Hardware Support for On-Demand Software Analysis

Joseph L. Greathouse

Advanced Computer Architecture Laboratory

University of Michigan

December 8, 2011

Software Errors Abound

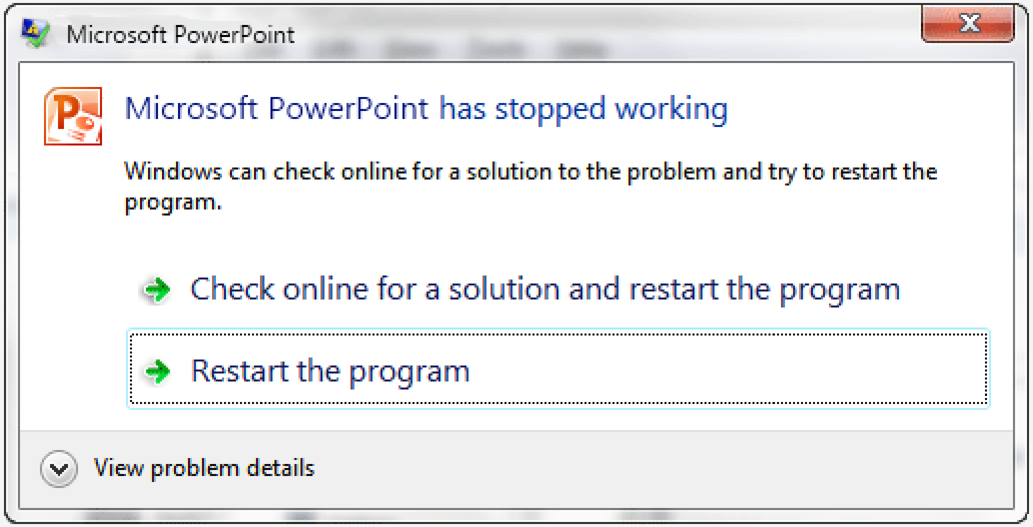
- NIST: Software errors cost U.S. ~$60 billion/year

Software Errors Abound

- NIST: Software errors cost U.S. ~$60 billion/year

Software Errors Abound

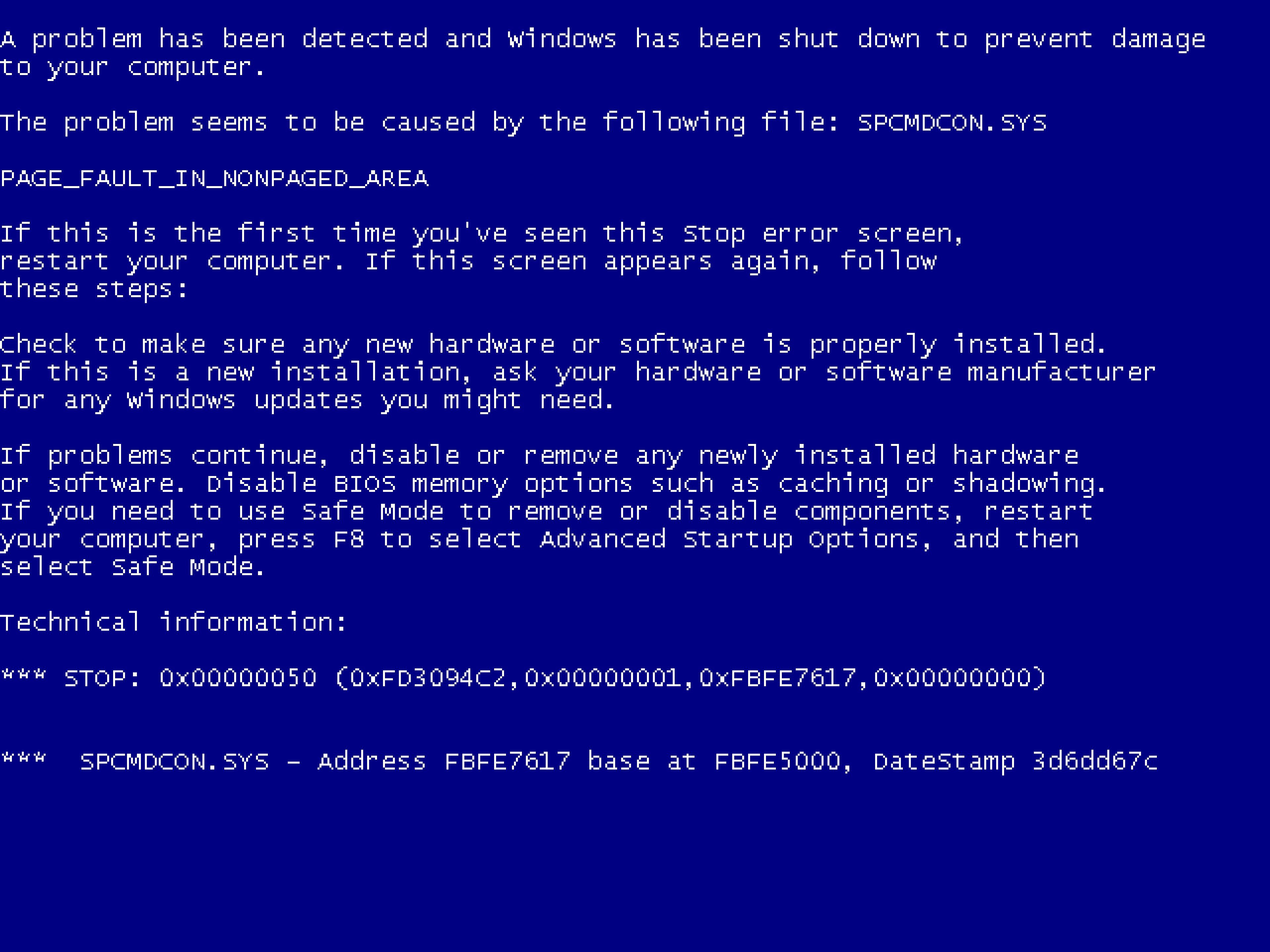
- NIST: Software errors cost U.S. ~$60 billion/year



Microsoft PowerPoint

Microsoft PowerPoint has stopped working

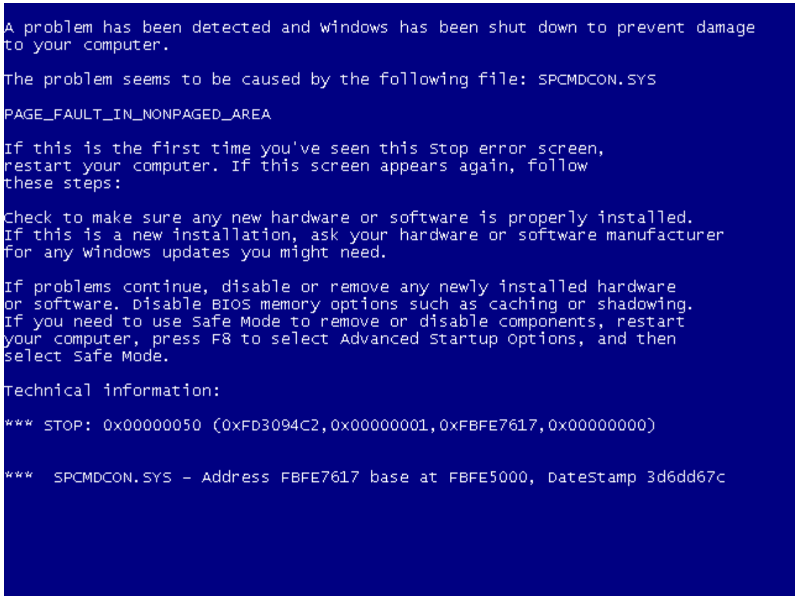
Windows can check online for a solution to the problem and try to restart the program.



```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)


***  SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c
```

Software Errors Abound

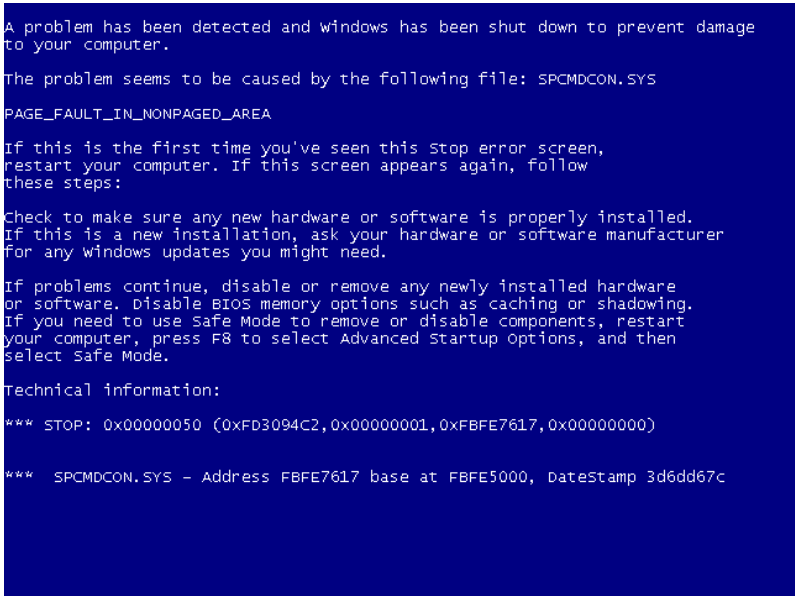
- NIST: Software errors cost U.S. ~$60 billion/year



```
A problem has been detected and Windows has been shut down to prevent damage
to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)


***  SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c
```

Software Errors Abound

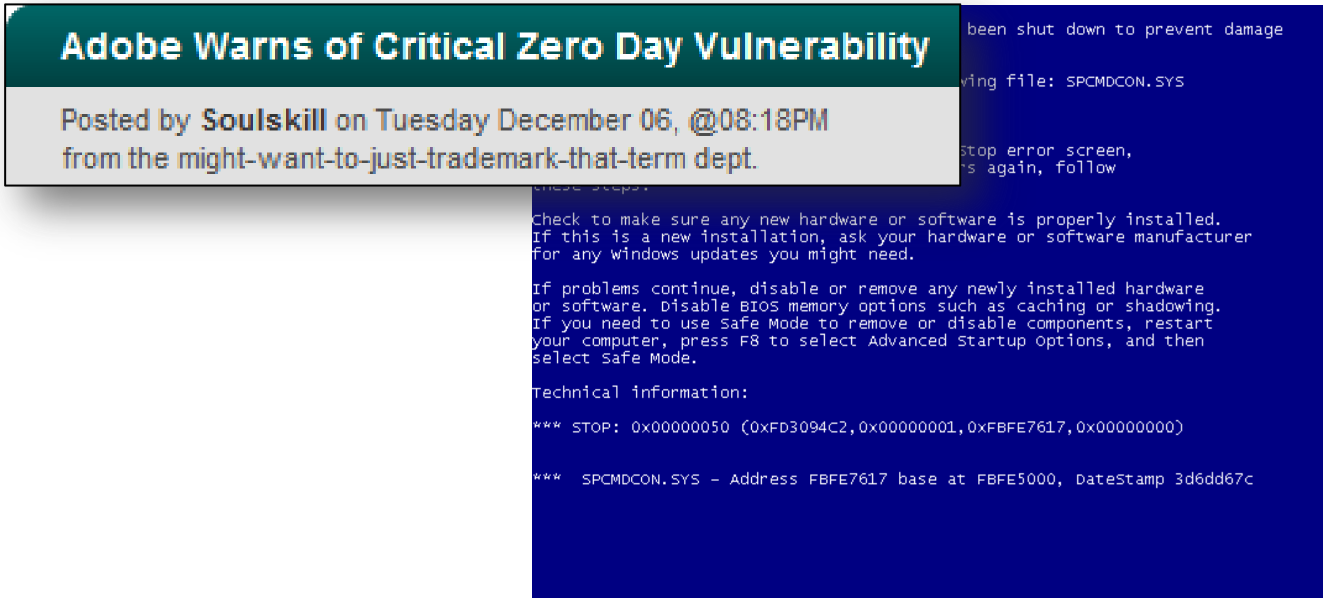
- NIST: Software errors cost U.S. ~$60 billion/year
- FBI: Security Issues cost U.S. $67 billion/year
  - >⅓ from viruses, network intrusion, etc.



```
A problem has been detected and Windows has been shut down to prevent damage
to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)


***  SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c
```

Software Errors Abound

- NIST: Software errors cost U.S. ~$60 billion/year
- FBI: Security Issues cost U.S. $67 billion/year
  - >⅓ from viruses, network intrusion, etc.



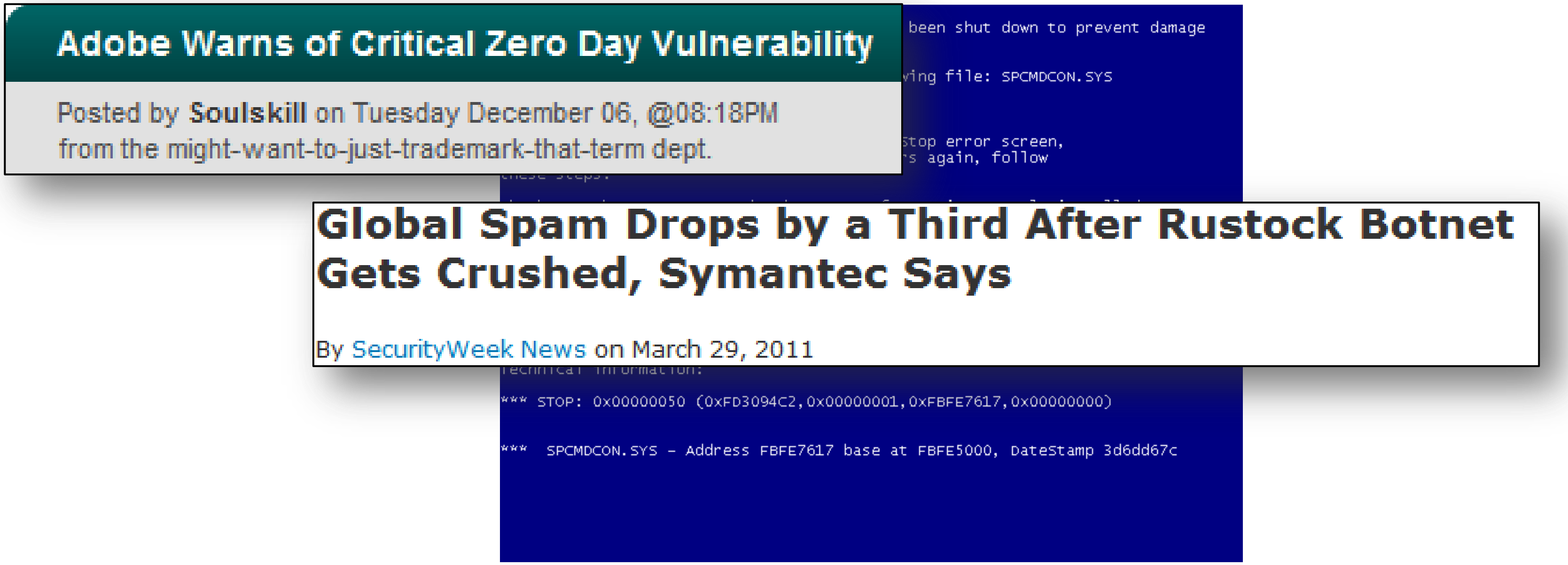
Adobe Warns of Critical Zero Day Vulnerability

Posted by **Soulskill** on Tuesday December 06, @08:18PM
from the might-want-to-just-trademark-that-term dept.

```
been shut down to prevent damage

ving file: SPCMDCON.SYS


stop error screen,
s again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)


***  SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c
```

Software Errors Abound

- NIST: Software errors cost U.S. ~$60 billion/year
- FBI: Security Issues cost U.S. $67 billion/year
  - >⅓ from viruses, network intrusion, etc.



Adobe Warns of Critical Zero Day Vulnerability

Posted by **Soulskill** on Tuesday December 06, @08:18PM
from the might-want-to-just-trademark-that-term dept.

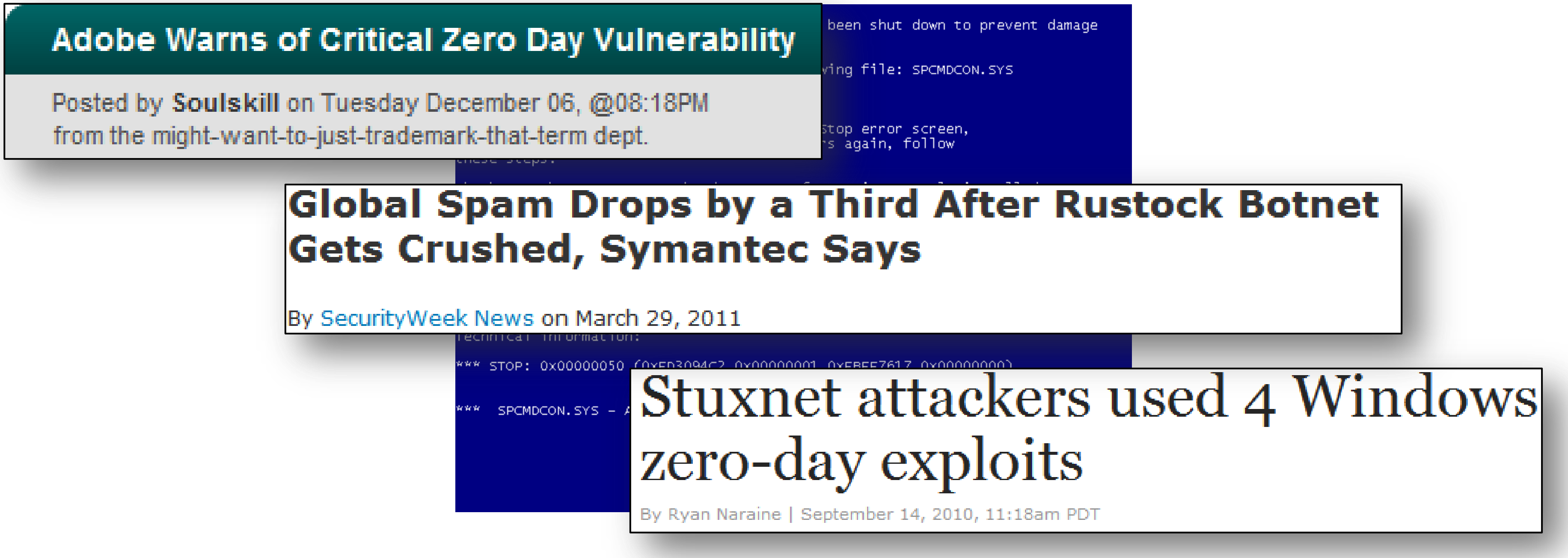
Global Spam Drops by a Third After Rustock Botnet Gets Crushed, Symantec Says

By SecurityWeek News on March 29, 2011

```
been shut down to prevent damage

wing file: SPCMDCON.SYS

stop error screen,
rs again, follow
these steps:

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)

***  SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c
```

Software Errors Abound

- NIST: Software errors cost U.S. ~$60 billion/year
- FBI: Security Issues cost U.S. $67 billion/year
  - >⅓ from viruses, network intrusion, etc.



Adobe Warns of Critical Zero Day Vulnerability

Posted by **Soulskill** on Tuesday December 06, @08:18PM
from the might-want-to-just-trademark-that-term dept.

```
been shut down to prevent damage
ving file: SPCMDCON.SYS

stop error screen,
s again, follow
these steps:

Technical information:
*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)

***  SPCMDCON.SYS - A
```

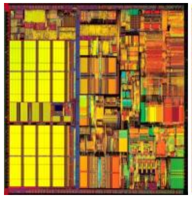
Global Spam Drops by a Third After Rustock Botnet Gets Crushed, Symantec Says

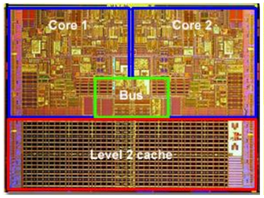
By SecurityWeek News on March 29, 2011

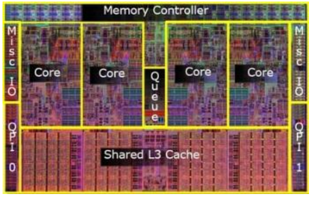
Stuxnet attackers used 4 Windows zero-day exploits

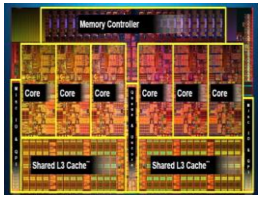
By Ryan Naraine | September 14, 2010, 11:18am PDT

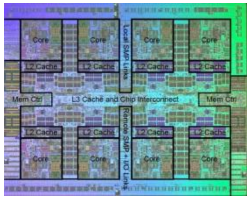
Hardware Plays a Role

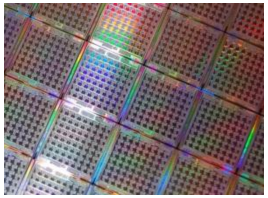


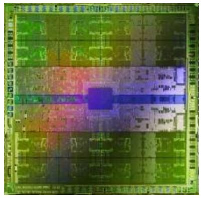




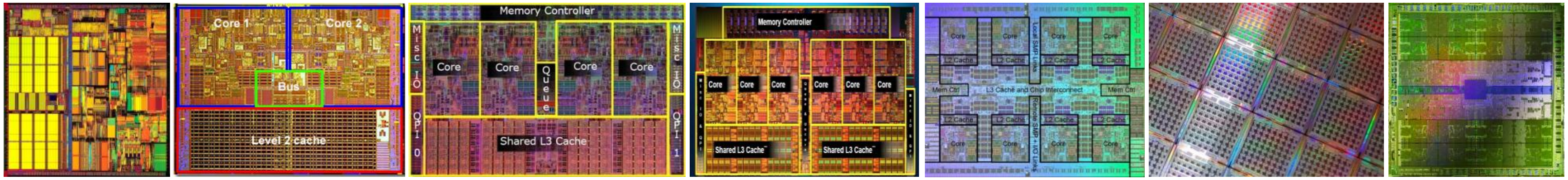






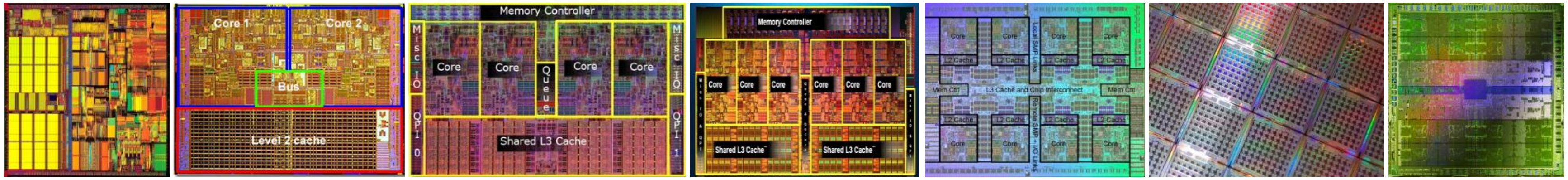


Hardware Plays a Role

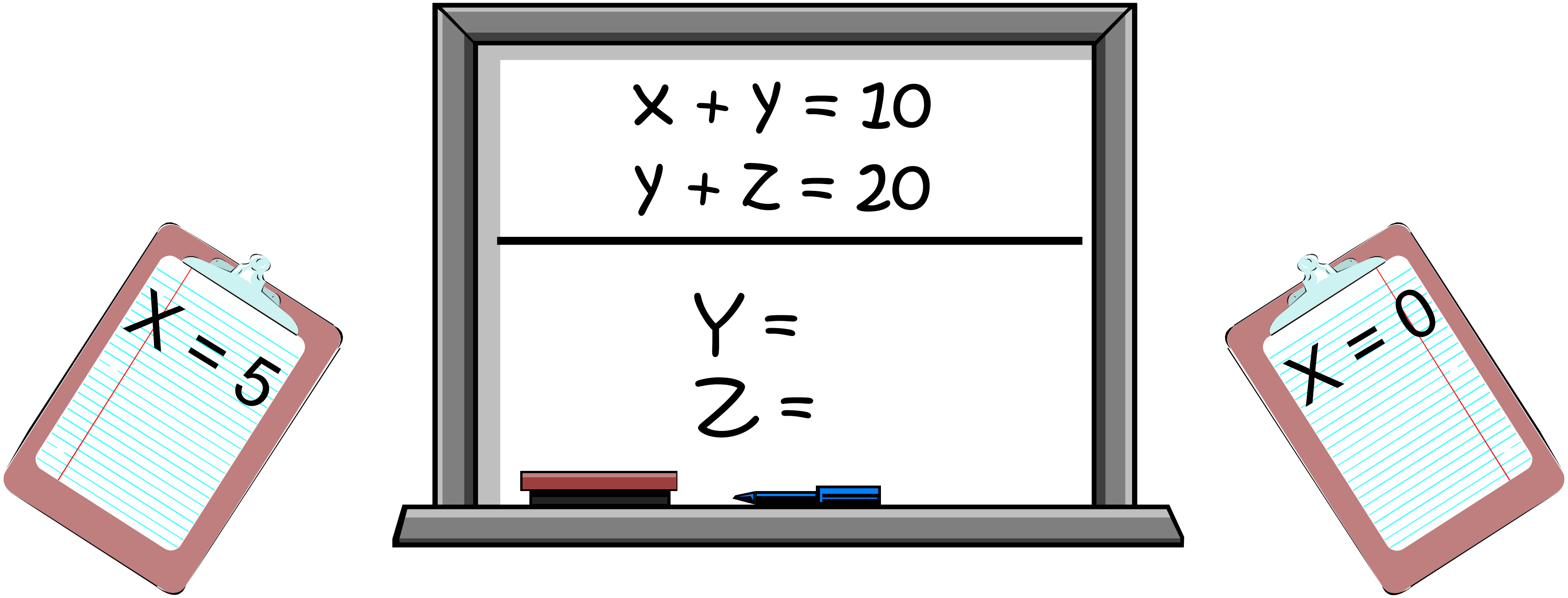


Parallel Programming is Here – And it’s Hard!

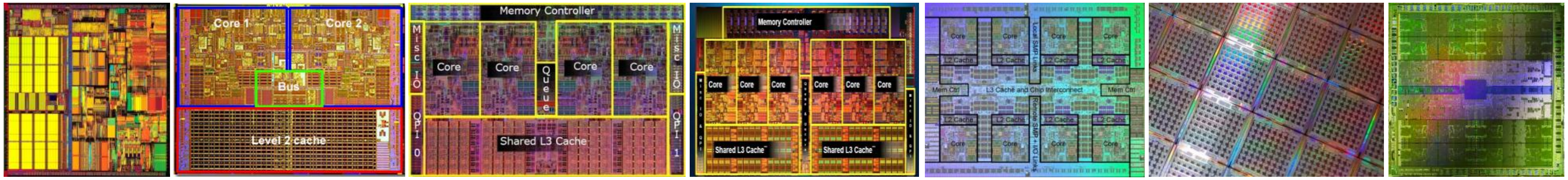
Hardware Plays a Role



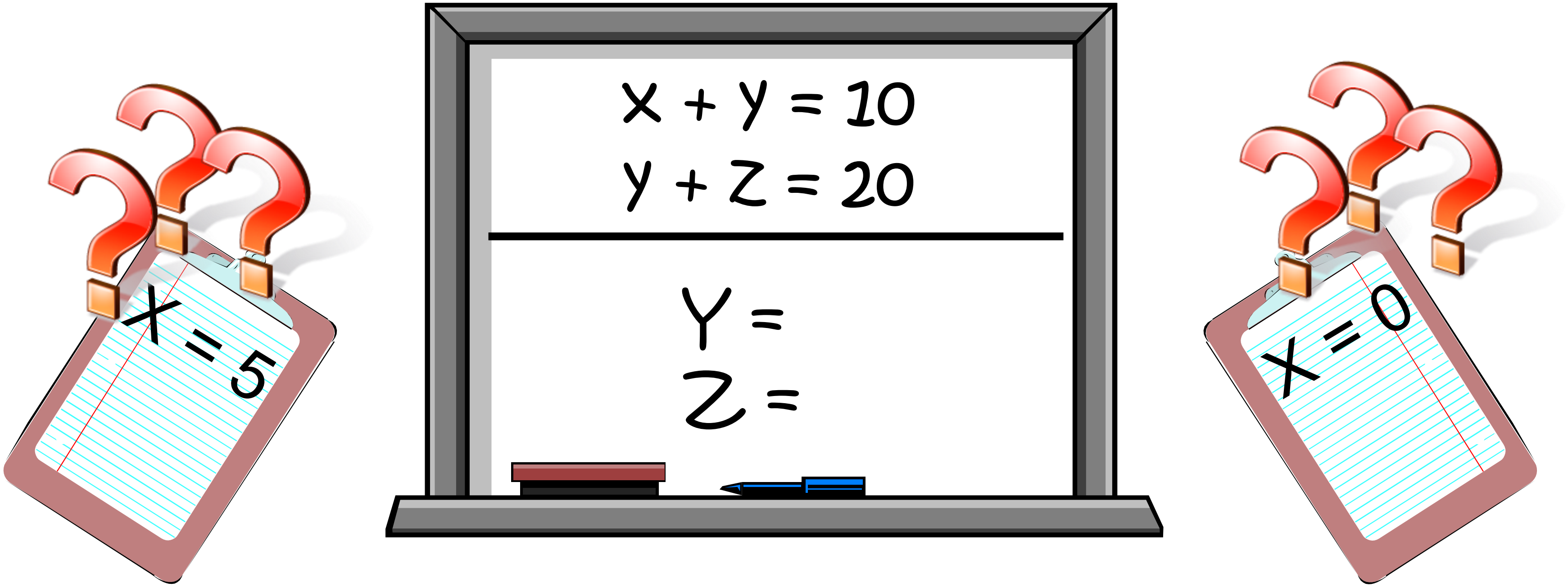
Parallel Programming is Here – And it's Hard!



