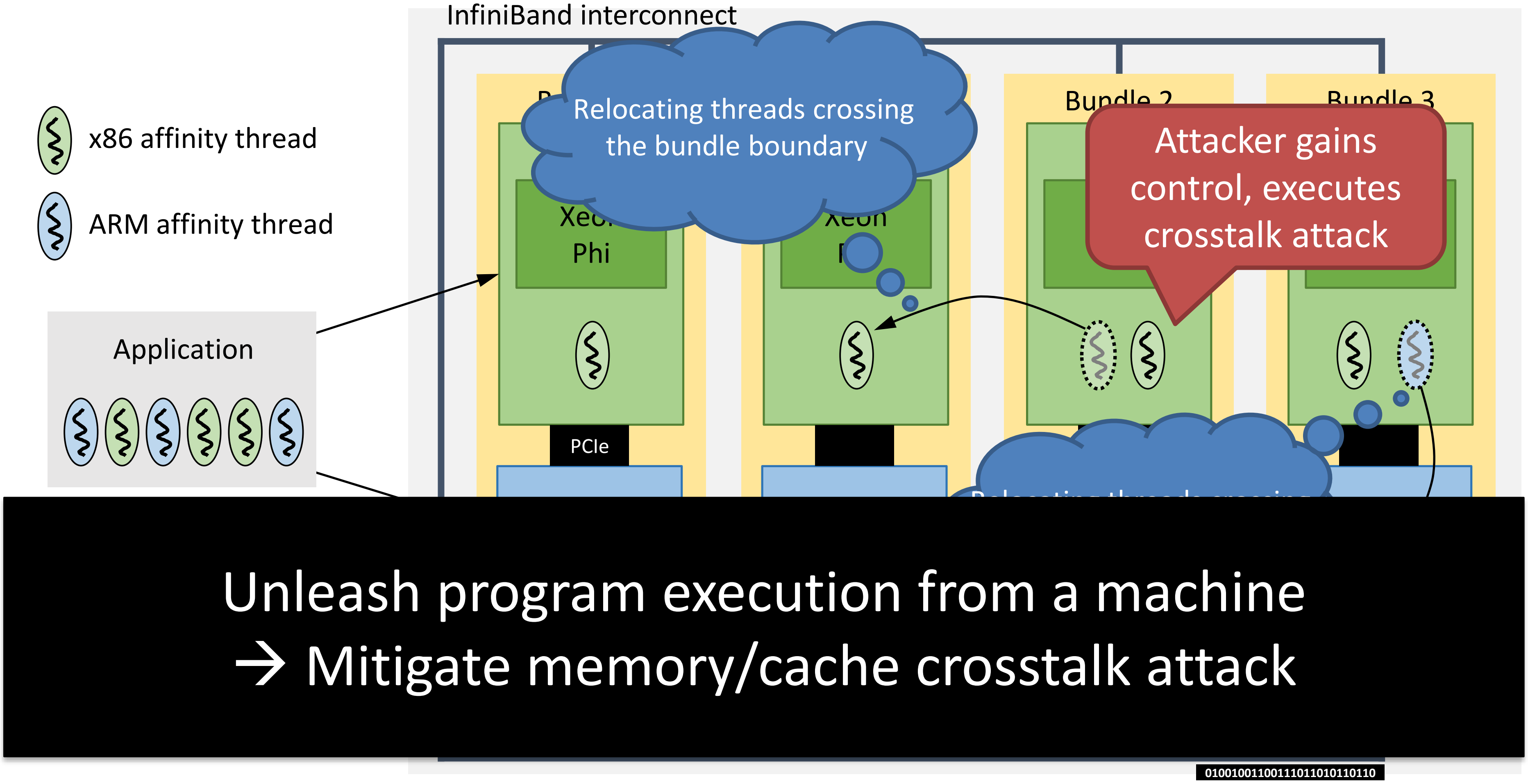
→ One bit-flip error

→ NaCl exploit, kernel exploit

New Project: Secure Popcorn

- **Runtime ISA migration**
  - Move application threads to physically different machine with different ISA
  - Mitigate memory crosstalk attack by running on a physical different machine
- **Runtime ISA randomization**
  - Upon migration, randomize code and program status
  - Thwart attacker’s knowledge by changing program code and data
- **Swift continuous code re-randomization**
  - Continuously re-randomize program code with very low overhead
  - Introduce real-time deadline (one re-randomization cycle) to attackers
- **Runtime integrity check (CFI & DFI)**
  - Check integrity of code and data at ISA boundary
  - Make low-overhead integrity check possible