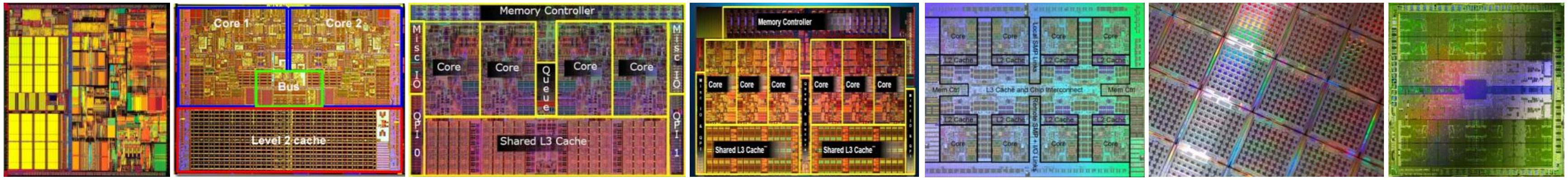
Hardware Plays a Role



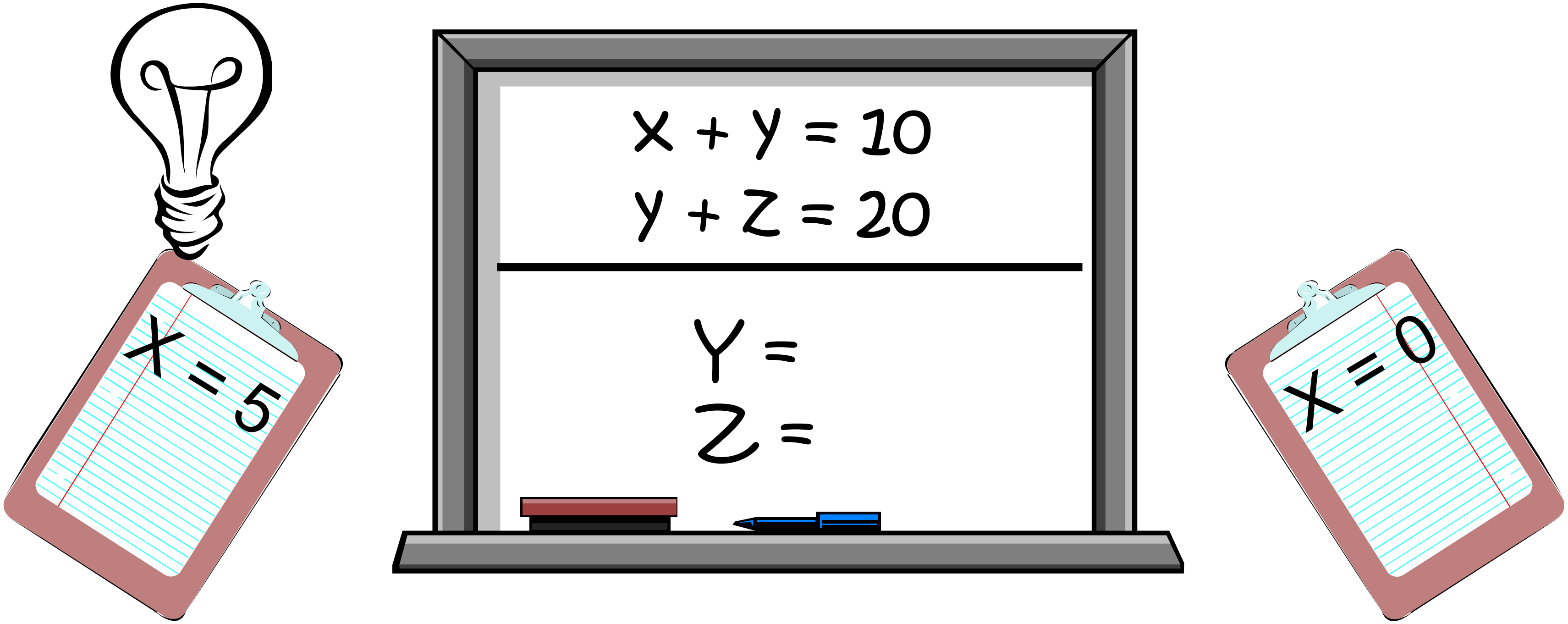
Parallel Programming is Here – And it's Hard!



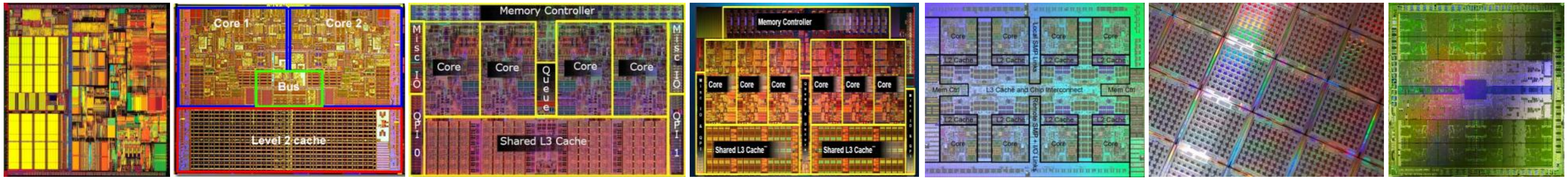
Hardware Plays a Role



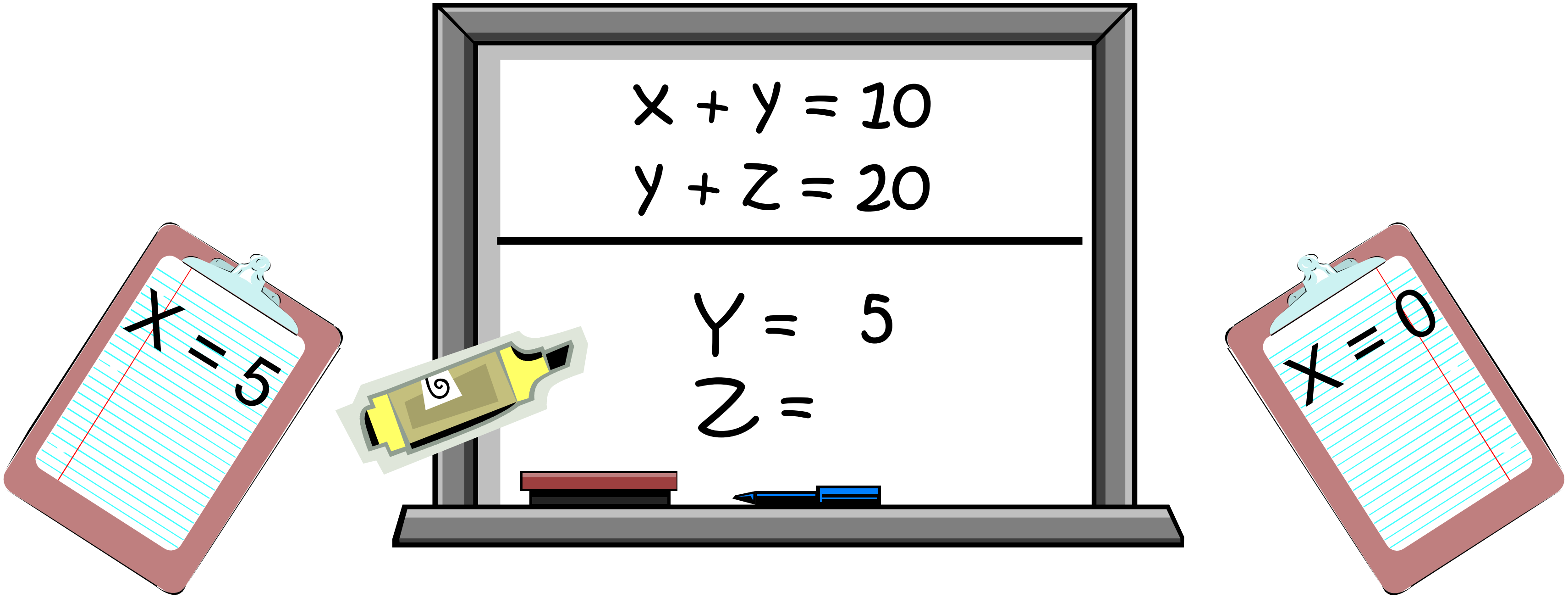
Parallel Programming is Here – And it's Hard!



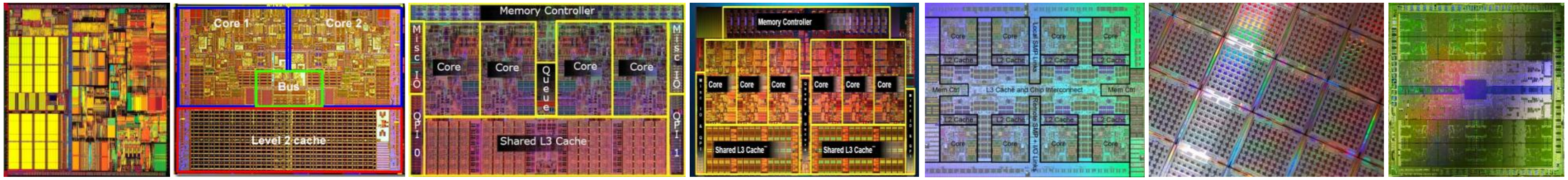
Hardware Plays a Role



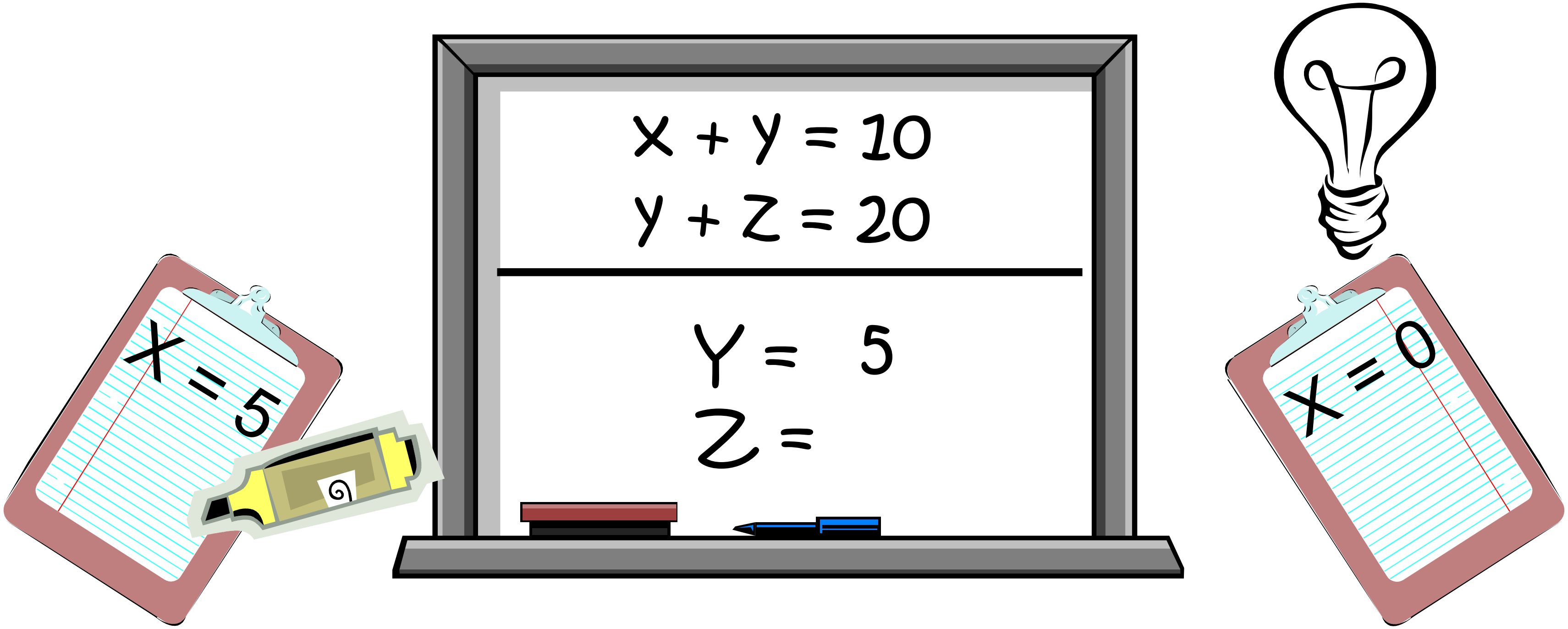
Parallel Programming is Here – And it’s Hard!



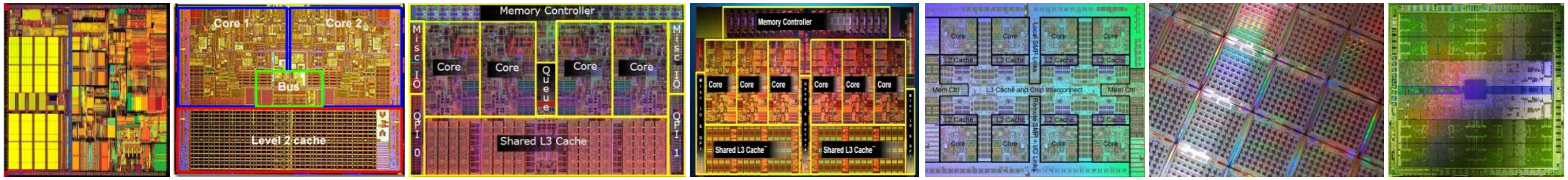
Hardware Plays a Role



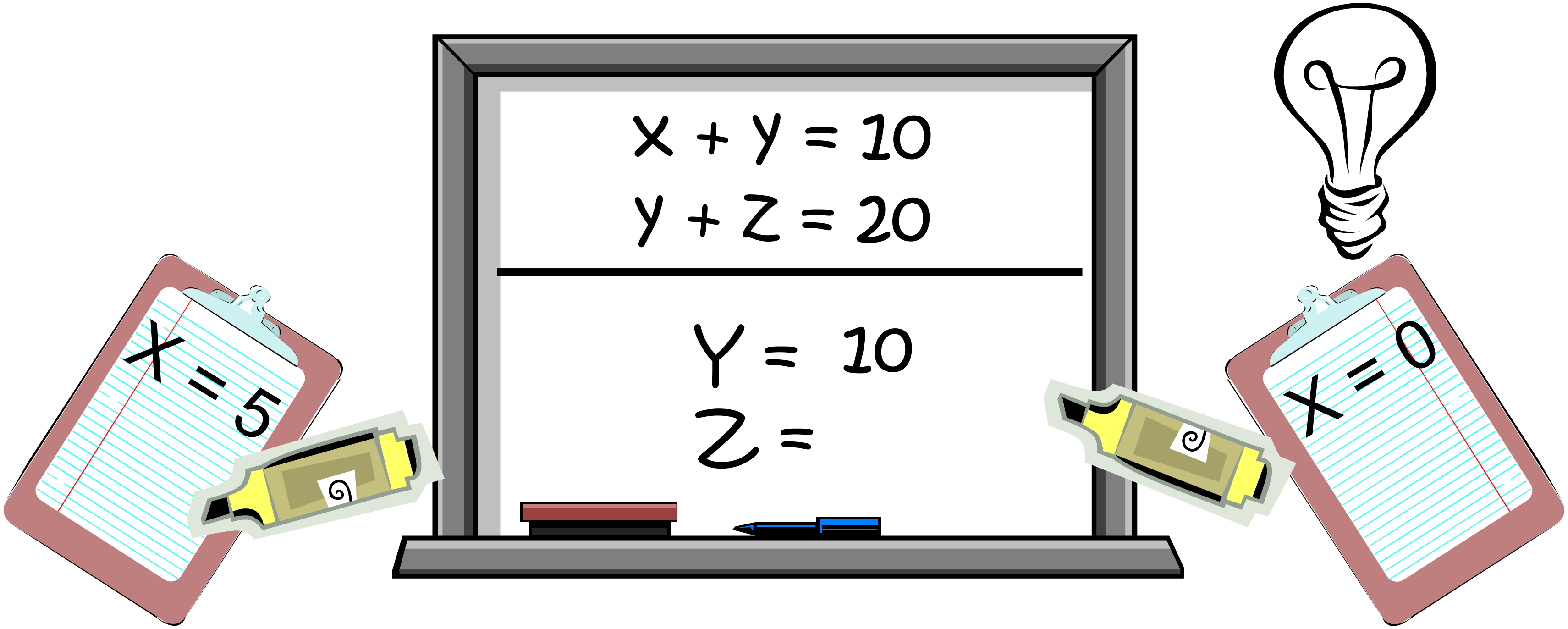
Parallel Programming is Here – And it's Hard!



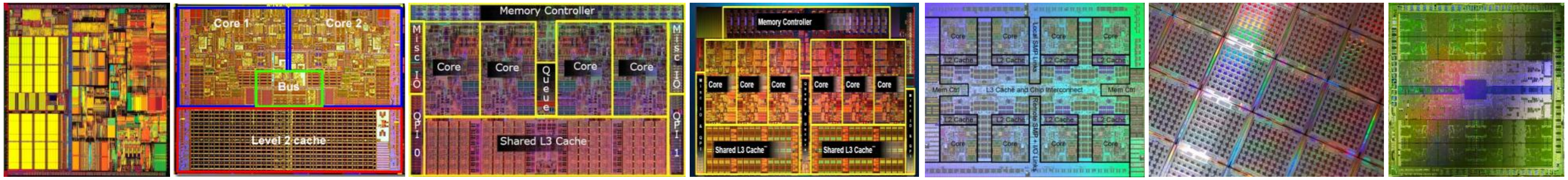
Hardware Plays a Role



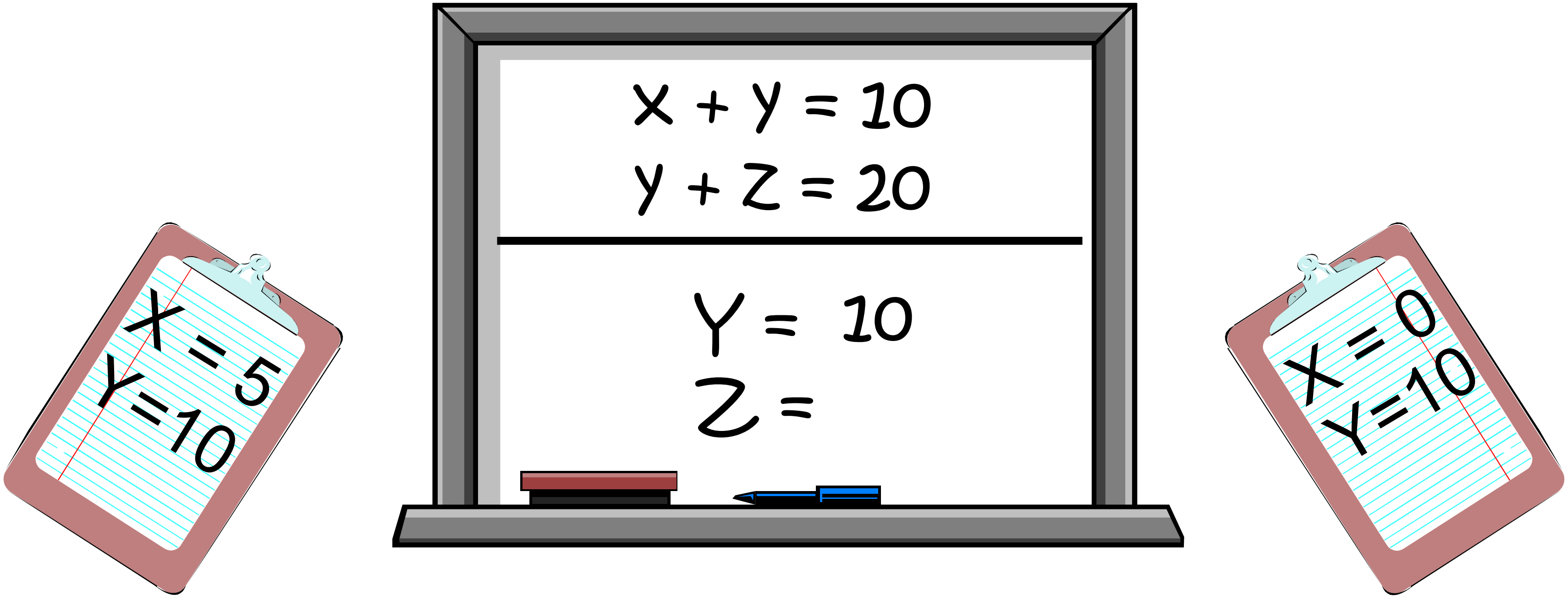
Parallel Programming is Here – And it's Hard!



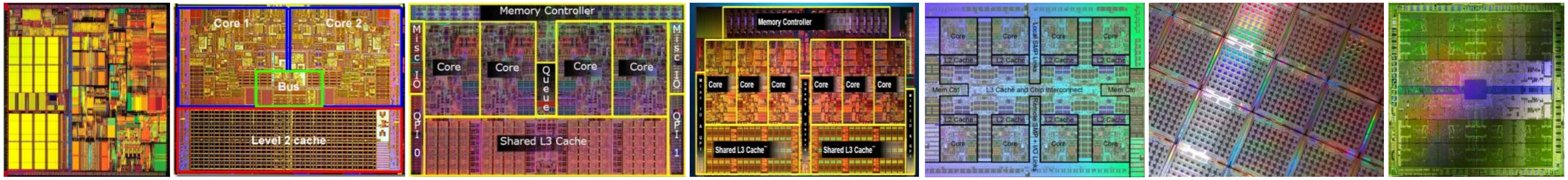
Hardware Plays a Role



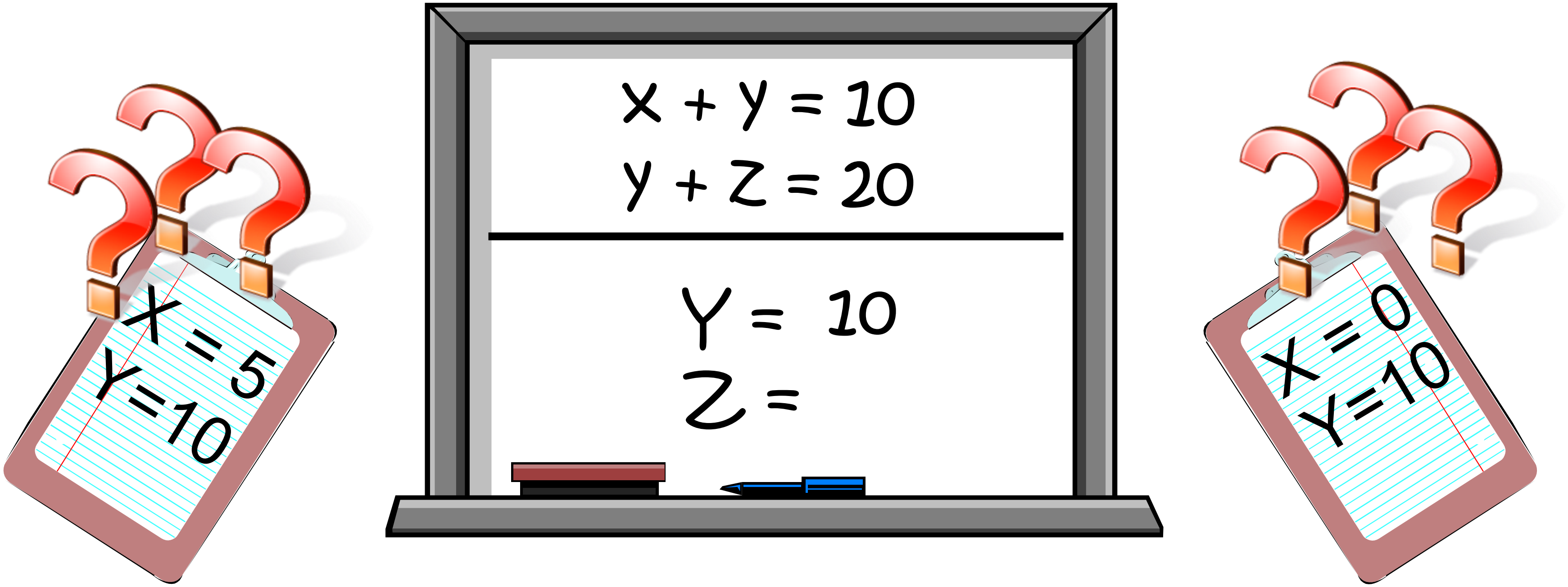
Parallel Programming is Here – And it's Hard!



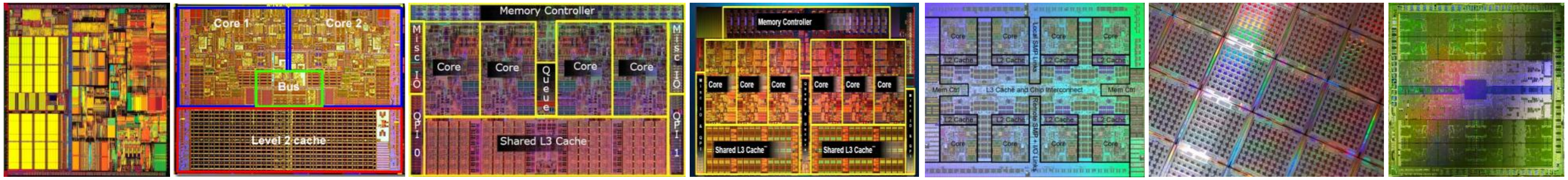
Hardware Plays a Role



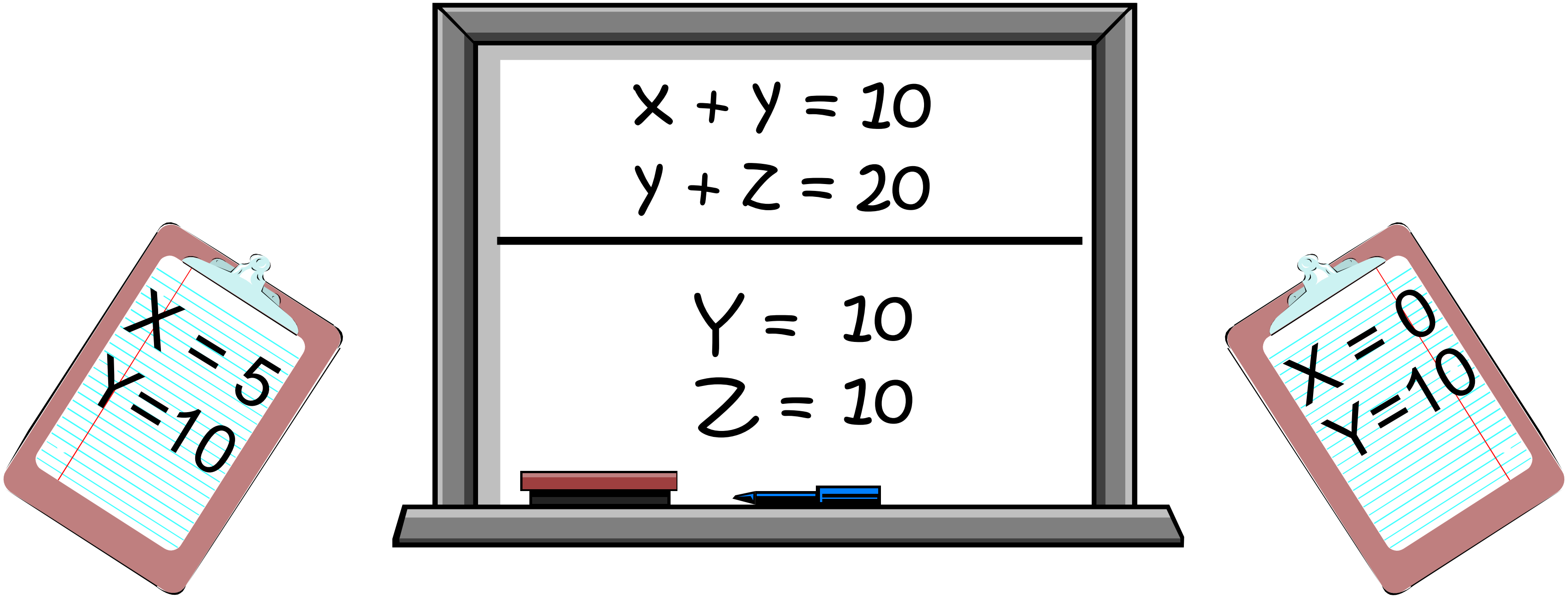
Parallel Programming is Here – And it’s Hard!



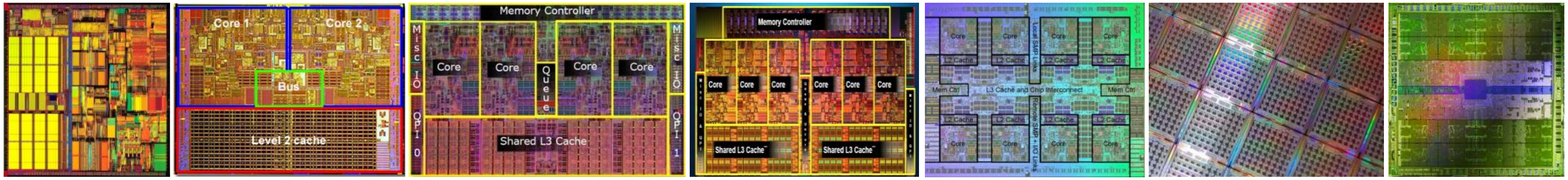
Hardware Plays a Role



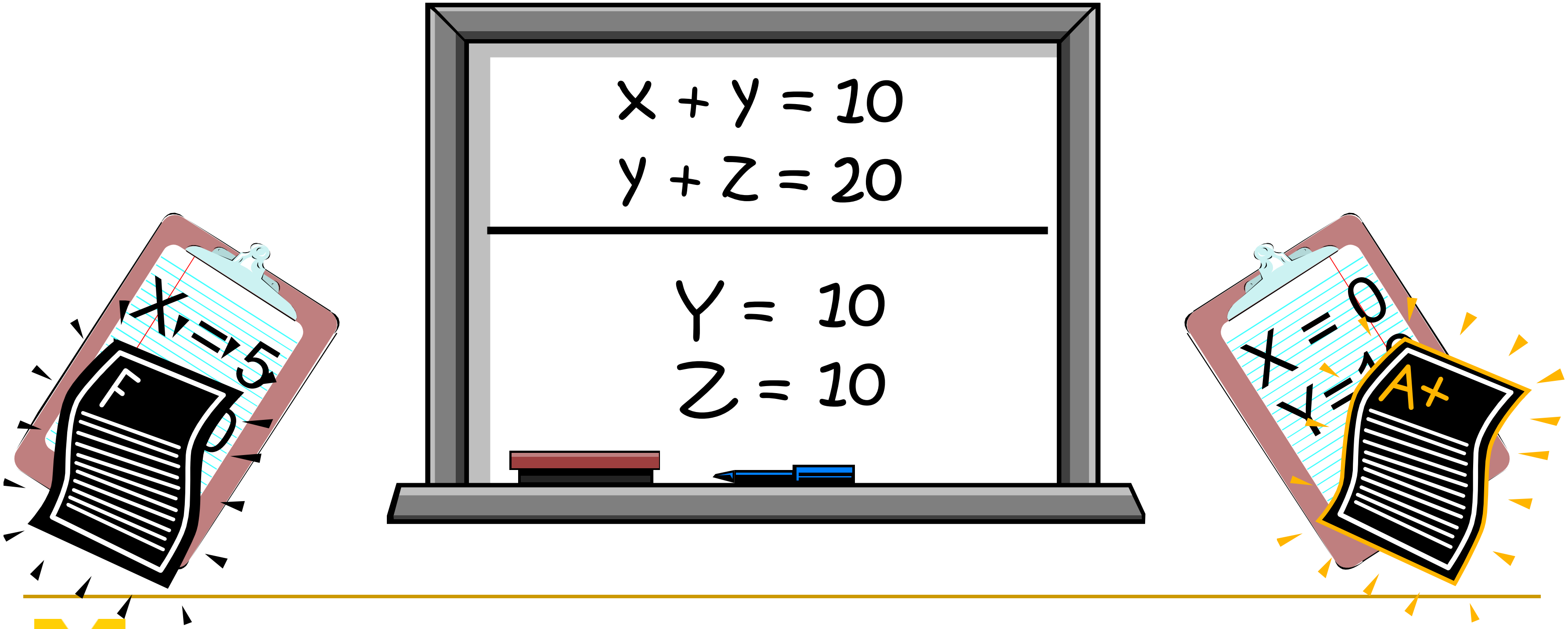
Parallel Programming is Here – And it’s Hard!



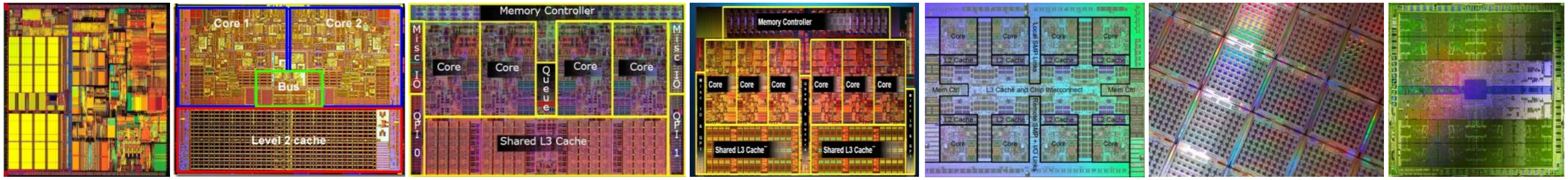
Hardware Plays a Role



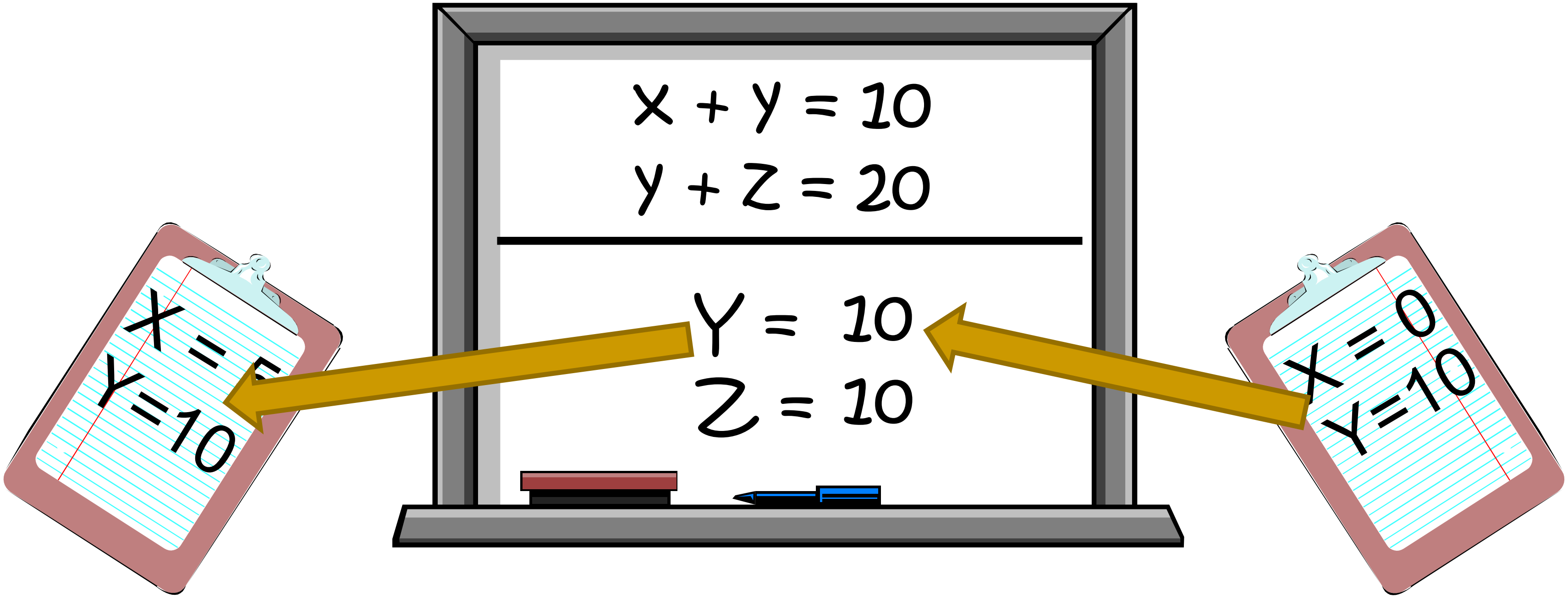
Parallel Programming is Here – And it's Hard!



Hardware Plays a Role

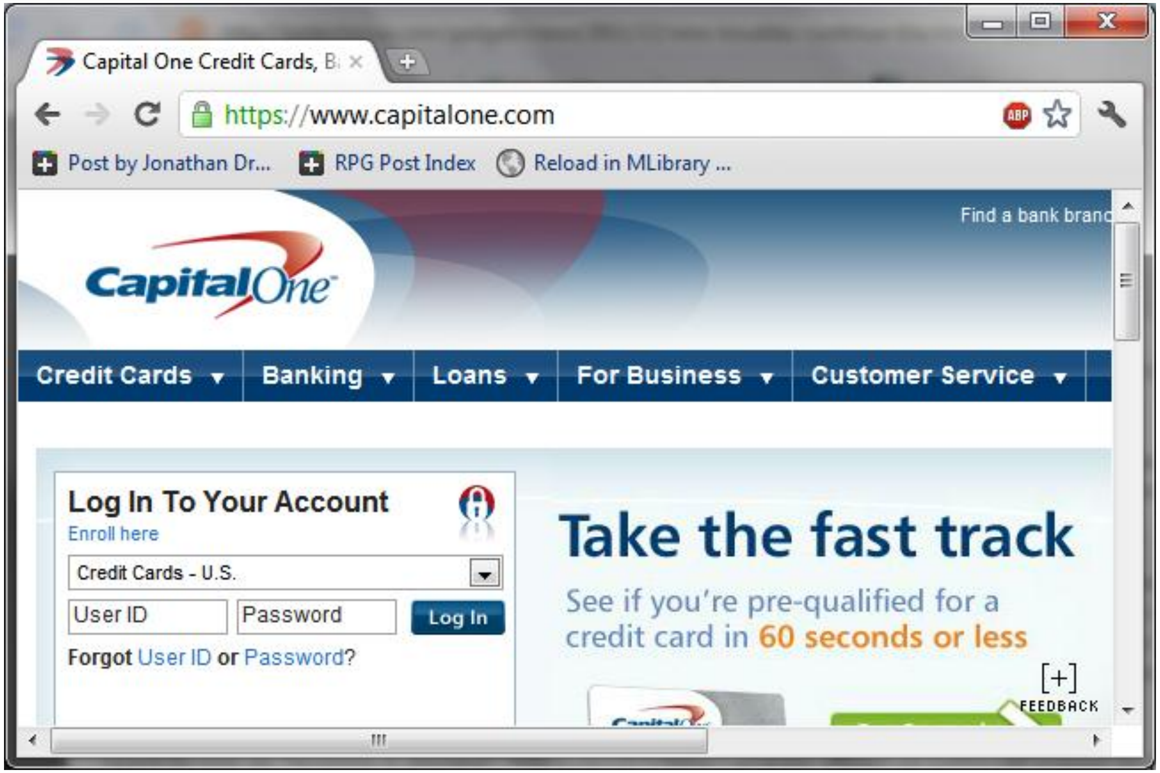


Parallel Programming is Here – And it's Hard!



Example of a Modern Bug

Nov. 2010 OpenSSL Security Flaw



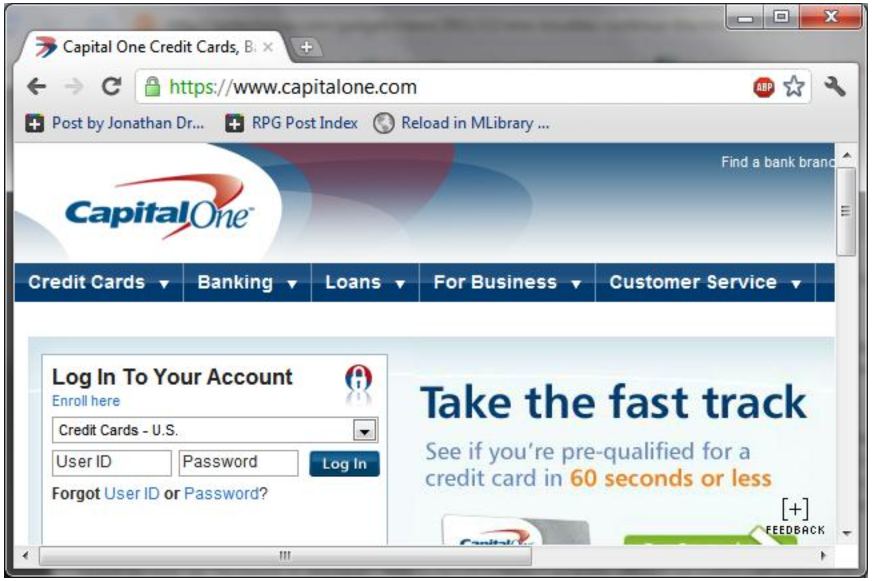
Example of a Modern Bug

```
if(ptr == NULL) {
   len=thread_local->mylen;
   ptr=malloc(len);
   memcpy(ptr, data, len);
}
```

Example of a Modern Bug

Thread 1
mylen=small

Thread 2
mylen=large

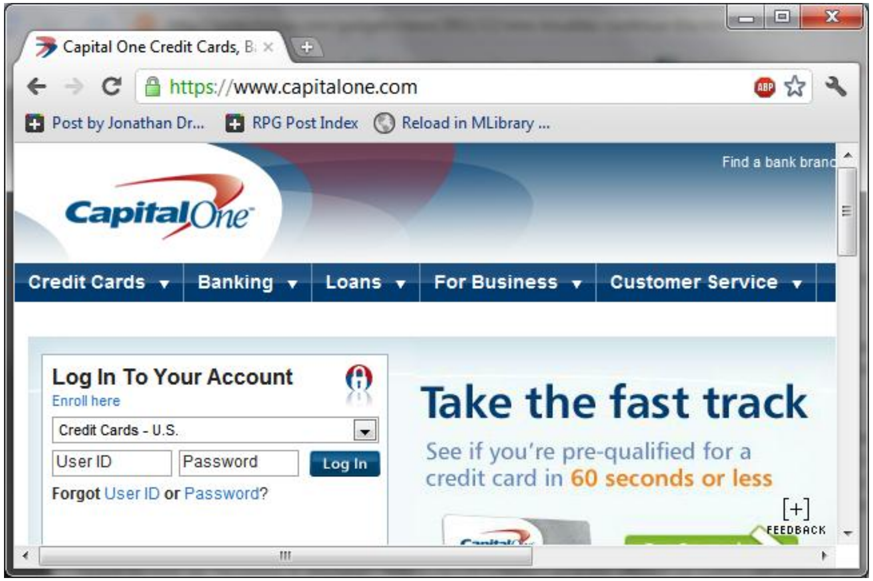


Log In To Your Account

Forgot User ID or Password?

Take the fast track

See if you're pre-qualified for a credit card in 60 seconds or less



Log In To Your Account

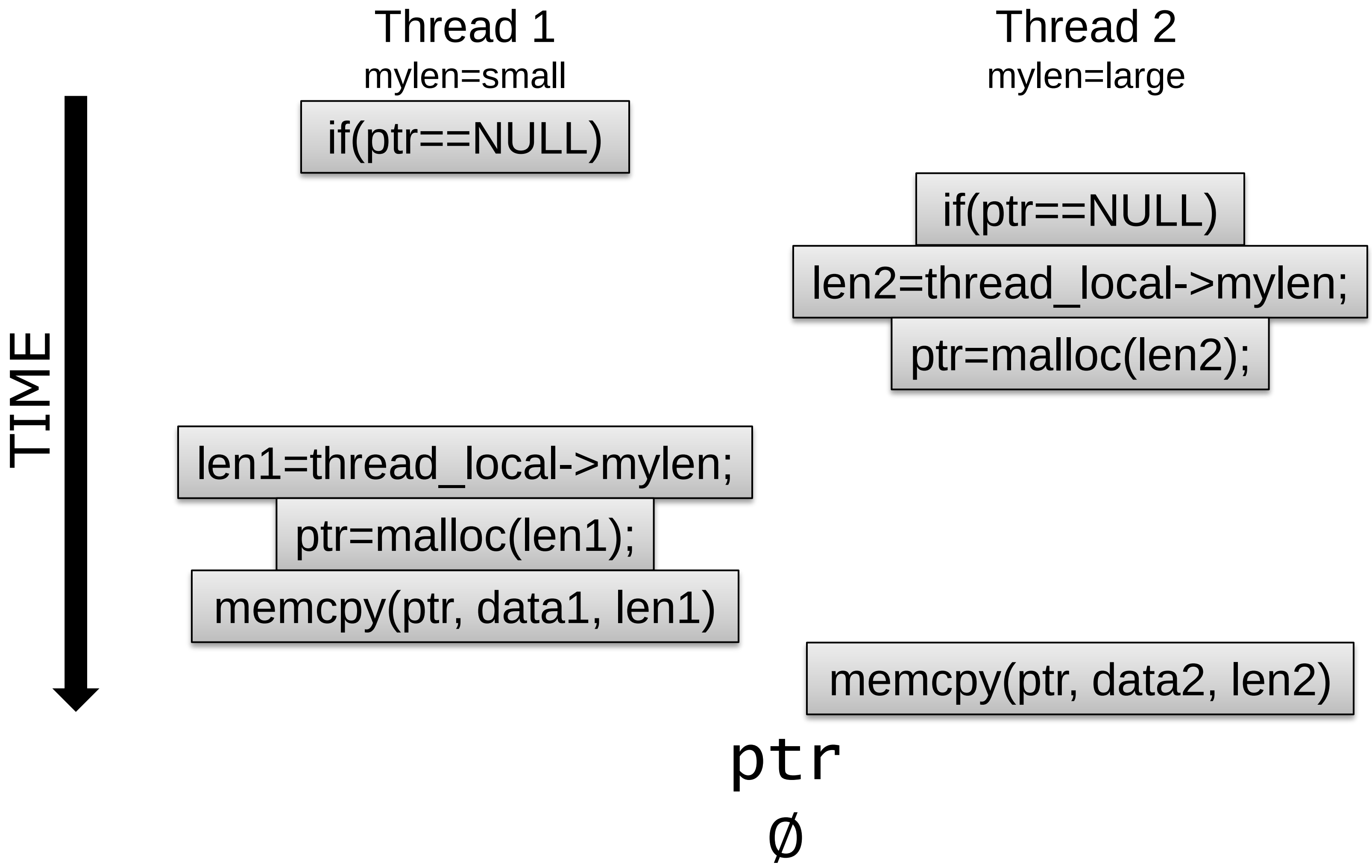
Forgot User ID or Password?

Take the fast track

See if you're pre-qualified for a credit card in 60 seconds or less

```
ptr
 Ø
```

Example of a Modern Bug



Thread 1

mylen=small

Thread 2

mylen=large

TIME

if(ptr==NULL)

if(ptr==NULL)

len2=thread_local->mylen;

ptr=malloc(len2);

len1=thread_local->mylen;

ptr=malloc(len1);

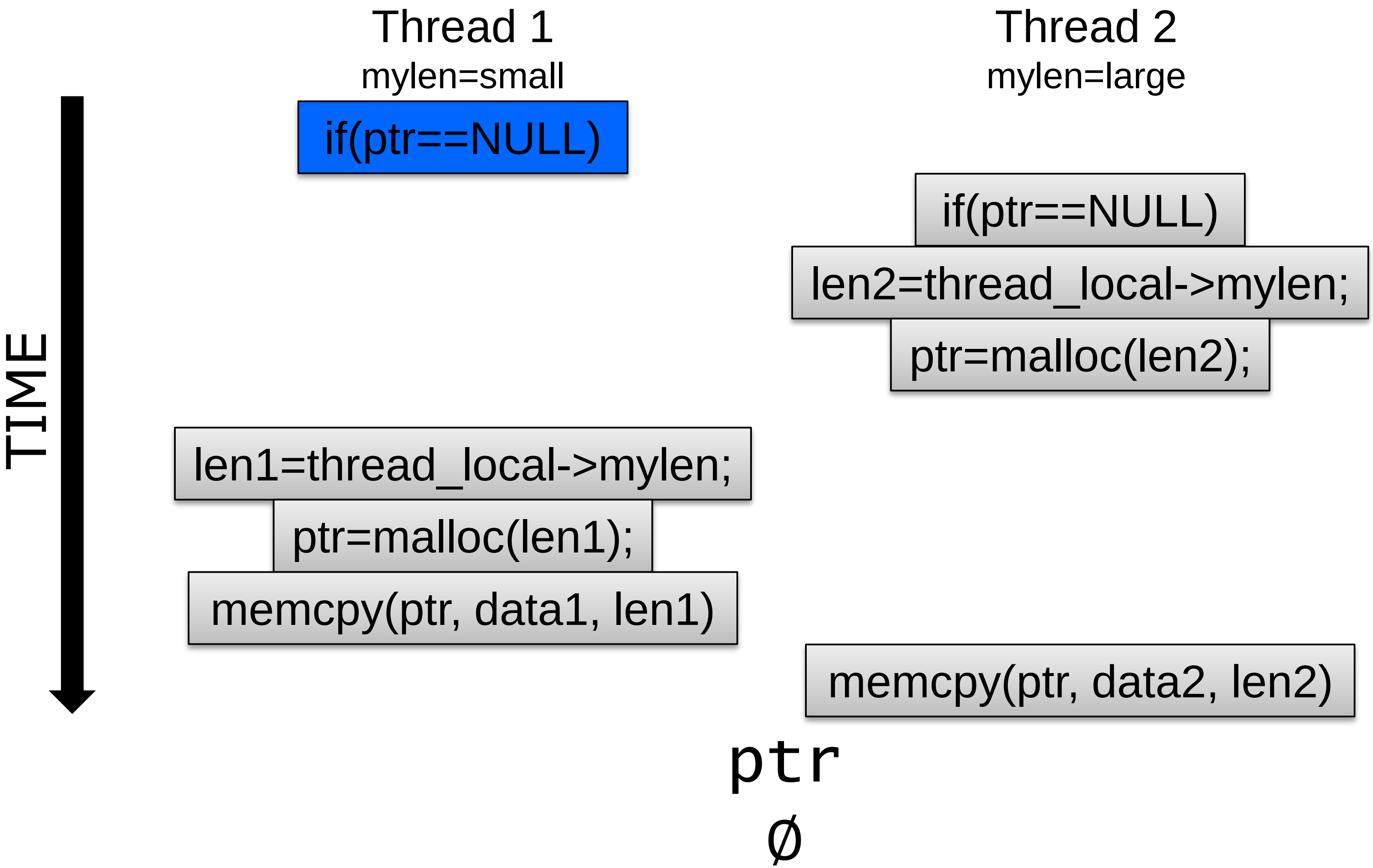
memcpy(ptr, data1, len1)

memcpy(ptr, data2, len2)

ptr

∅

Example of a Modern Bug



TIME

Thread 1

mylen=small

```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread 2

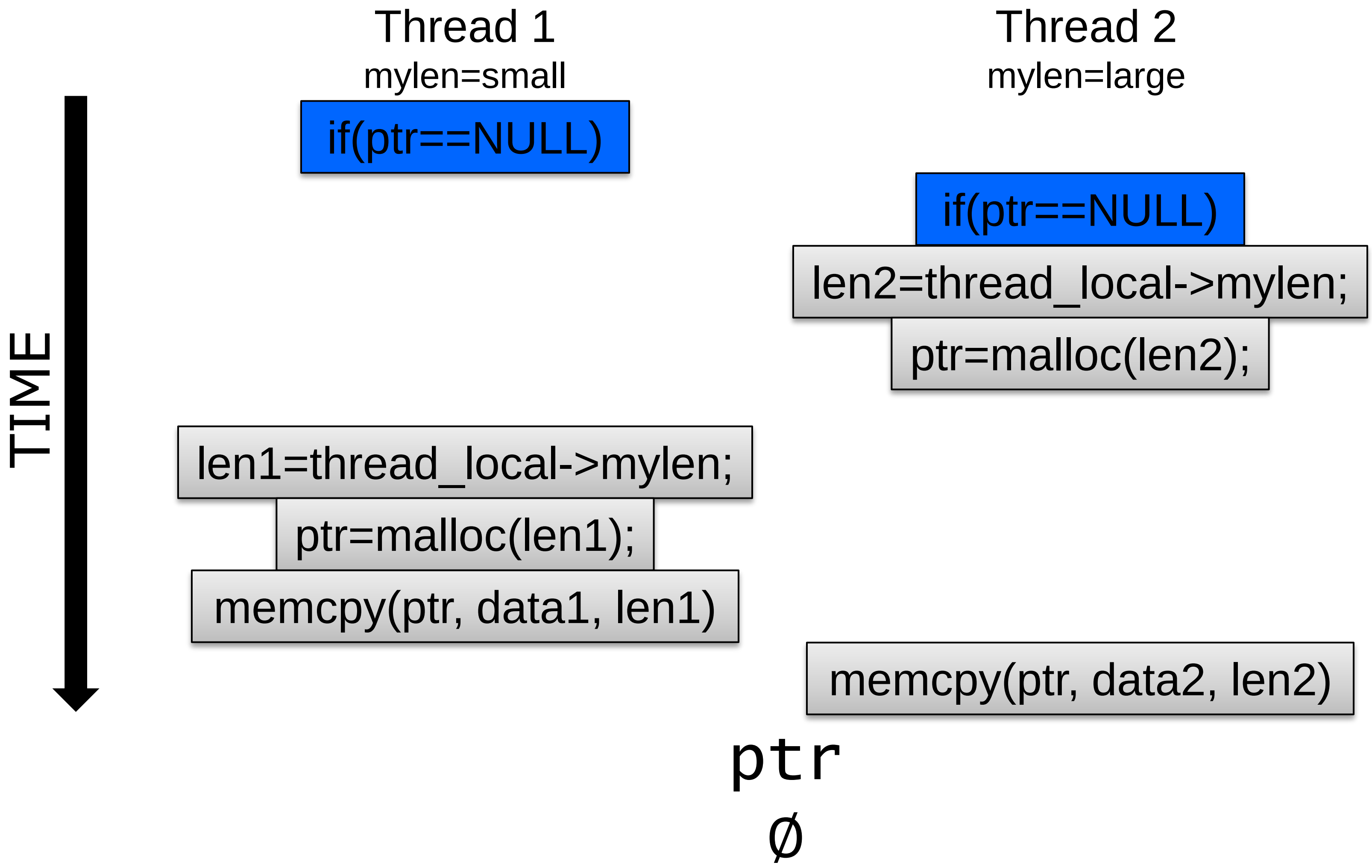
mylen=large

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

`ptr`

∅

Example of a Modern Bug



Thread 1

mylen=small

```
if(ptr==NULL)

len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

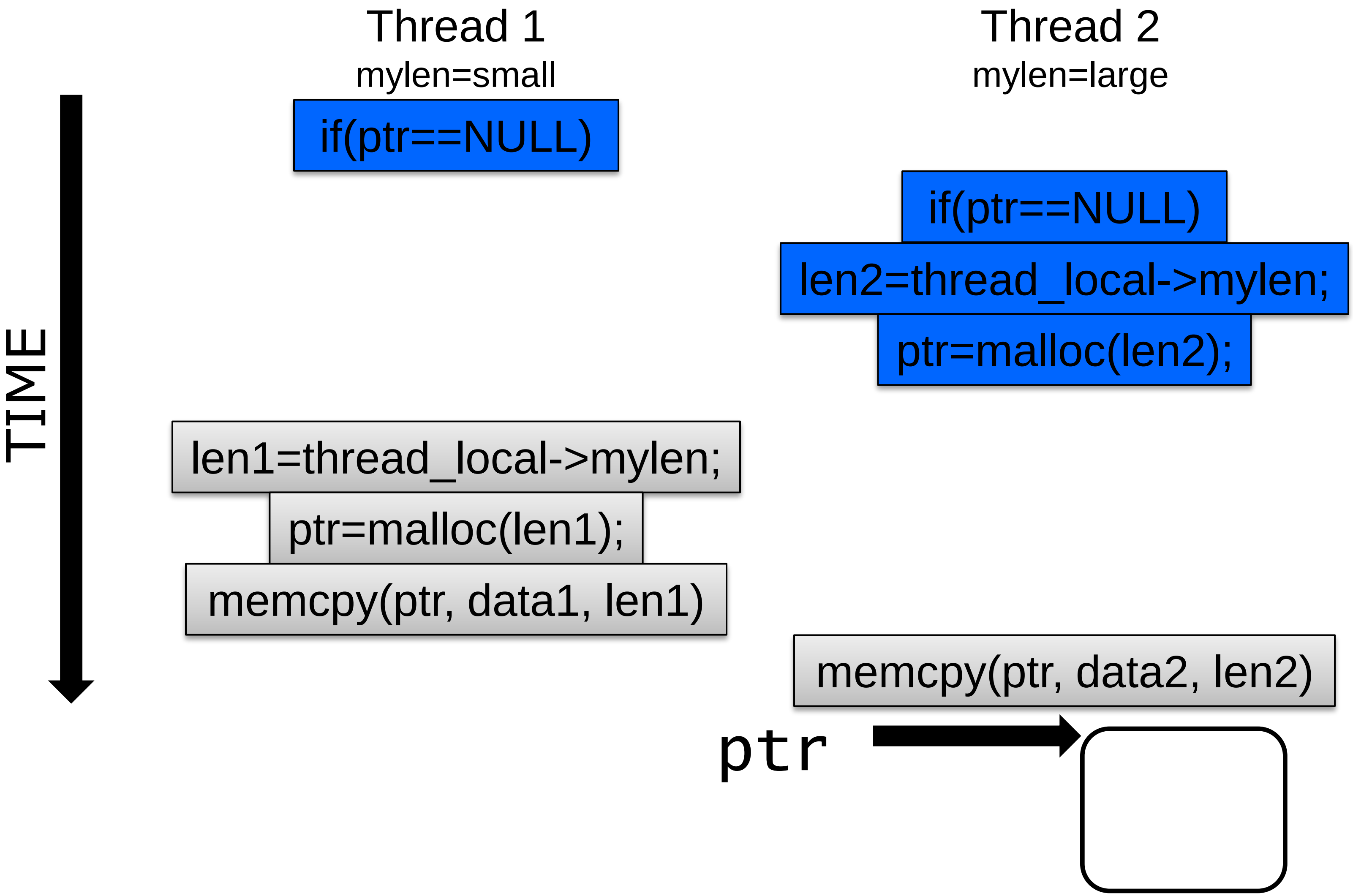
Thread 2

mylen=large

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);

memcpy(ptr, data2, len2)
```

```
ptr
∅
```

Example of a Modern Bug



Thread 1

mylen=small

Thread 2

mylen=large

TIME