New Project: Secure Popcorn

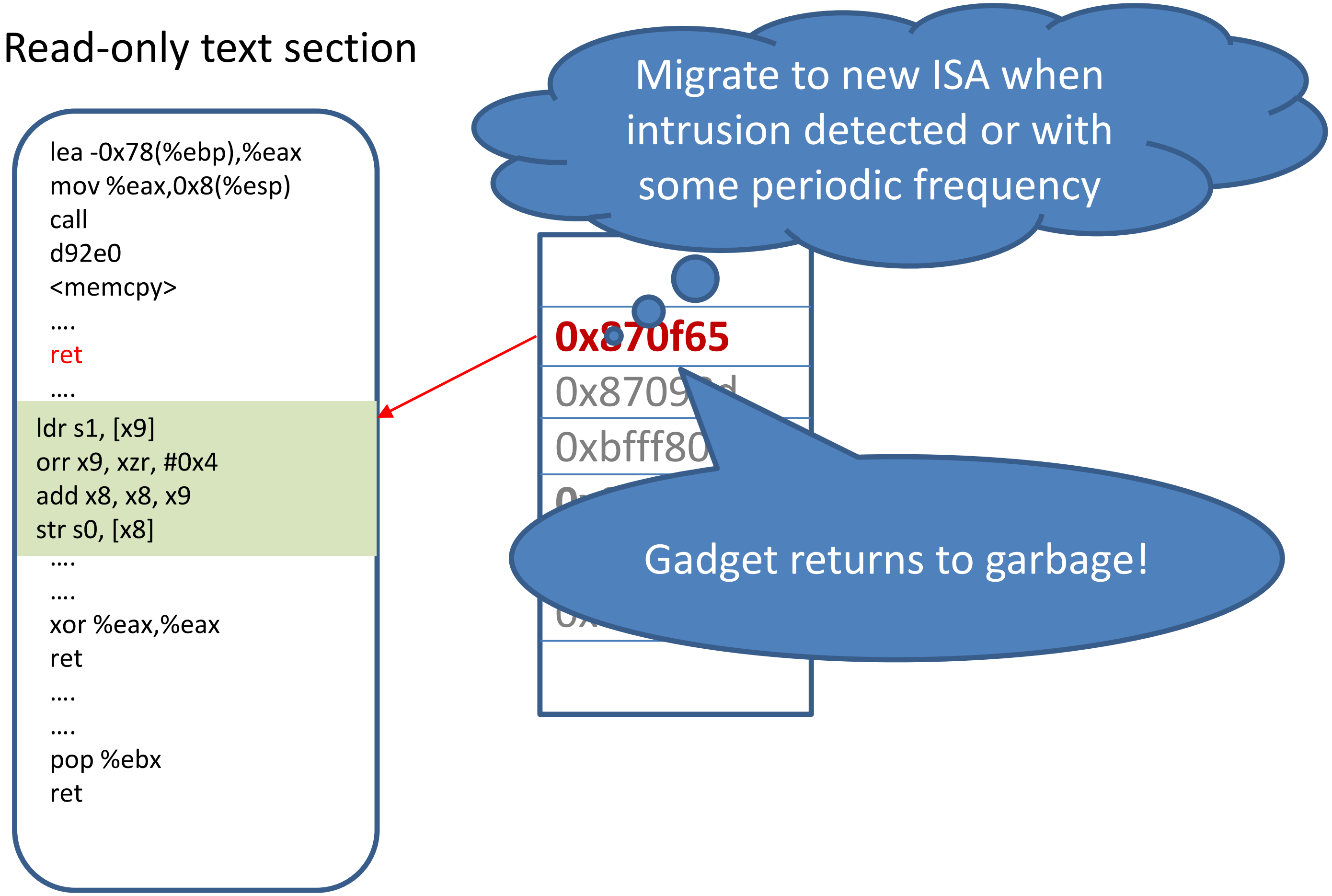


Relocating threads crossing the bundle boundary

Attacker gains control, executes crosstalk attack

Unleash program execution from a machine
→ Mitigate memory/cache crosstalk attack

New Project: Secure Popcorn



Read-only text section

```
lea -0x78(%ebp),%eax
mov %eax,0x8(%esp)
call
d92e0
<memcpy>
....
ret
....
ldr s1, [x9]
orr x9, xzr, #0x4
add x8, x8, x9
str s0, [x8]
....
....
xor %eax,%eax
ret
....
....
pop %ebx
ret
```

Migrate to new ISA when intrusion detected or with some periodic frequency

Gadget returns to garbage!

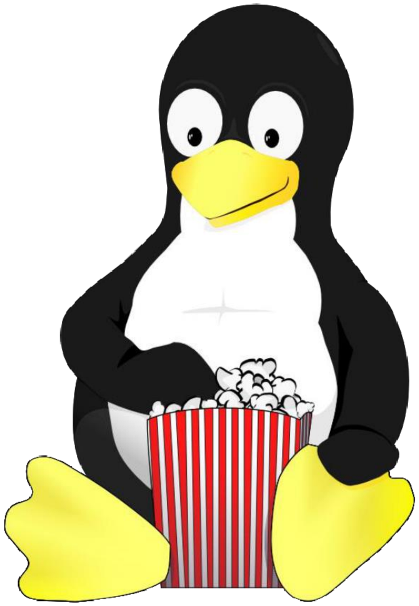
Conclusion

- Heterogeneity is permeating all corners of computer architecture
- Popcorn Linux gives developers a better way to program heterogeneous-ISA systems – shared memory!
  - A compiler which builds multi-ISA binaries
  - An OS which enables cross-ISA thread and data migration
  - A state transformation runtime which converts ISA-specific data
- Allows developers to transparently take advantage of heterogeneity
  - Performance
  - Power/energy efficiency
  - Security

More Information

- Popcorn Linux is open source and available online at http://popcornlinux.org

- This work is supported in part by ONR under grants N00014-13-1-0317 and N00014-16-1-2711, AFOSR under grant FA9550-14-1-0163, and NAVSEA/NEEC under grants 3003279297 and N00174-16-C-0018. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of ONR, AFOSR, and NAVSEA.



Questions?