```
if(ptr==NULL)
```

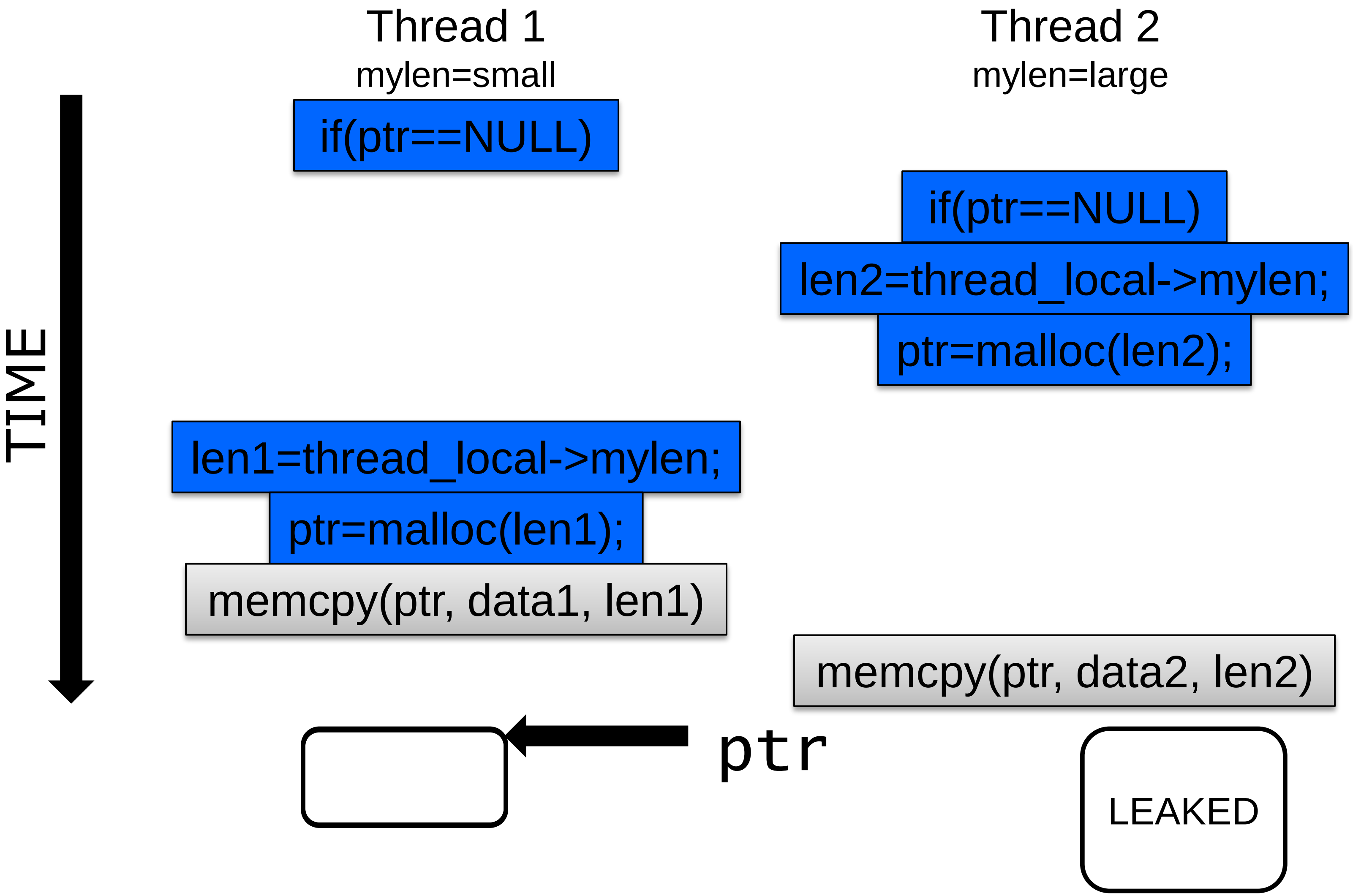
```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
```

```
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

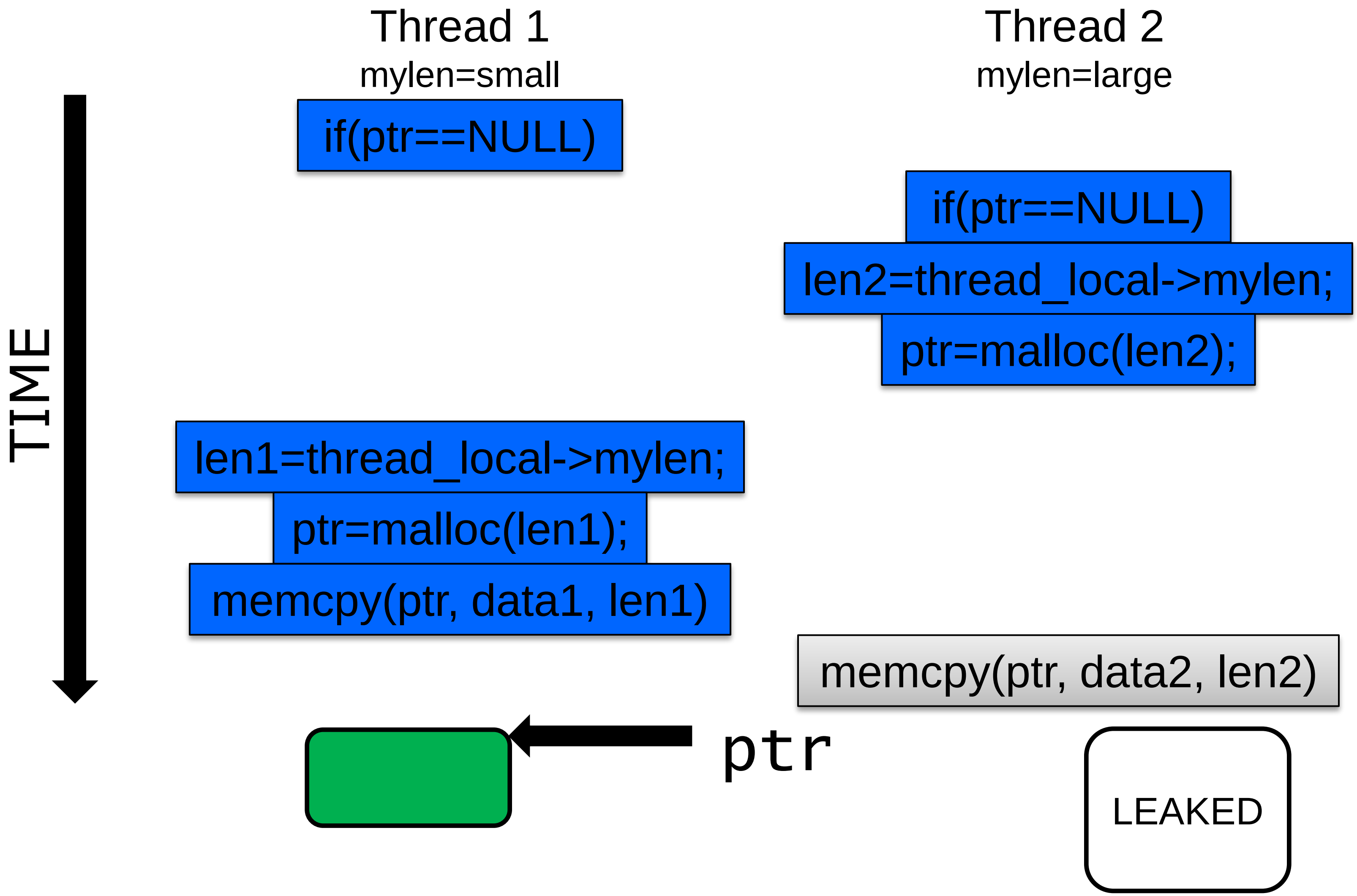
```
memcpy(ptr, data2, len2)
```

ptr

Example of a Modern Bug



Example of a Modern Bug



Thread 1
mylen=small

Thread 2
mylen=large

TIME

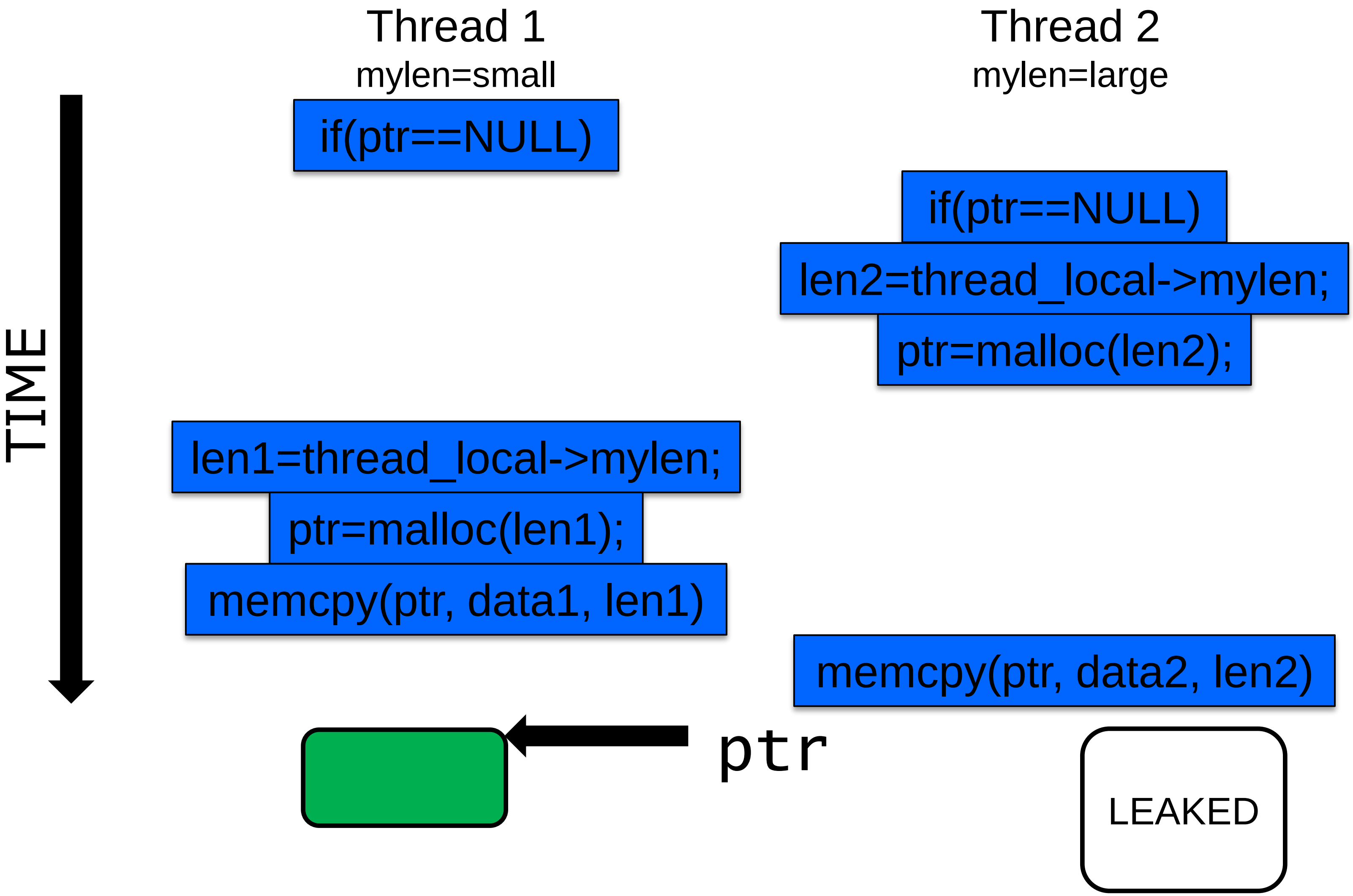
```
if(ptr==NULL)
```

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
```

```
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

```
memcpy(ptr, data2, len2)
```

Example of a Modern Bug



TIME

Thread 1

mylen=small

if(ptr==NULL)

len1=thread_local->mylen;

ptr=malloc(len1);

memcpy(ptr, data1, len1)

Thread 2

mylen=large

if(ptr==NULL)

len2=thread_local->mylen;

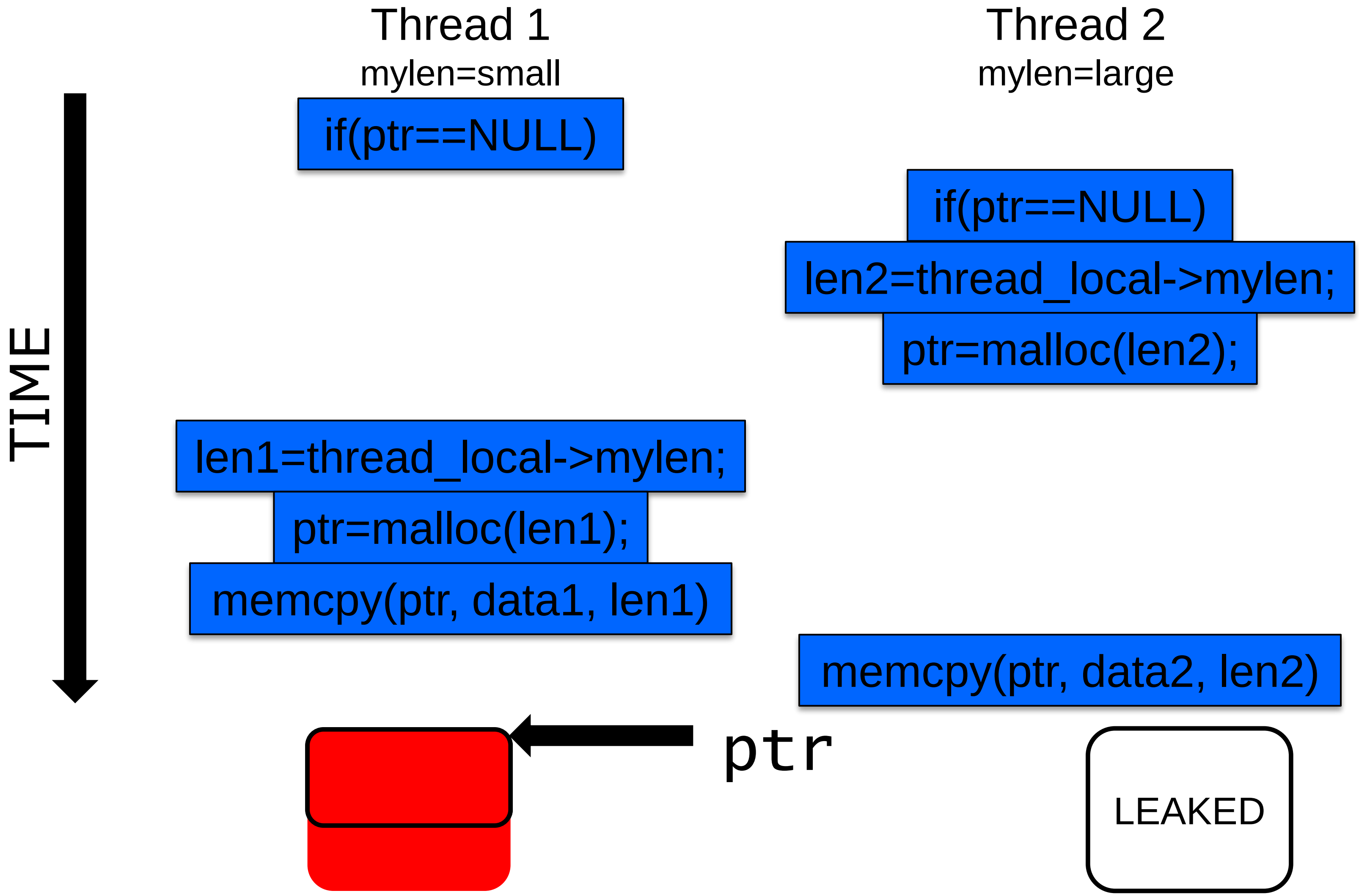
ptr=malloc(len2);

memcpy(ptr, data2, len2)

ptr

LEAKED

Example of a Modern Bug



Thread 1

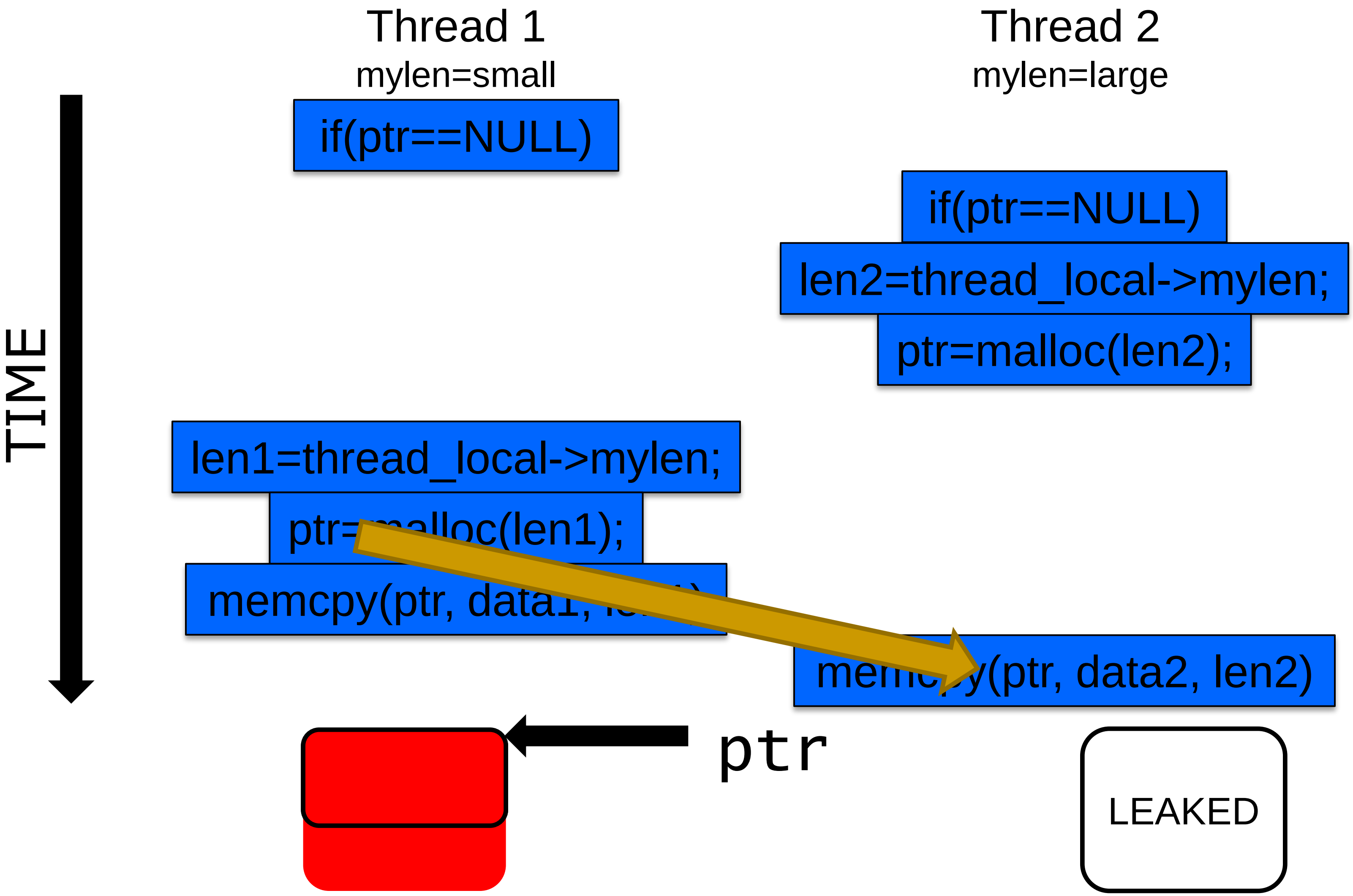
mylen=small

Thread 2

mylen=large

```
Thread 1: if(ptr==NULL)
Thread 2: if(ptr==NULL)
Thread 2: len2=thread_local->mylen;
Thread 2: ptr=malloc(len2);
Thread 1: len1=thread_local->mylen;
Thread 1: ptr=malloc(len1);
Thread 1: memcpy(ptr, data1, len1)
Thread 2: memcpy(ptr, data2, len2)
```

Example of a Modern Bug



Thread 1

mylen=small

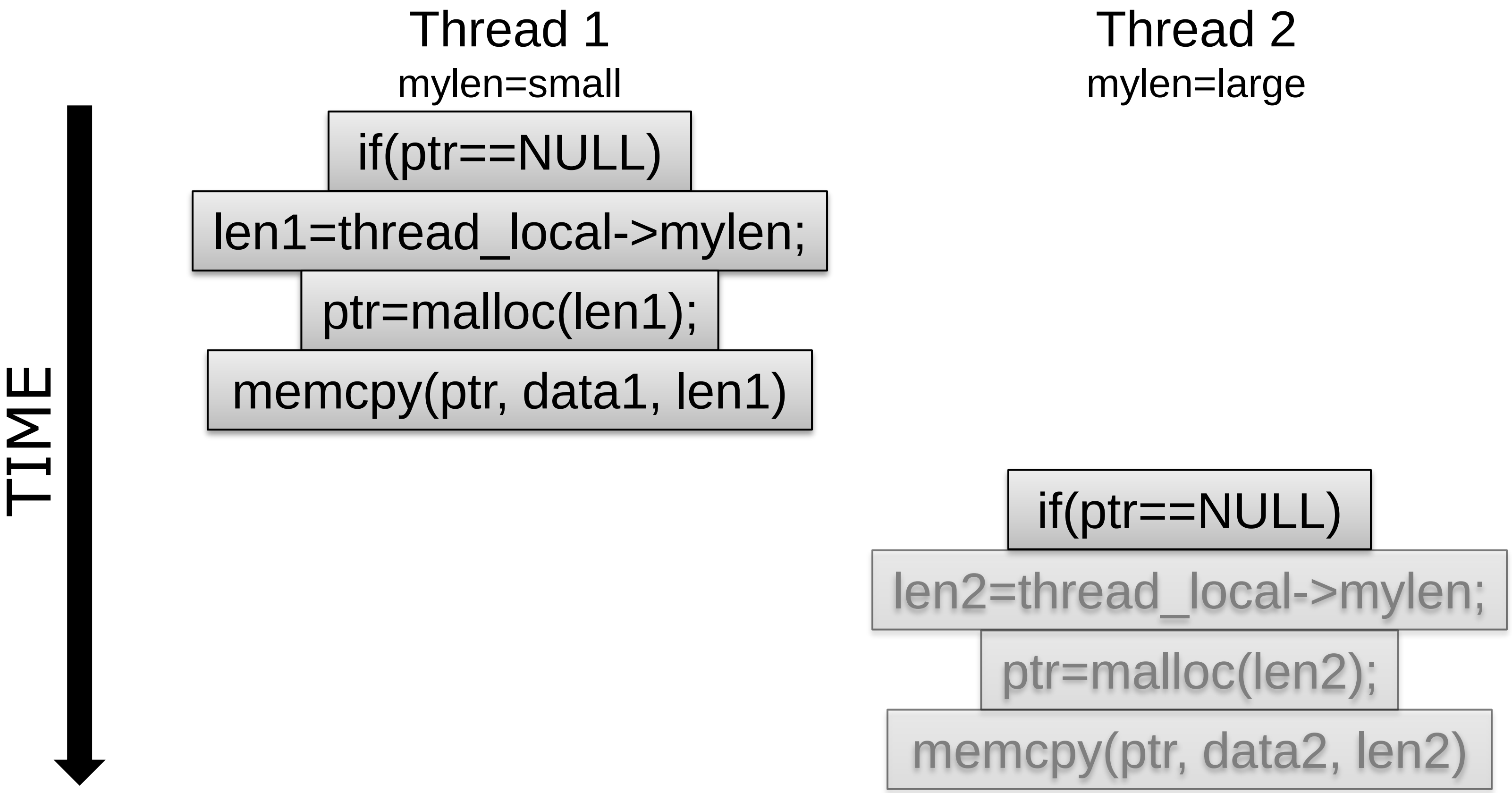
```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread 2

mylen=large

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Data Race Detection



TIME

Thread 1

mylen=small

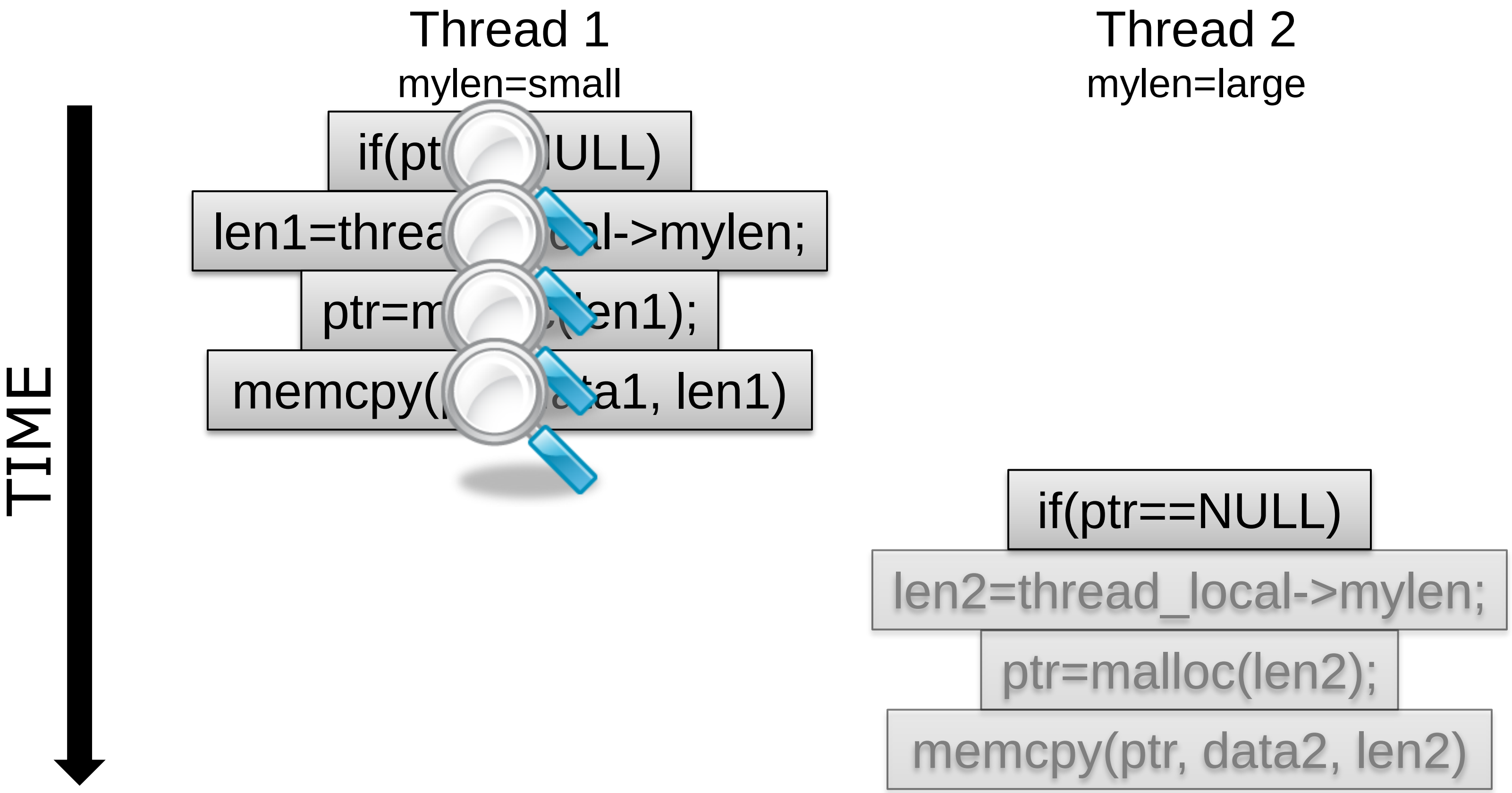
```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread 2

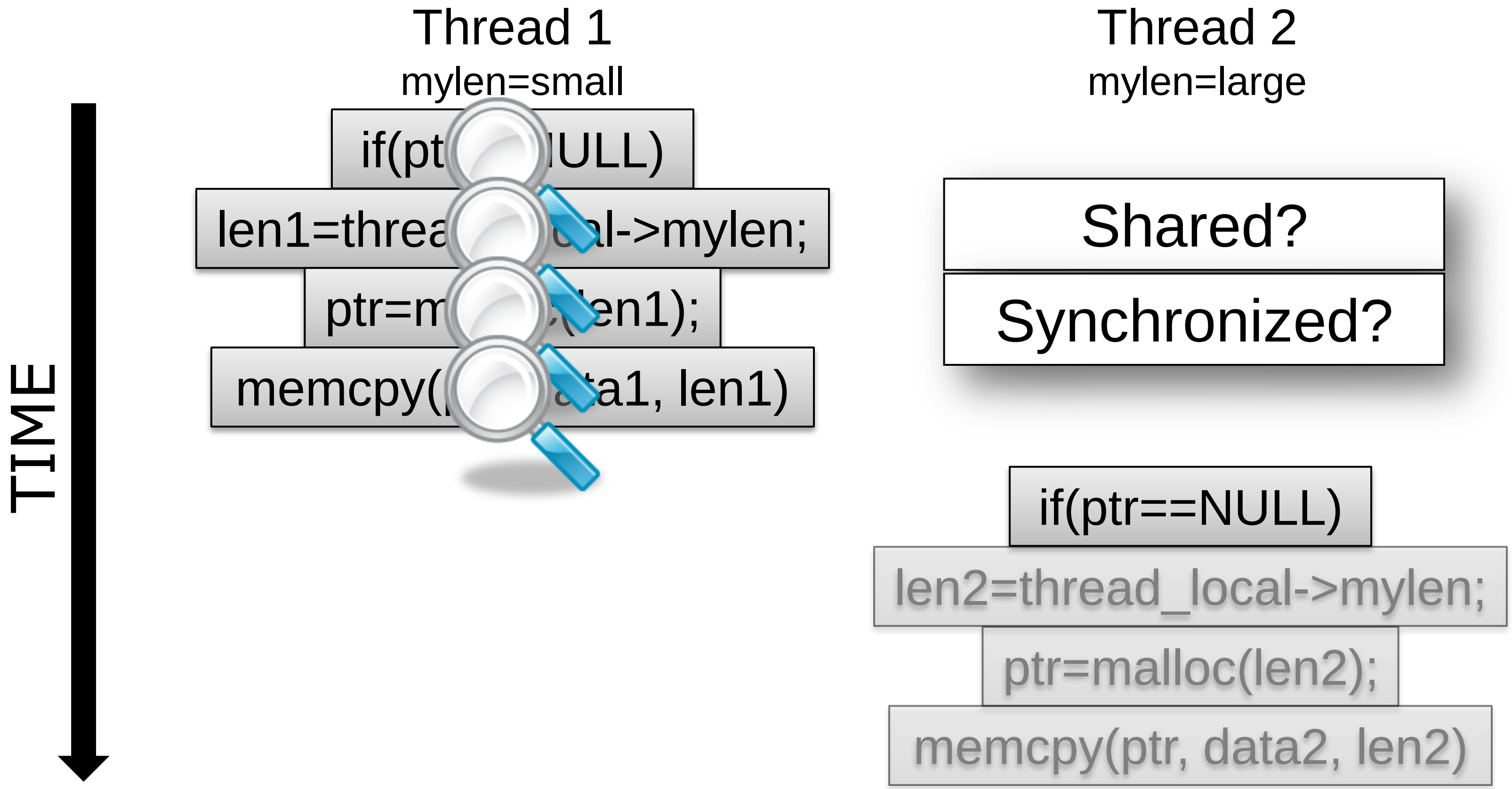
mylen=large

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Data Race Detection



Data Race Detection



Thread 1

mylen=small

```
if(pt      ULL)
len1=threa      al->mylen;
ptr=m      (len1);
memcpy(      ta1, len1)
```

Thread 2

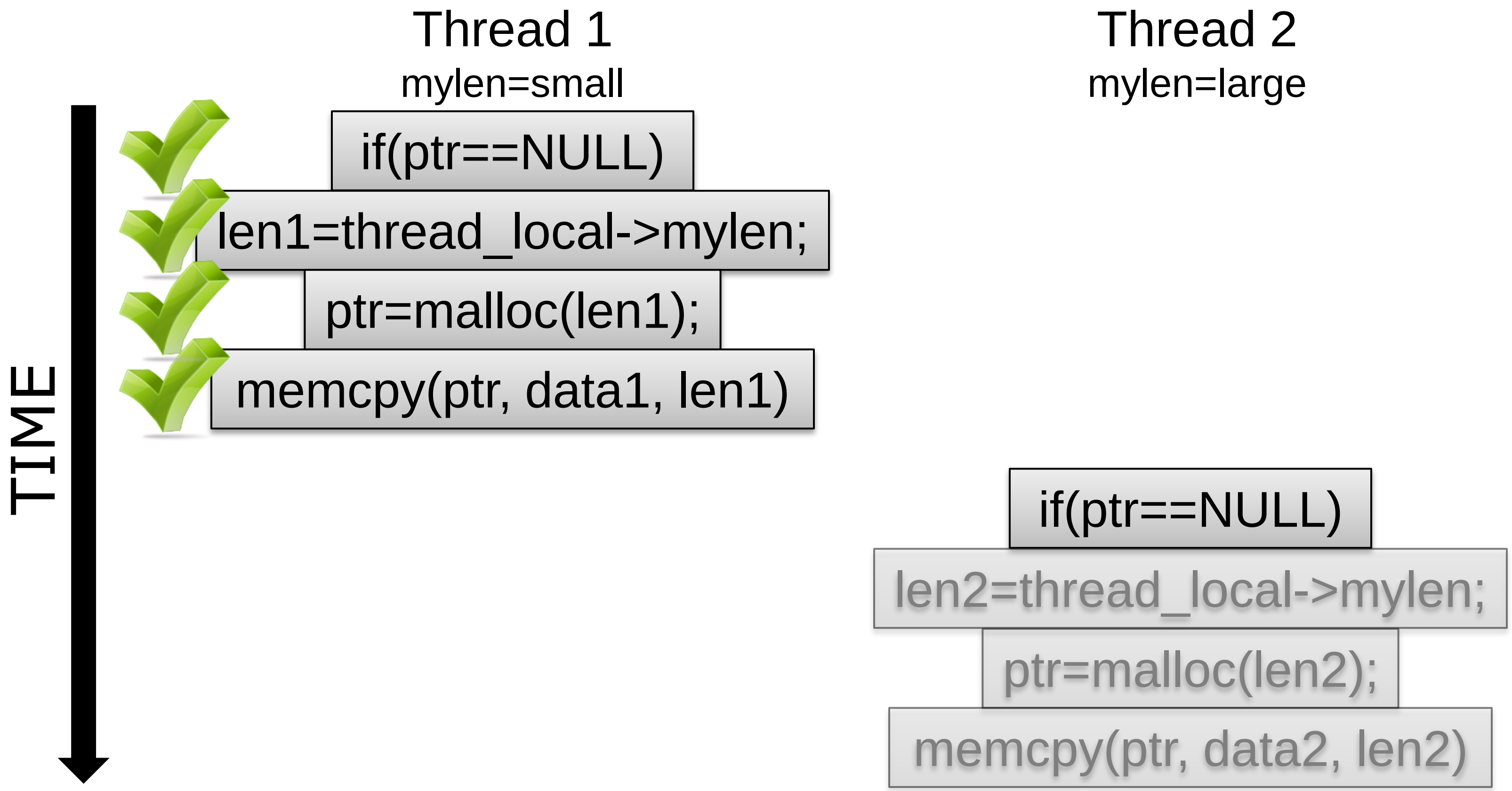
mylen=large

Shared?

Synchronized?

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Data Race Detection



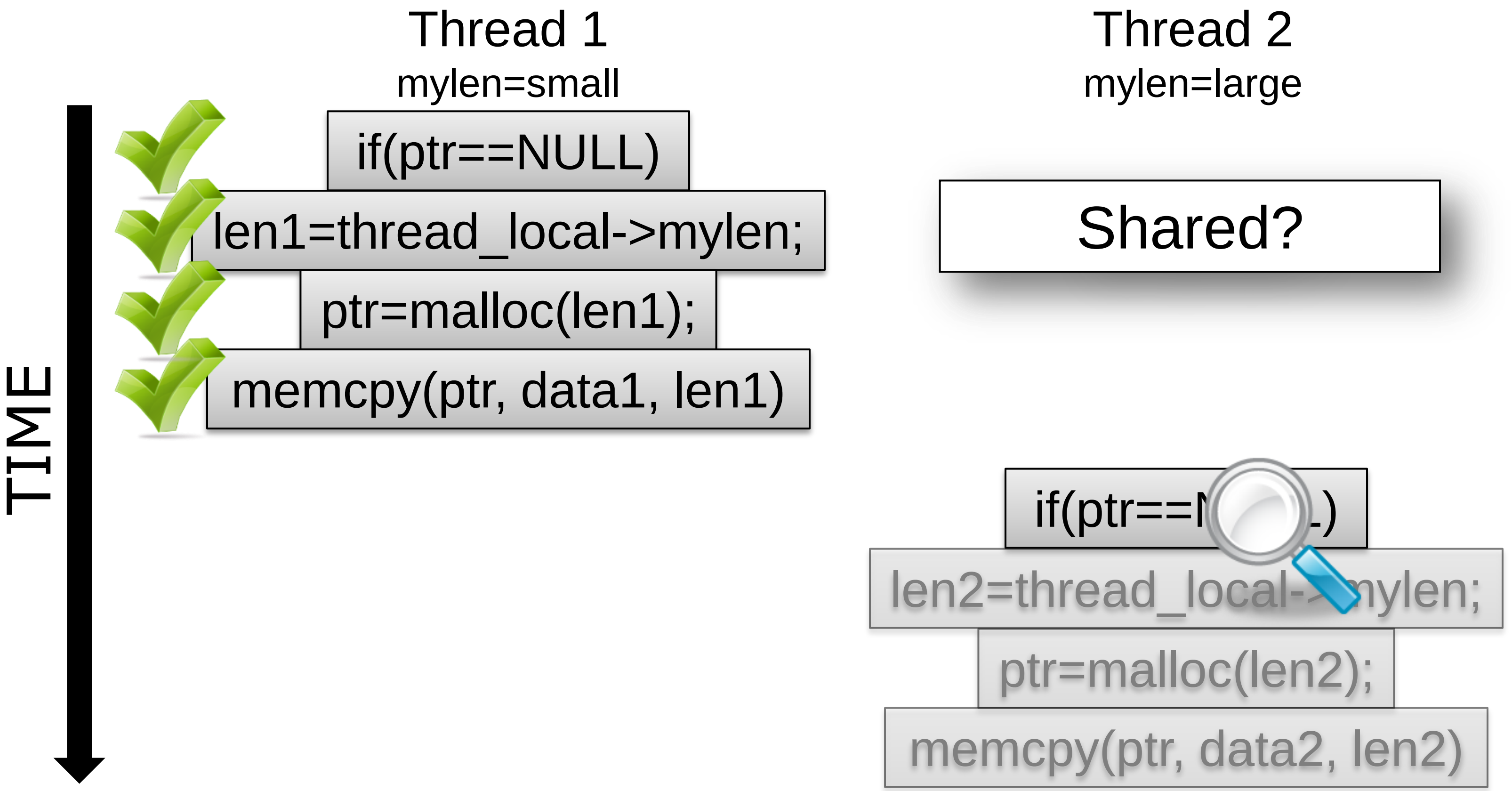
Thread 1
mylen=small

```
✓ if(ptr==NULL)
✓ len1=thread_local->mylen;
✓ ptr=malloc(len1);
✓ memcpy(ptr, data1, len1)
```

Thread 2
mylen=large

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Example of Data Race Detection



Thread 1

mylen=small

✓ if(ptr==NULL)

✓ len1=thread_local->mylen;

✓ ptr=malloc(len1);

✓ memcpy(ptr, data1, len1)

Thread 2

mylen=large

Shared?

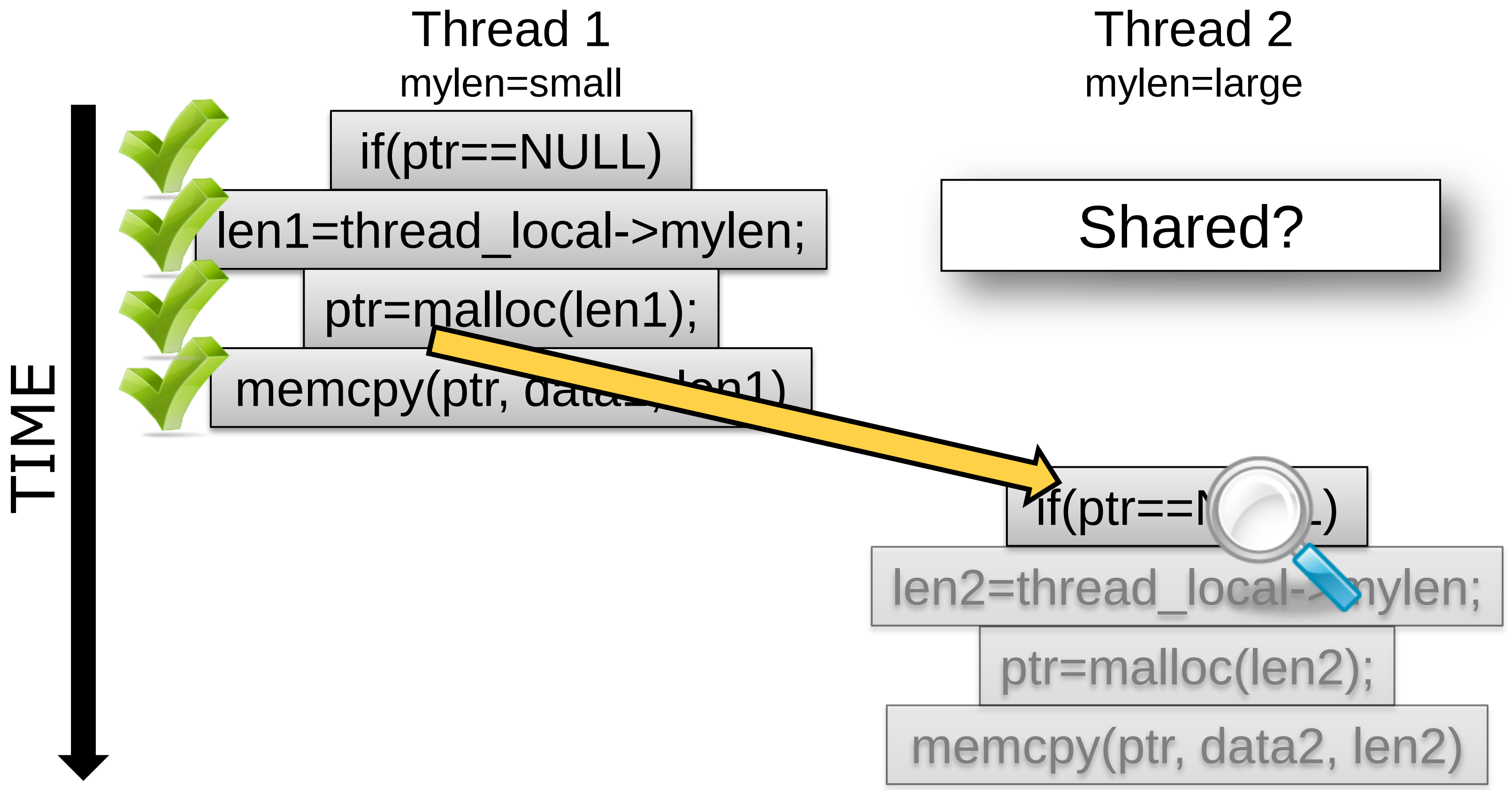
if(ptr==NULL)

len2=thread_local->mylen;

ptr=malloc(len2);

memcpy(ptr, data2, len2)

Example of Data Race Detection



Thread 1

mylen=small

```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

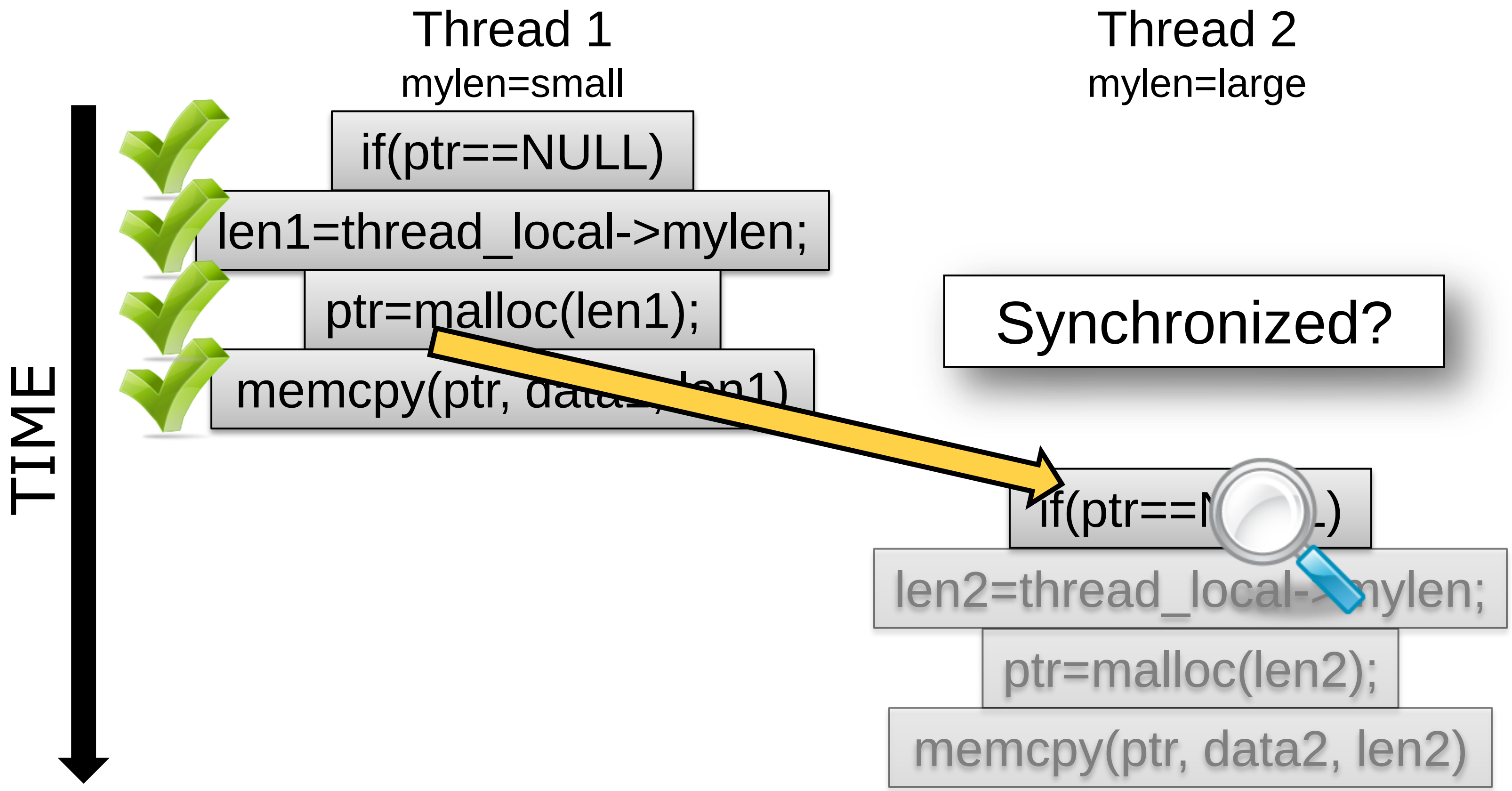
Thread 2

mylen=large

Shared?

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Example of Data Race Detection



TIME

Thread 1

mylen=small

```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

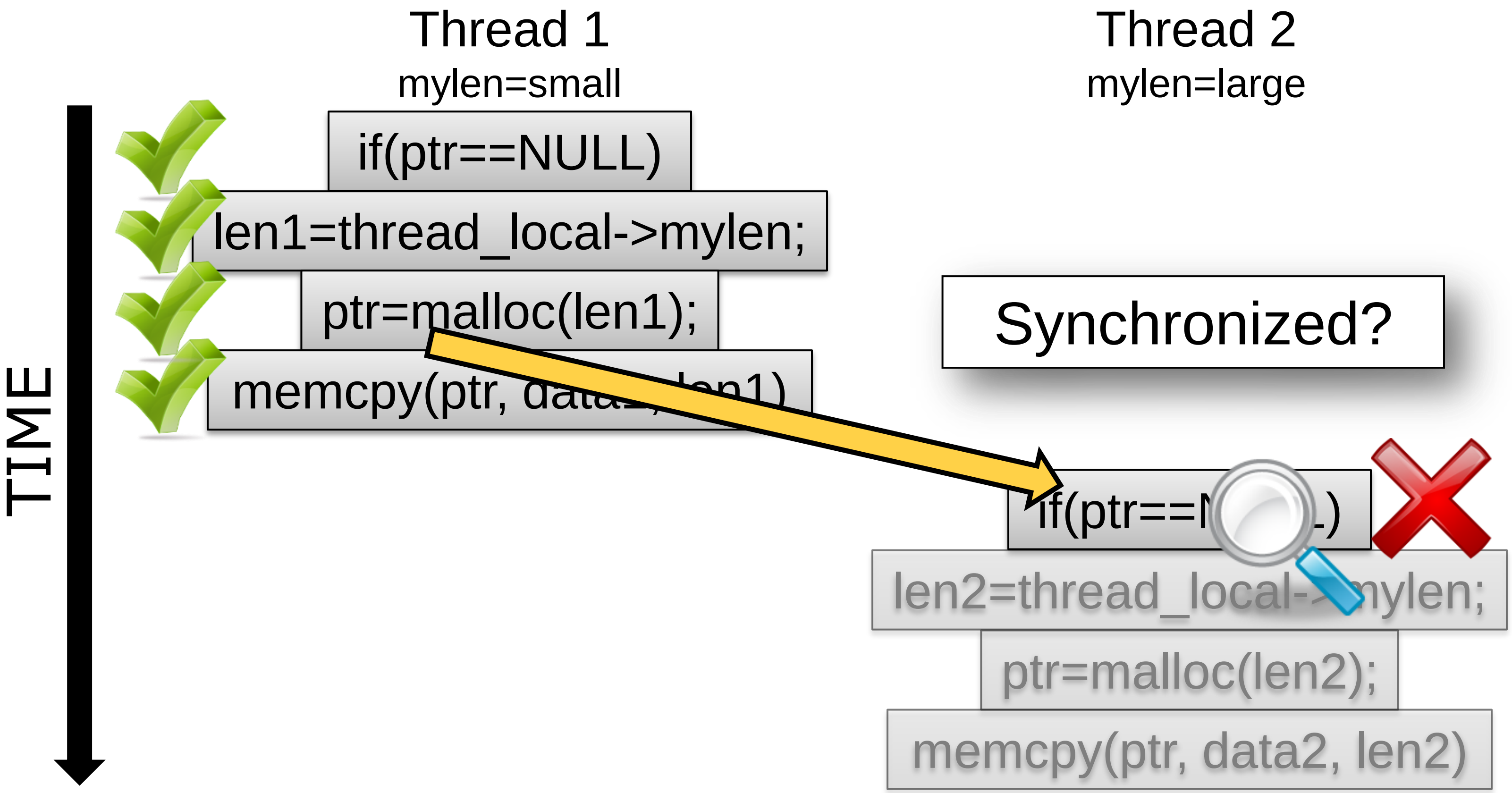
Thread 2

mylen=large

Synchronized?

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Example of Data Race Detection



Thread 1

mylen=small

if(ptr==NULL)

len1=thread_local->mylen;

ptr=malloc(len1);

memcpy(ptr, data1, len1)

Thread 2

mylen=large

Synchronized?

if(ptr==NULL)

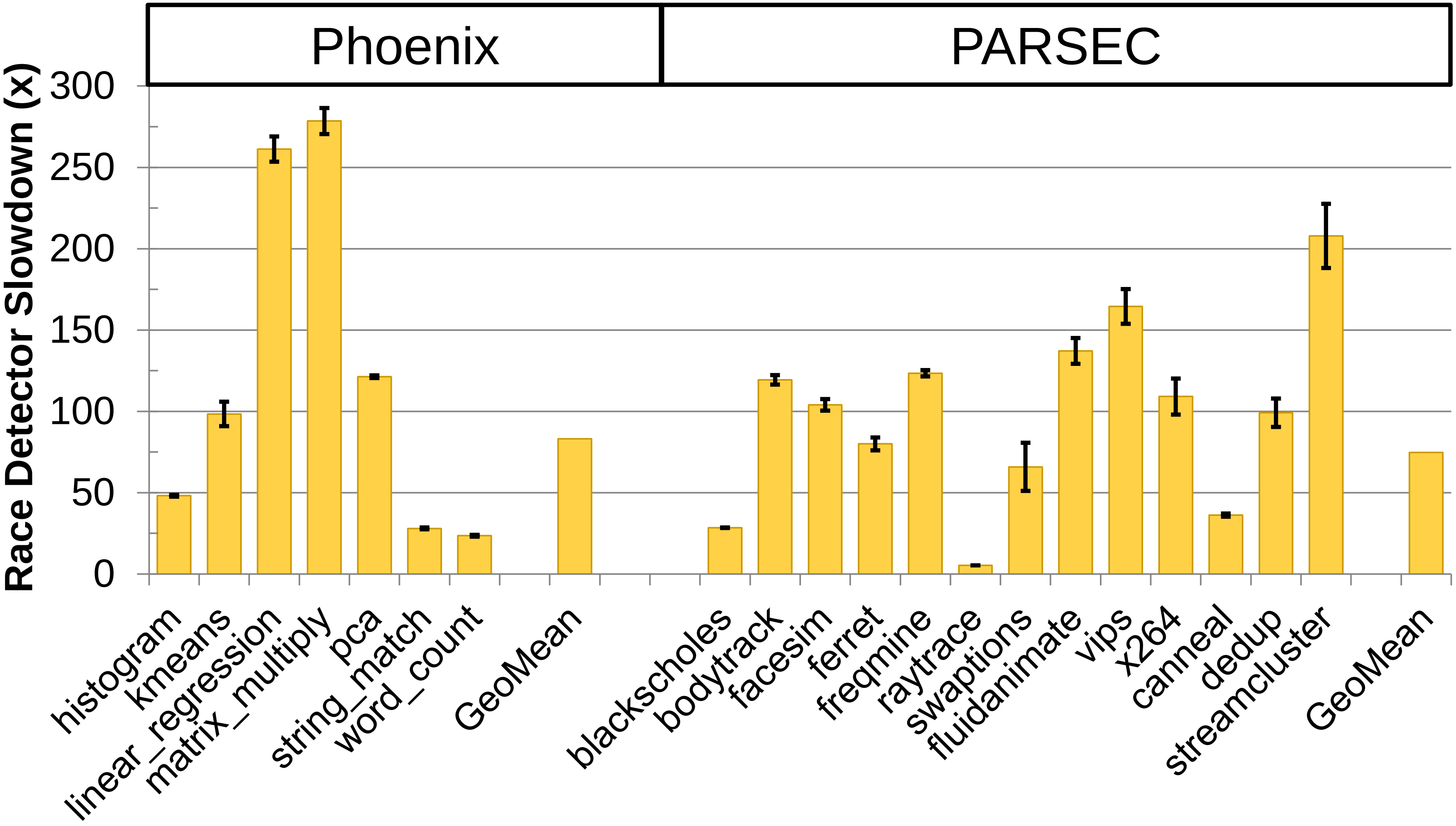
len2=thread_local->mylen;

ptr=malloc(len2);

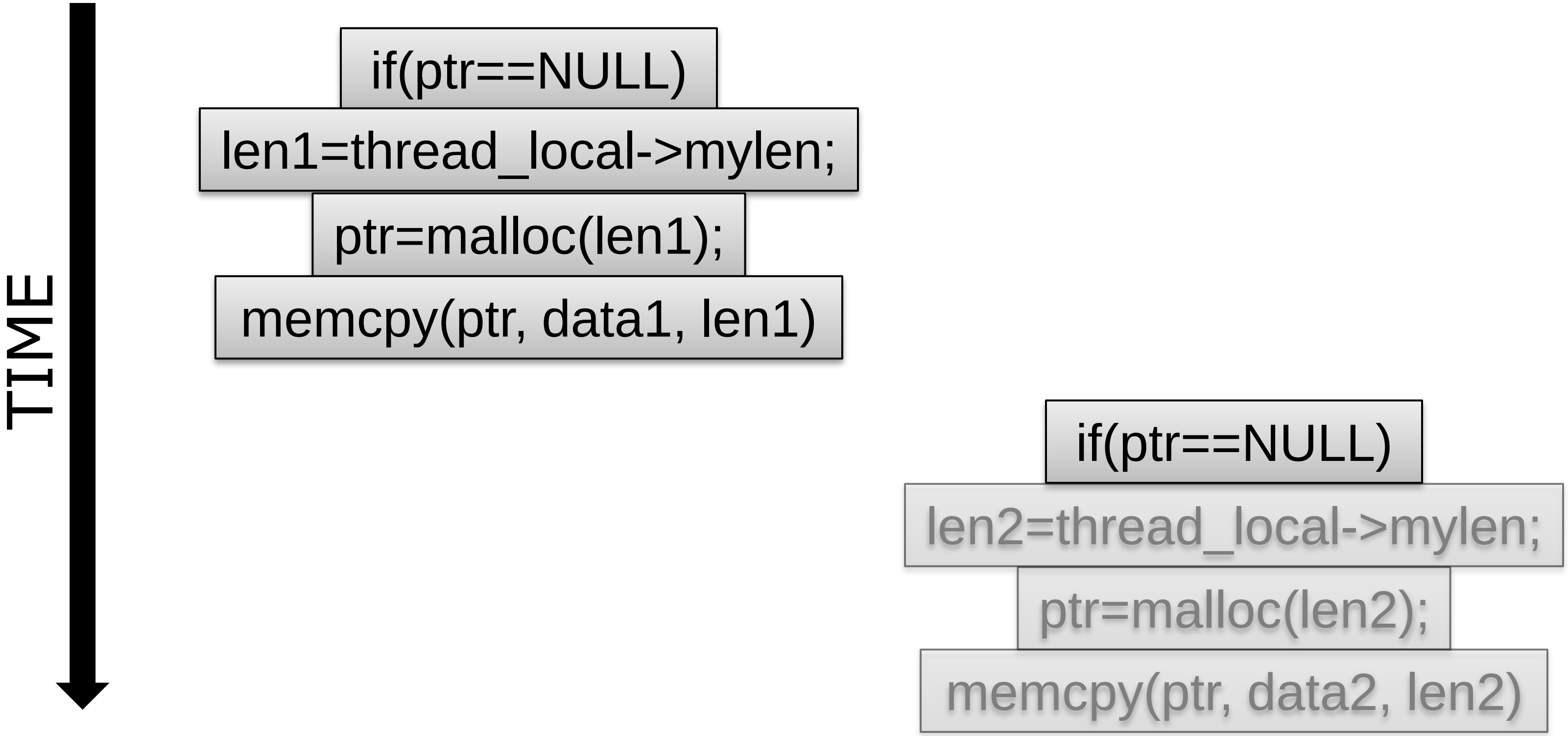
memcpy(ptr, data2, len2)

TIME

Data Race Detection is Slow



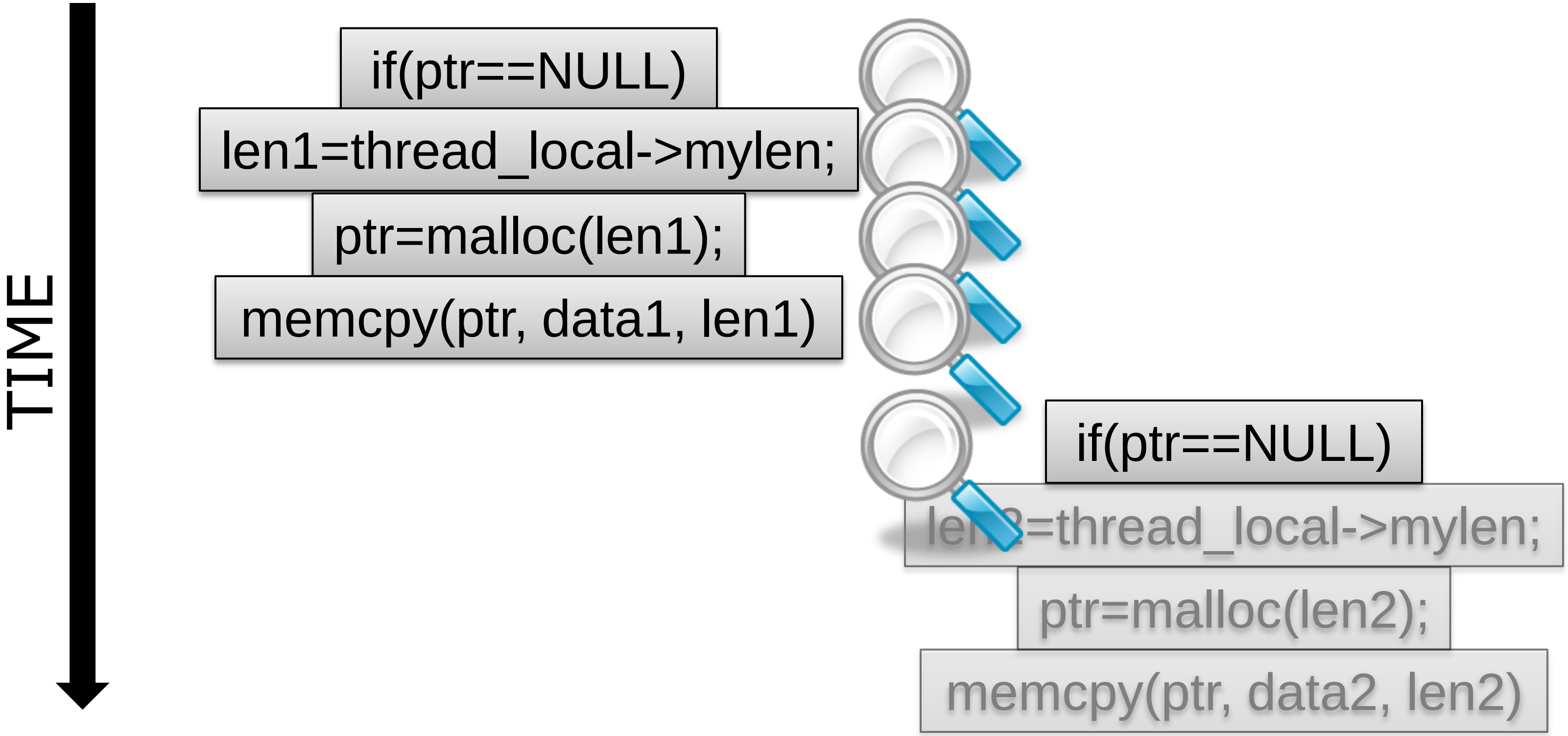
Inter-thread Sharing is What's Important



```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

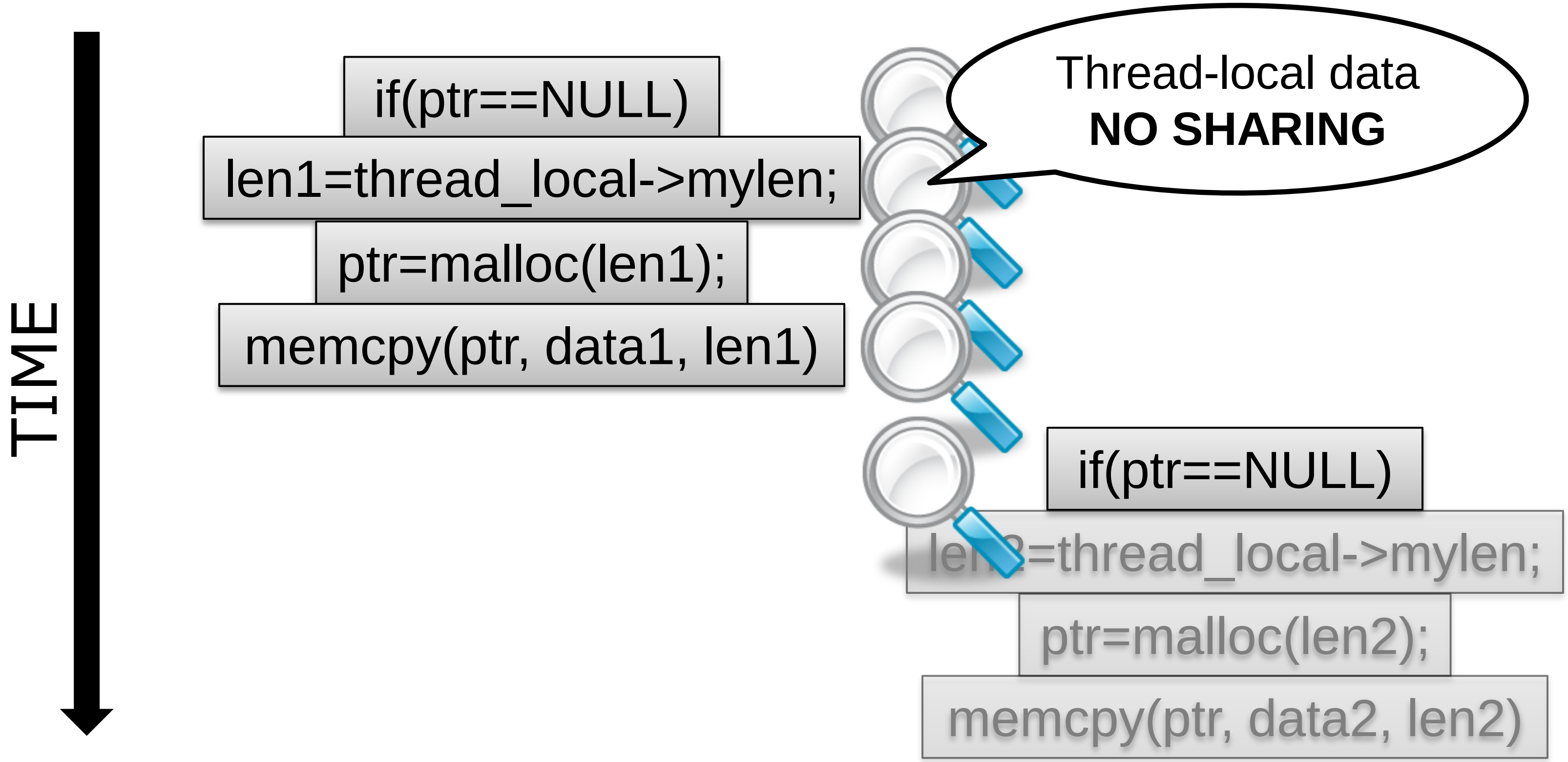
Inter-thread Sharing is What's Important



```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Inter-thread Sharing is What's Important



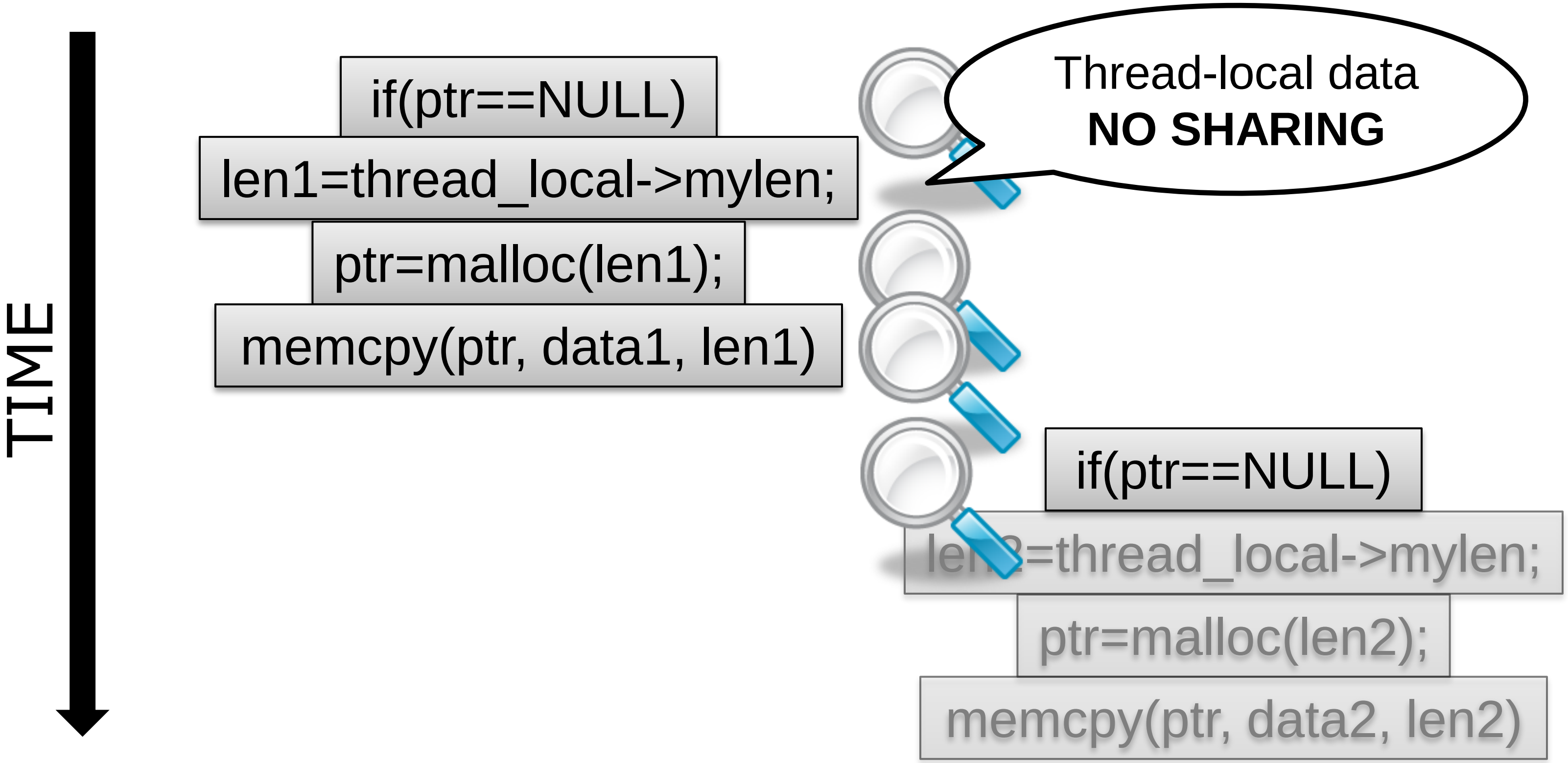
TIME

```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread-local data
NO SHARING

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Inter-thread Sharing is What's Important



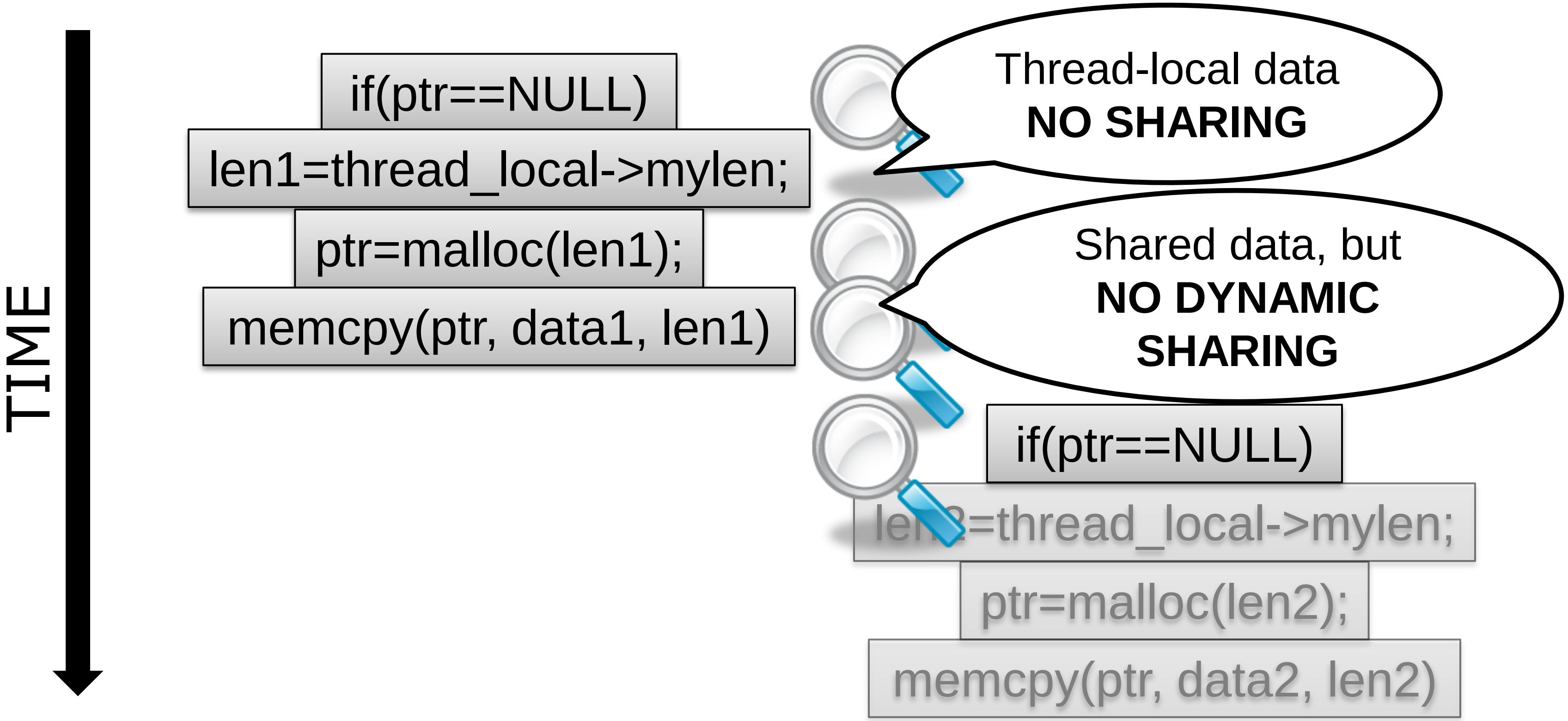
TIME

```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread-local data
NO SHARING

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Inter-thread Sharing is What's Important



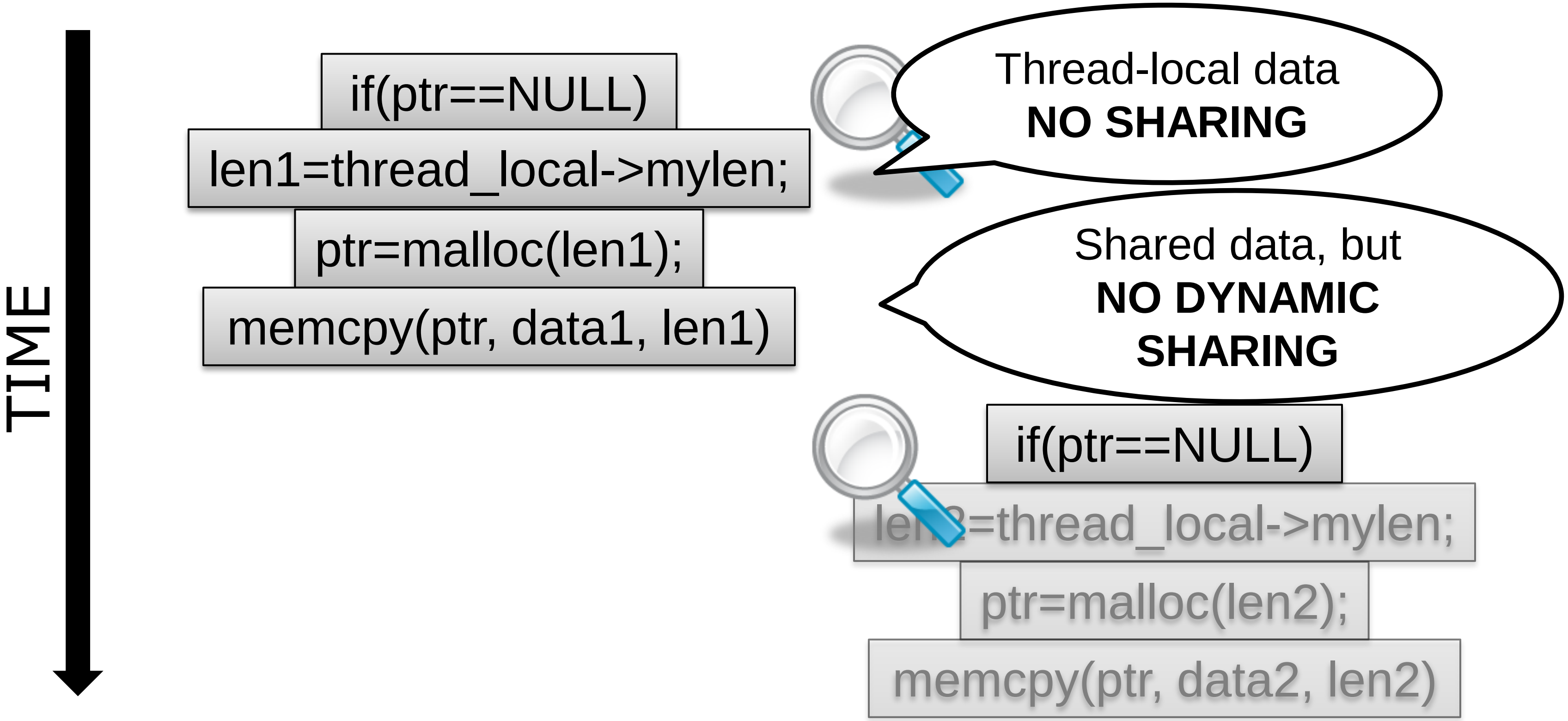
```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread-local data
NO SHARING

Shared data, but
NO DYNAMIC SHARING

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

Inter-thread Sharing is What's Important



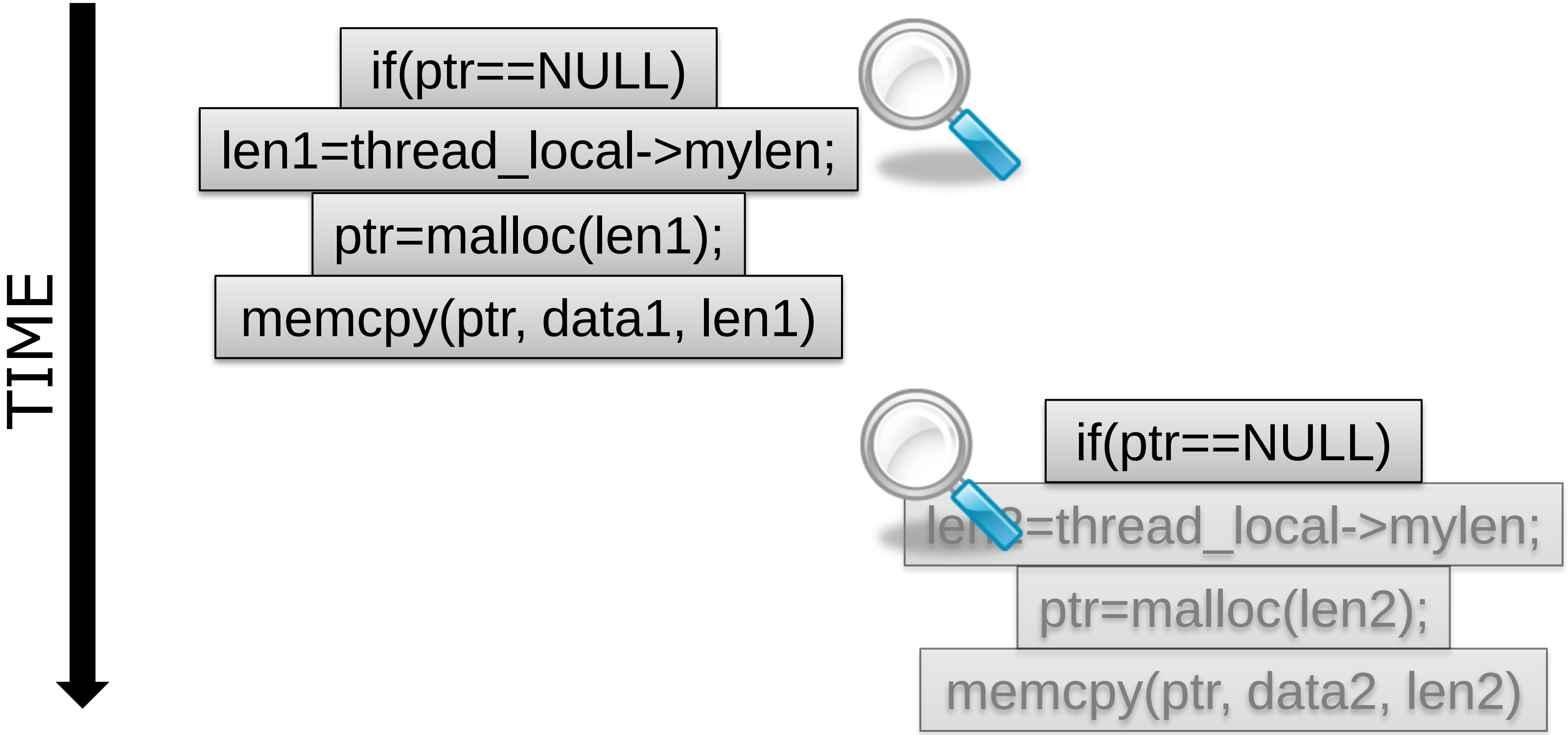
```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

Thread-local data
NO SHARING

Shared data, but
NO DYNAMIC SHARING

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

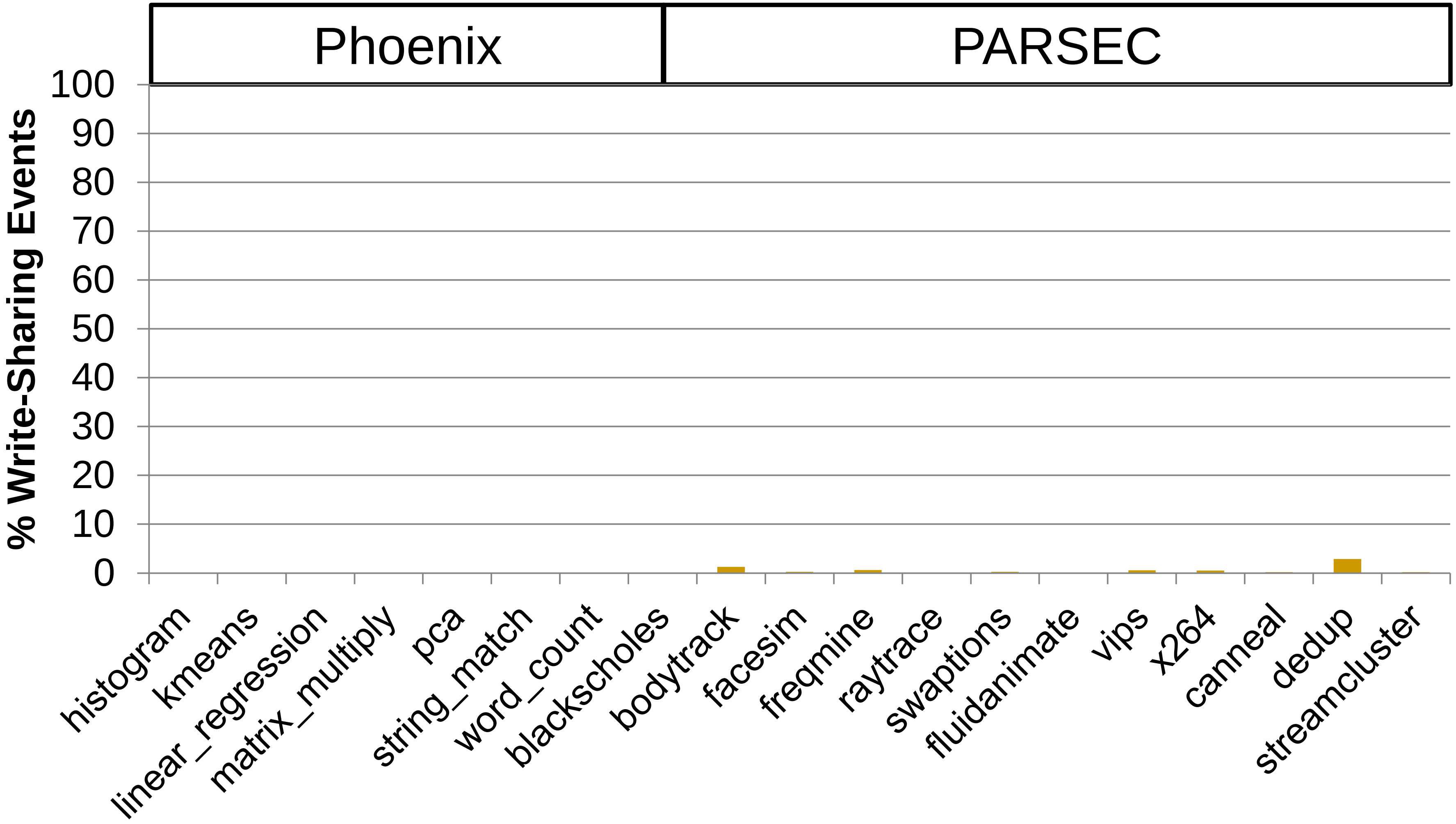
Inter-thread Sharing is What’s Important



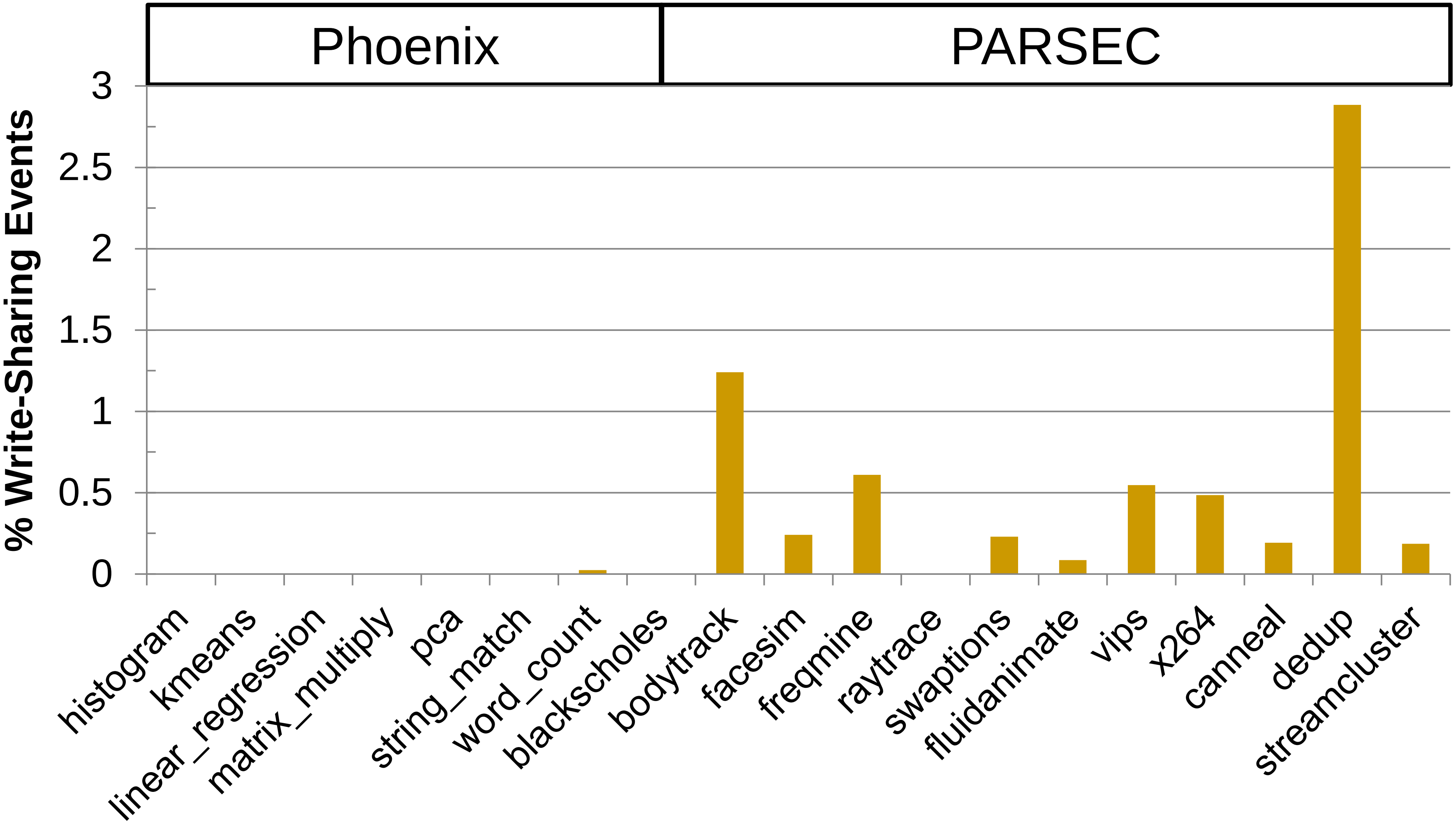
```
if(ptr==NULL)
len1=thread_local->mylen;
ptr=malloc(len1);
memcpy(ptr, data1, len1)
```

```
if(ptr==NULL)
len2=thread_local->mylen;
ptr=malloc(len2);
memcpy(ptr, data2, len2)
```

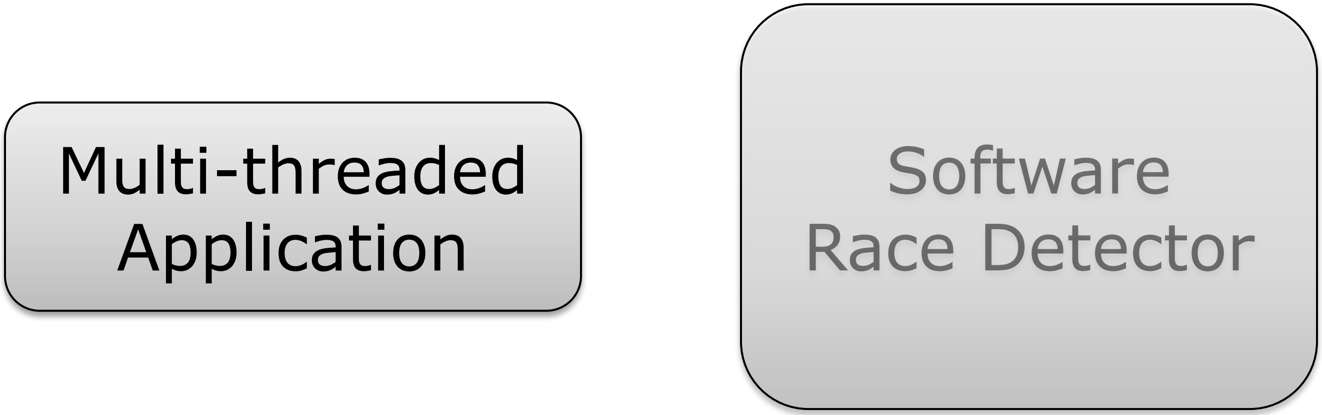
Very Little Dynamic Sharing



Very Little Dynamic Sharing



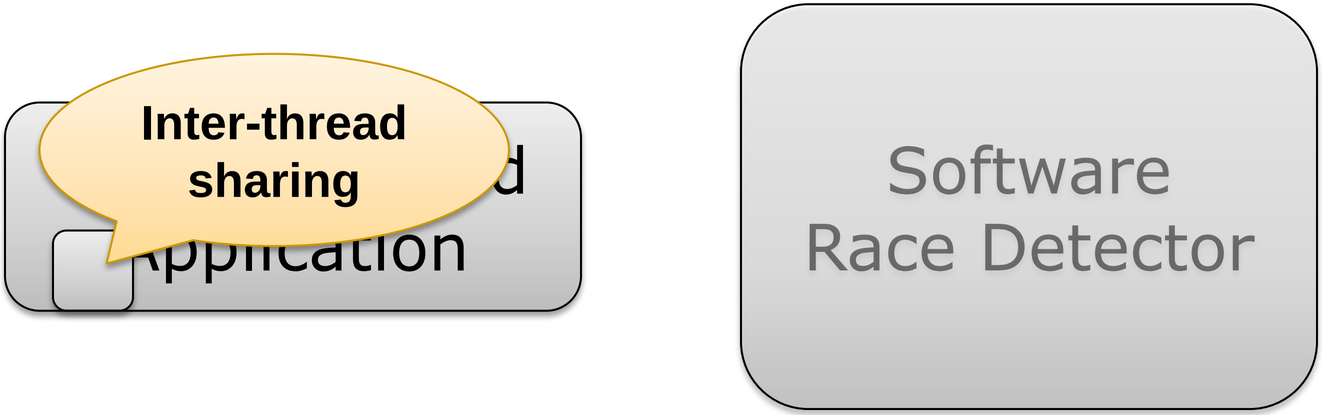
Little Sharing Means Wasted Work



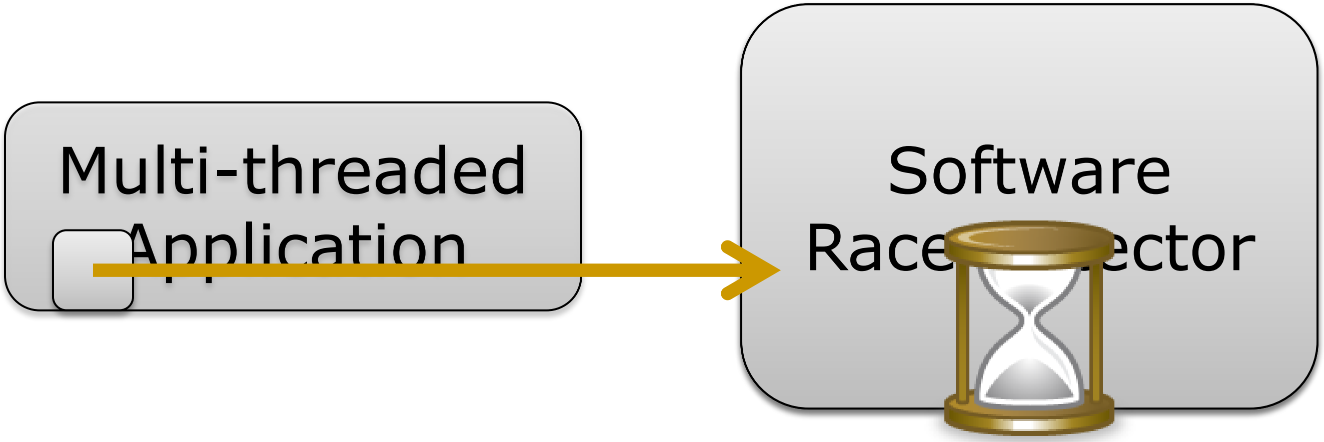
Multi-threaded Application

Software Race Detector

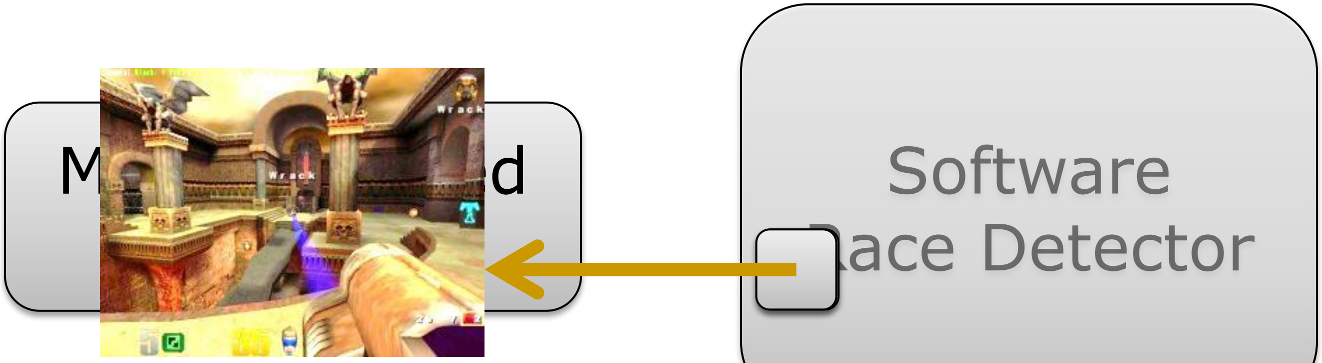
Little Sharing Means Wasted Work



Little Sharing Means Wasted Work



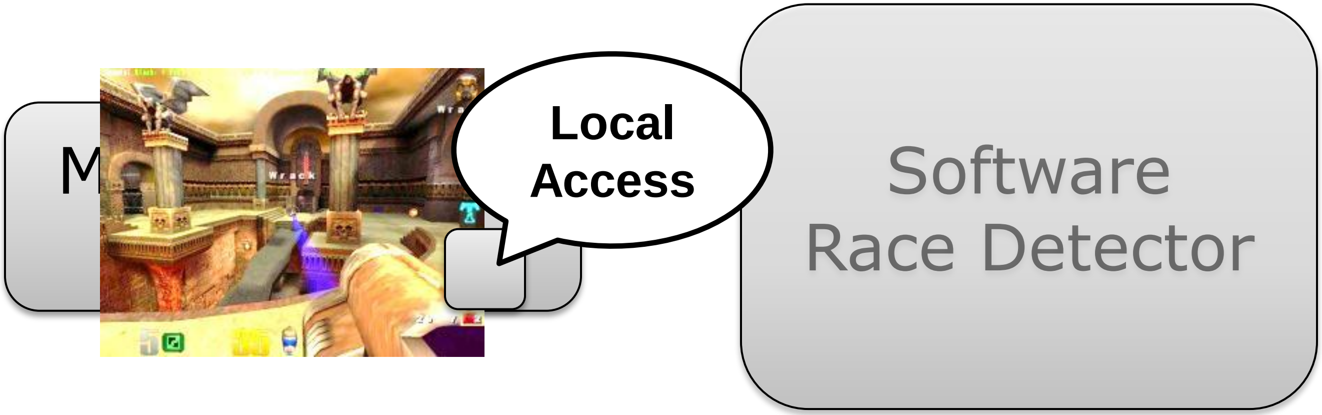
Little Sharing Means Wasted Work



M d

Software Race Detector

Little Sharing Means Wasted Work

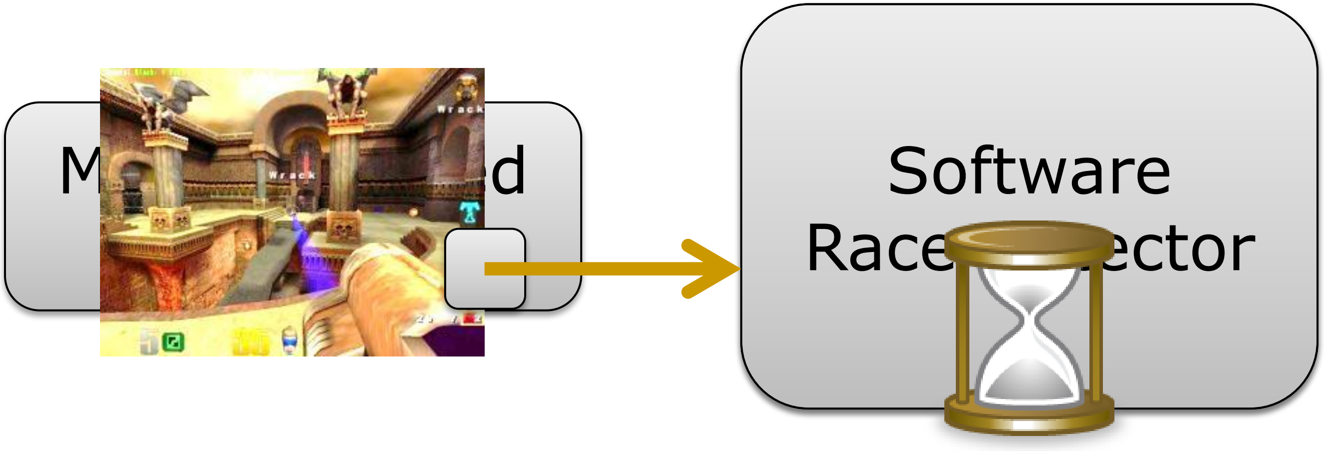


M

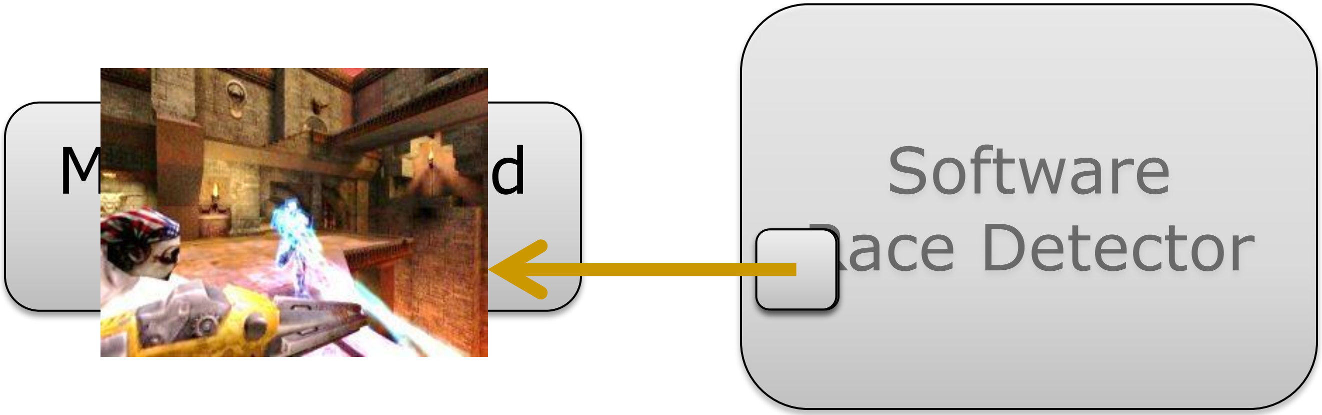
Local Access

Software Race Detector

Little Sharing Means Wasted Work



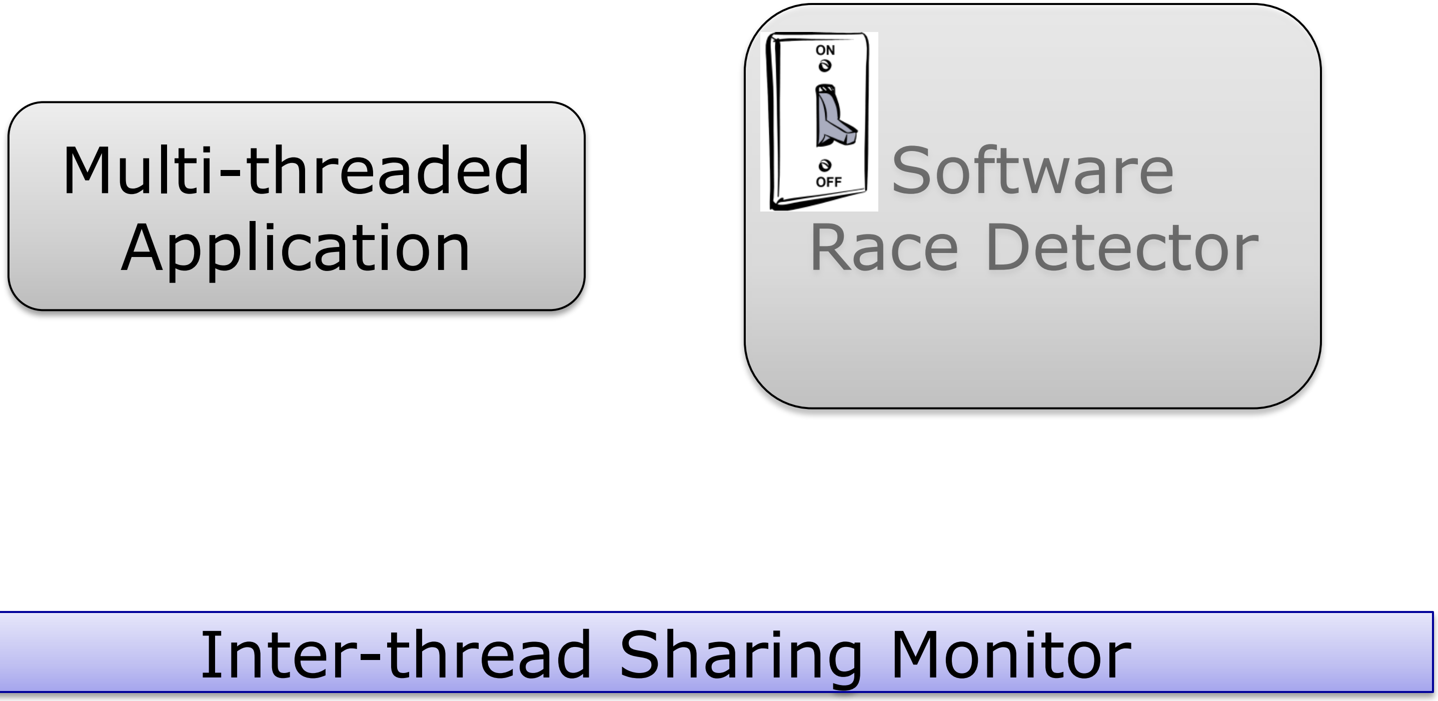
Little Sharing Means Wasted Work



M d

Software Race Detector

Use Demand-Driven Analysis!

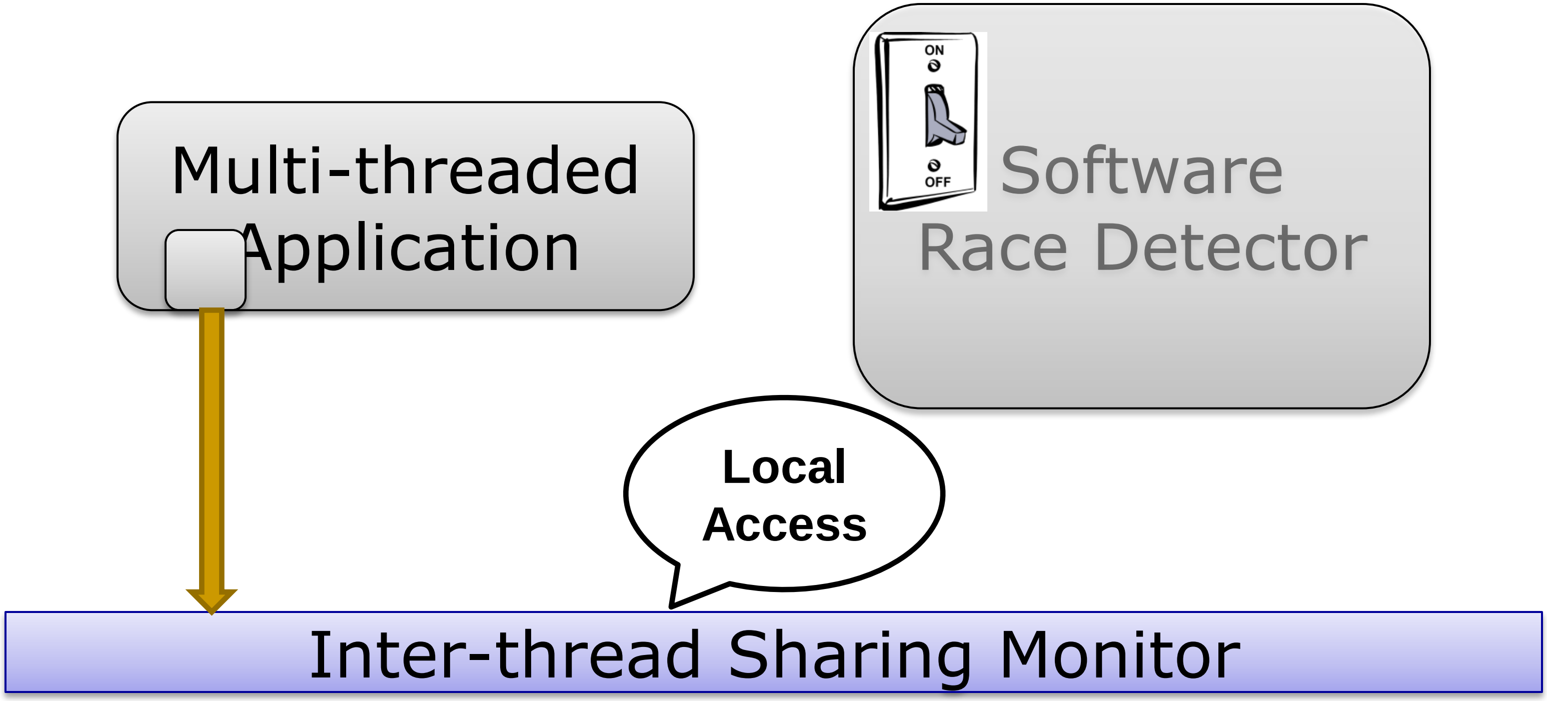


Multi-threaded Application

Software Race Detector

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!



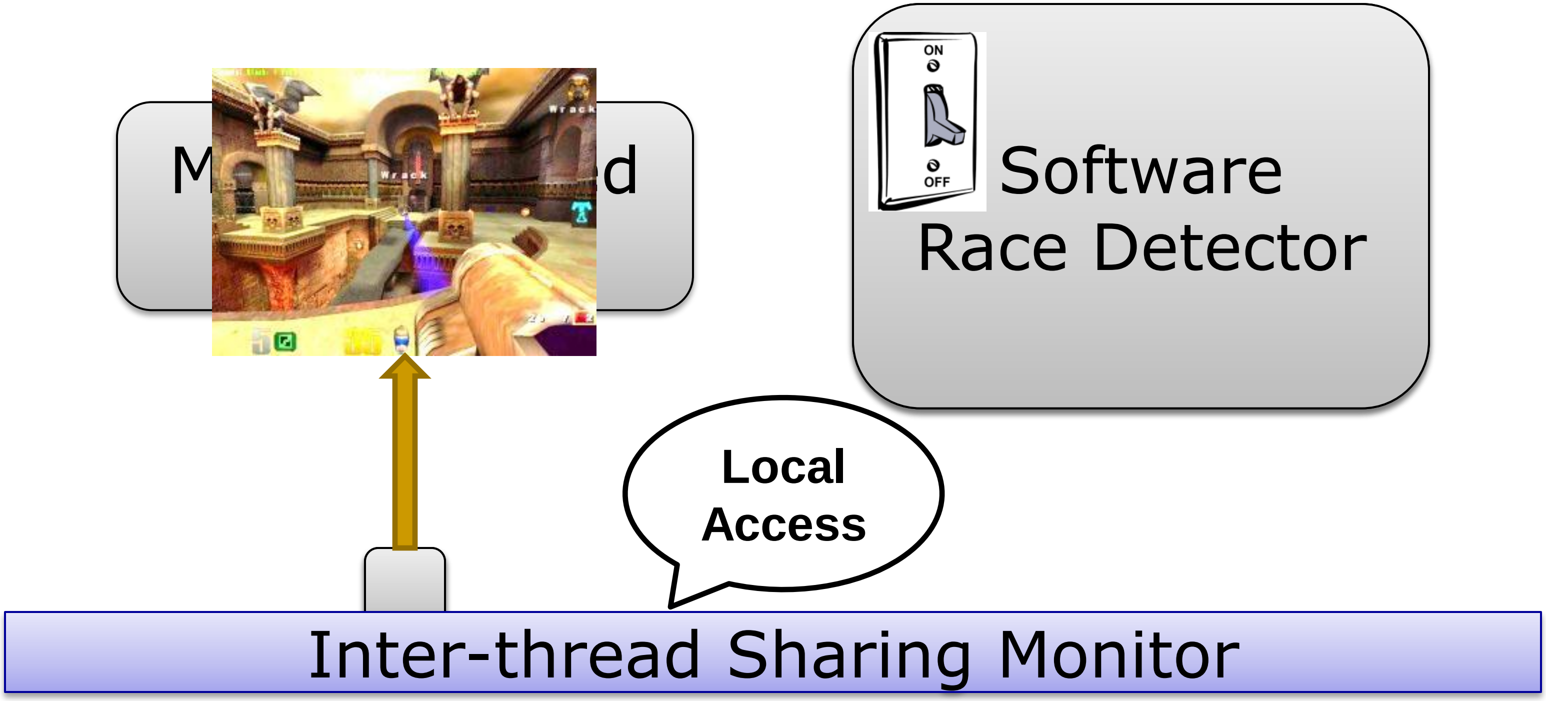
Multi-threaded Application

Software Race Detector

Local Access

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!

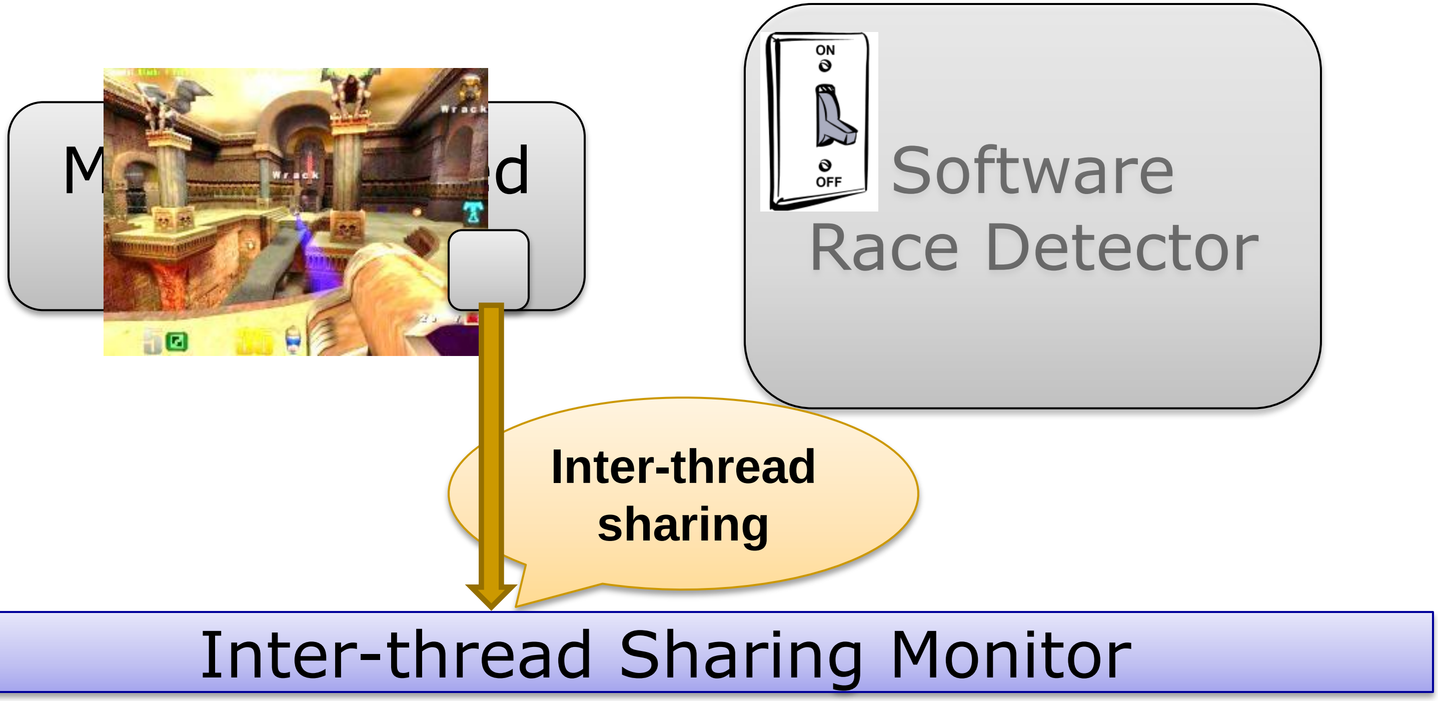


Software Race Detector

Local Access

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!

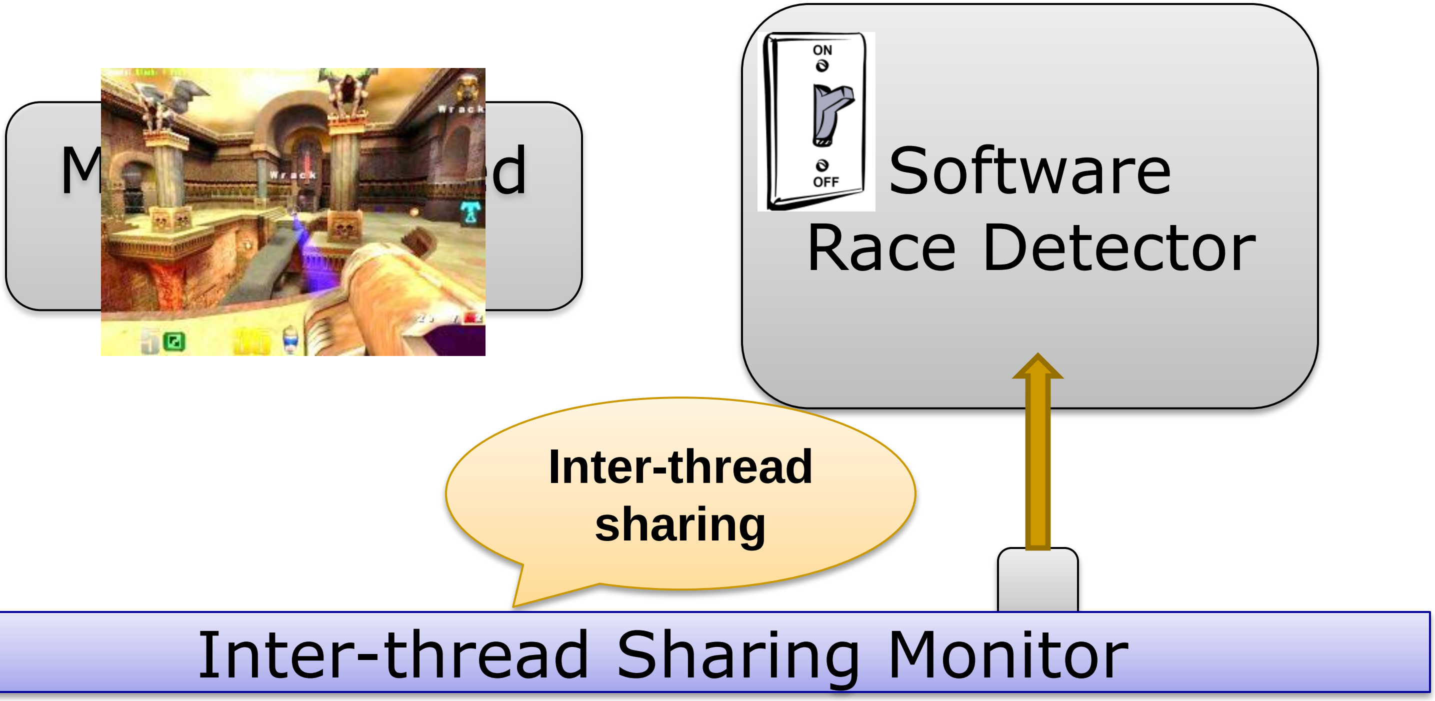


Software Race Detector

Inter-thread sharing

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!



Software Race Detector

Inter-thread sharing

Inter-thread Sharing Monitor

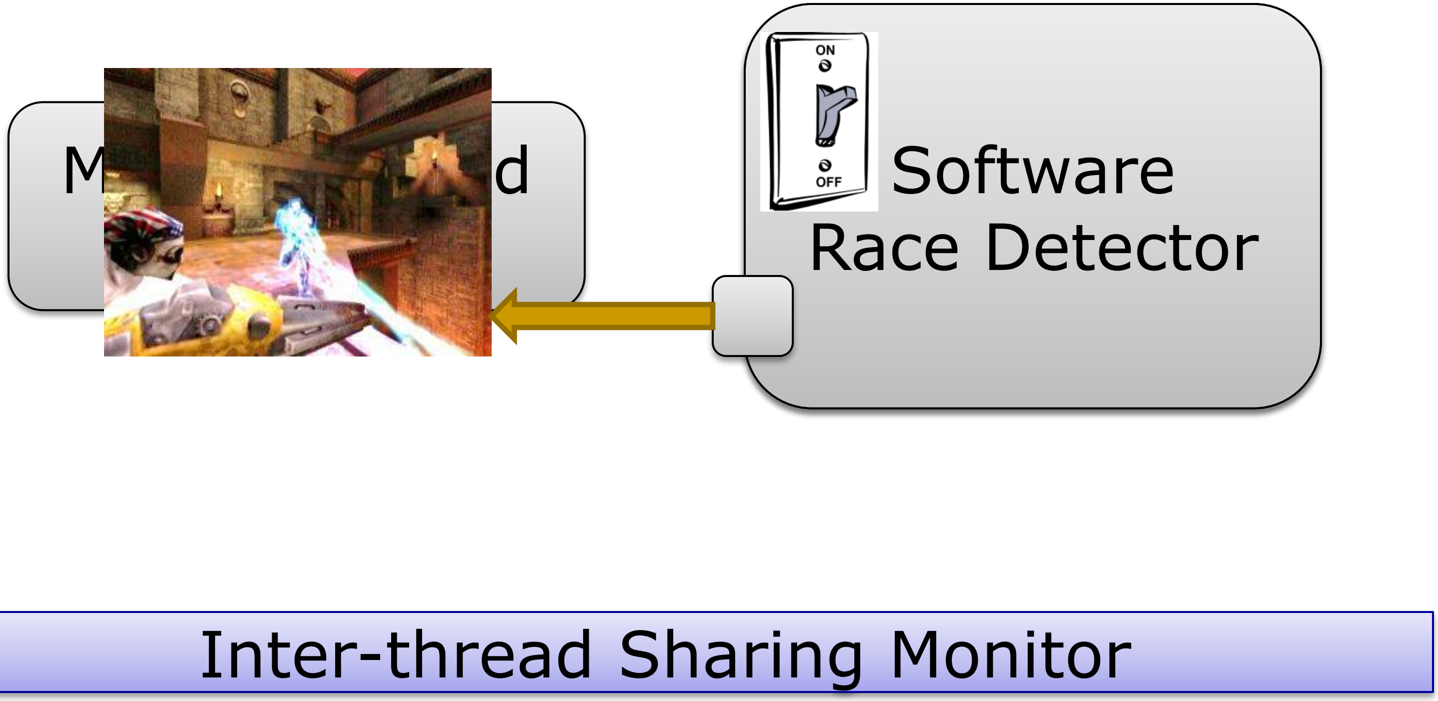
Use Demand-Driven Analysis!

M d

Software
Race ector

Inter-thread Sharing Monitor

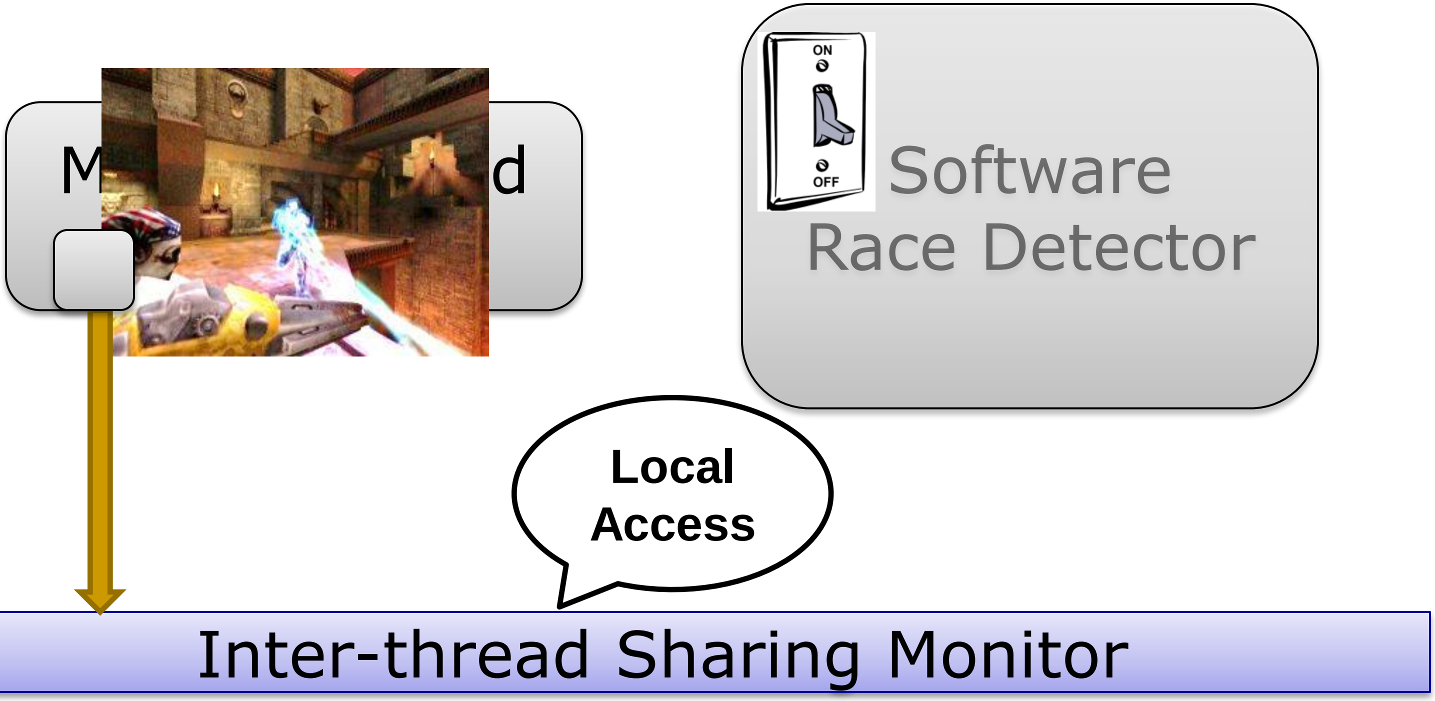
Use Demand-Driven Analysis!



Software Race Detector

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!



M

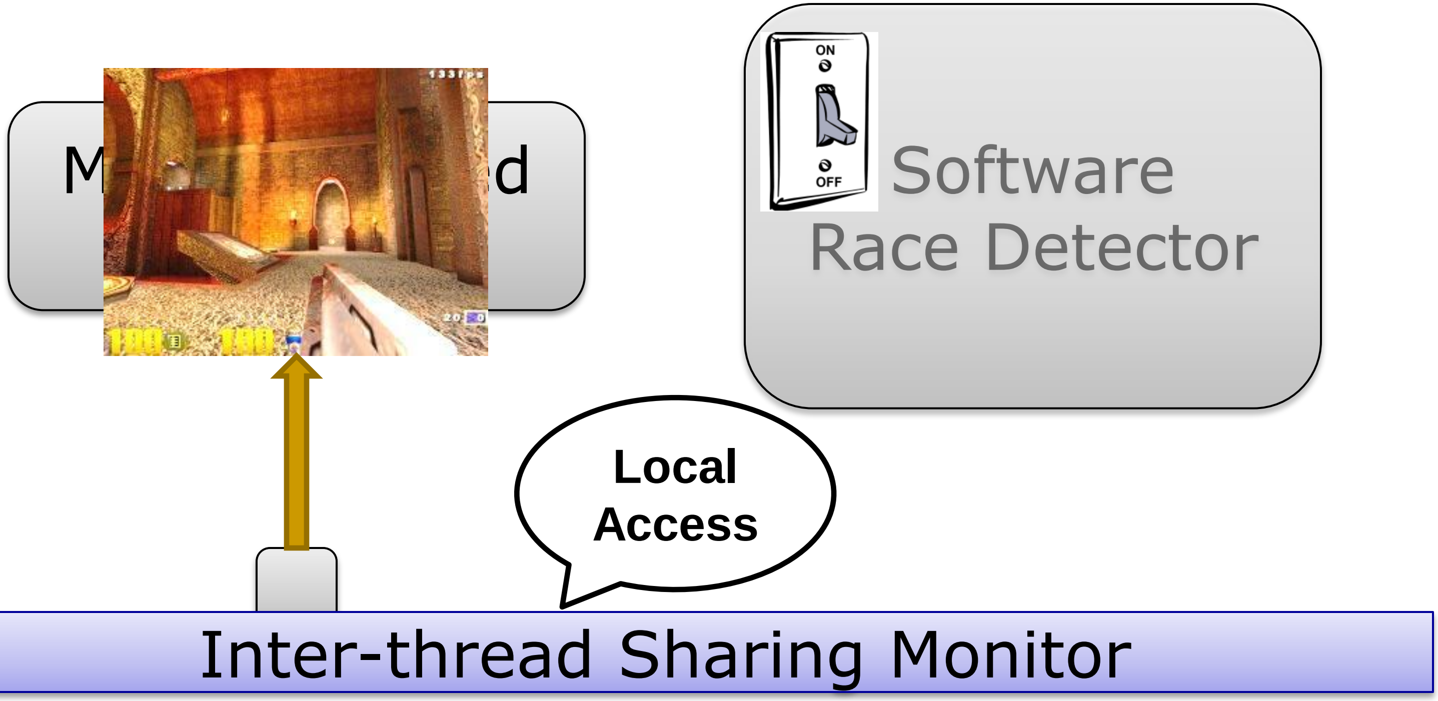
d

Software Race Detector

Local Access

Inter-thread Sharing Monitor

Use Demand-Driven Analysis!



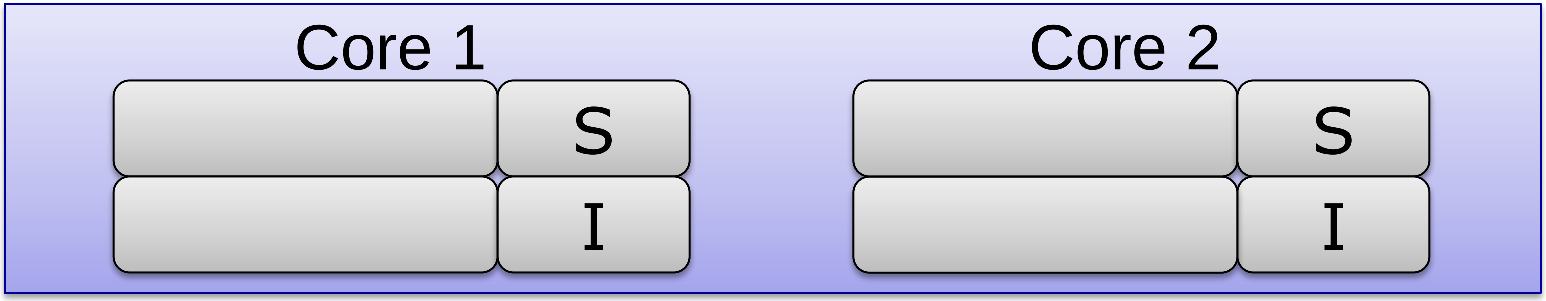
Software Race Detector

Local Access

Inter-thread Sharing Monitor

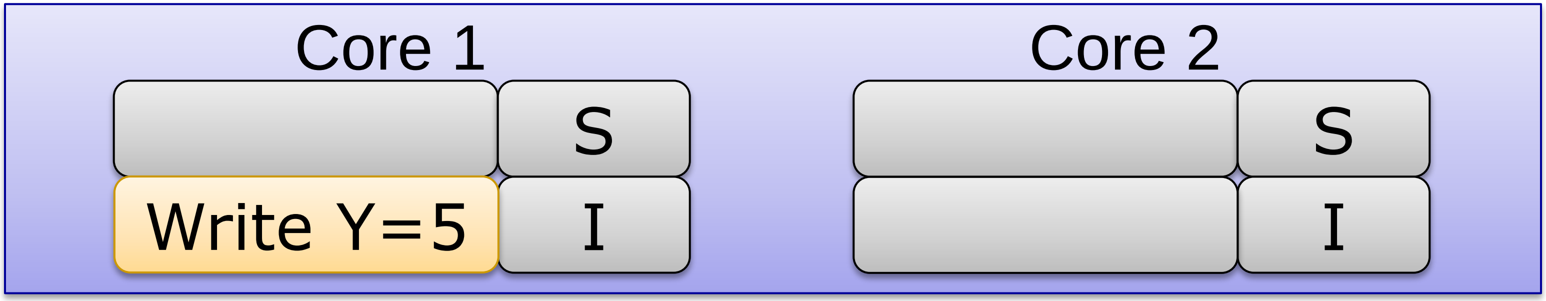
Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing



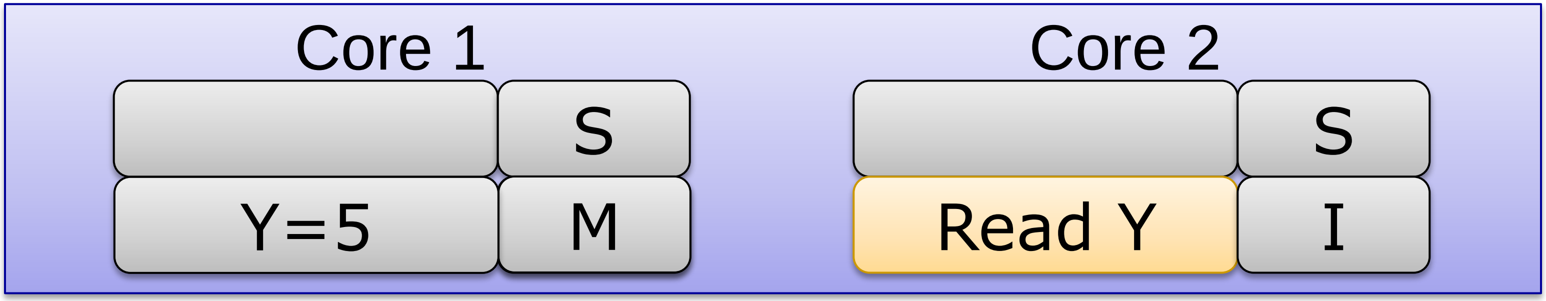
Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing



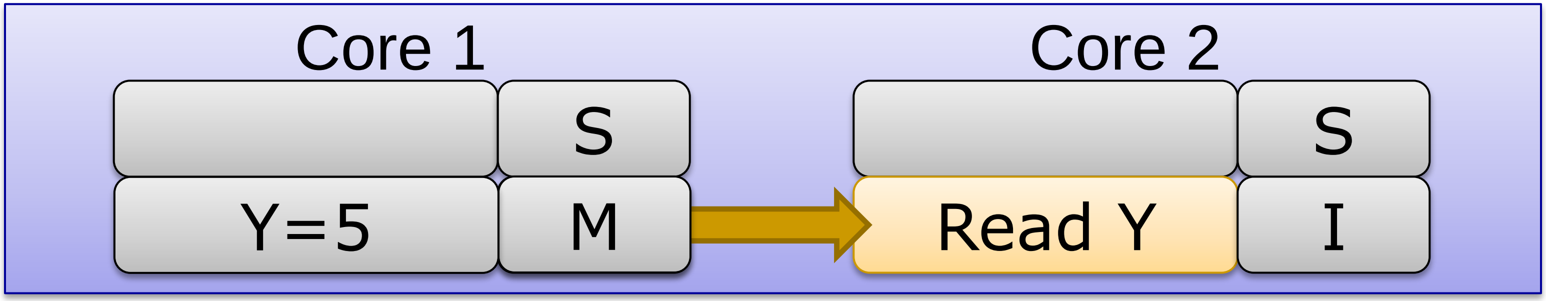
Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing



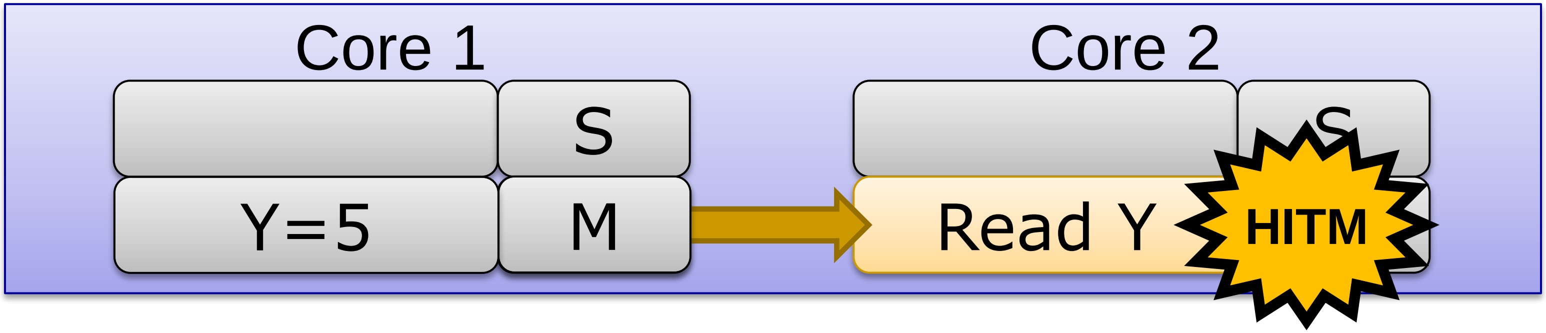
Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing



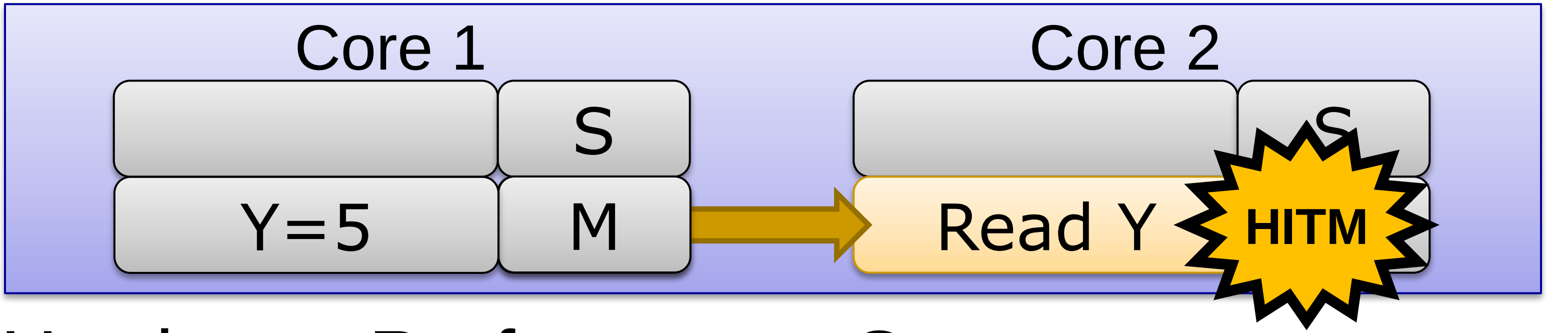
Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

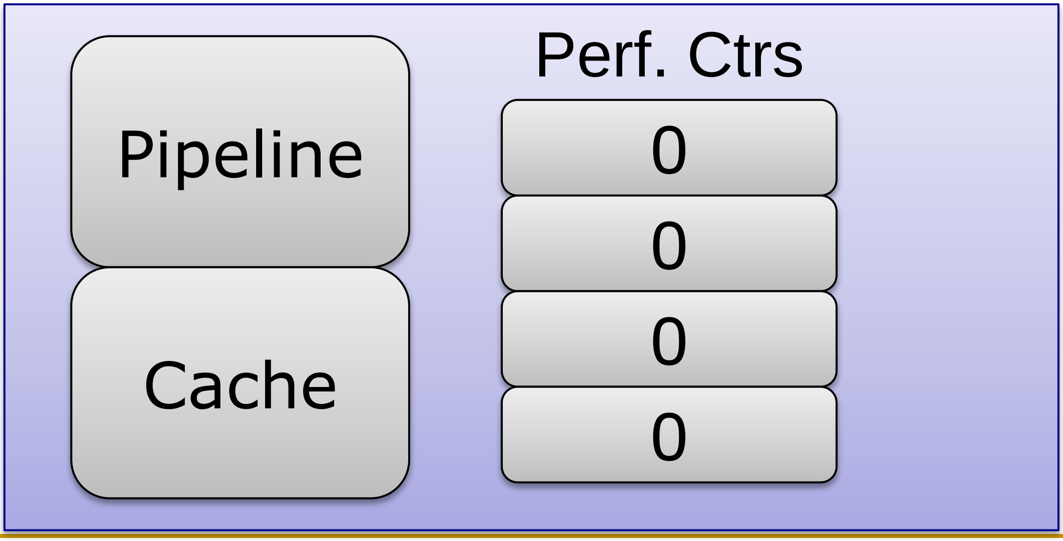


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

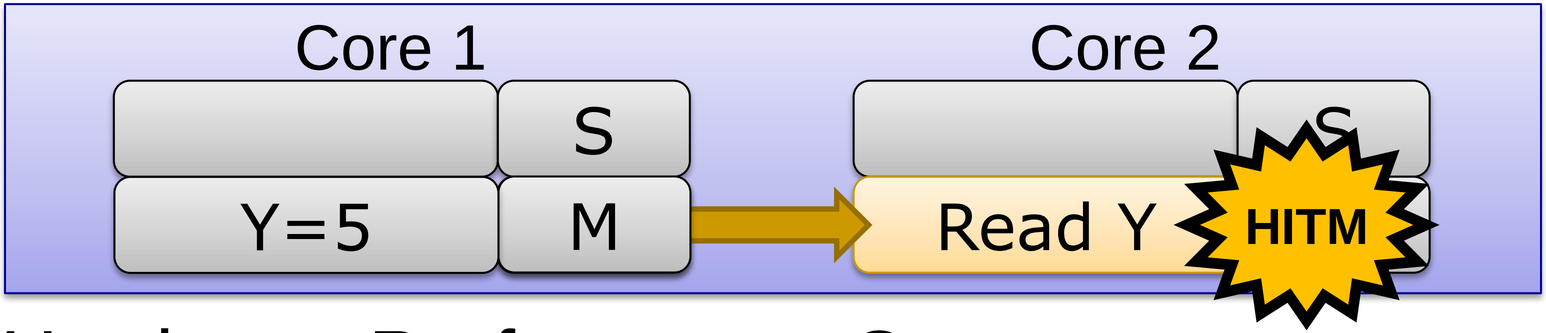


- Hardware Performance Counters

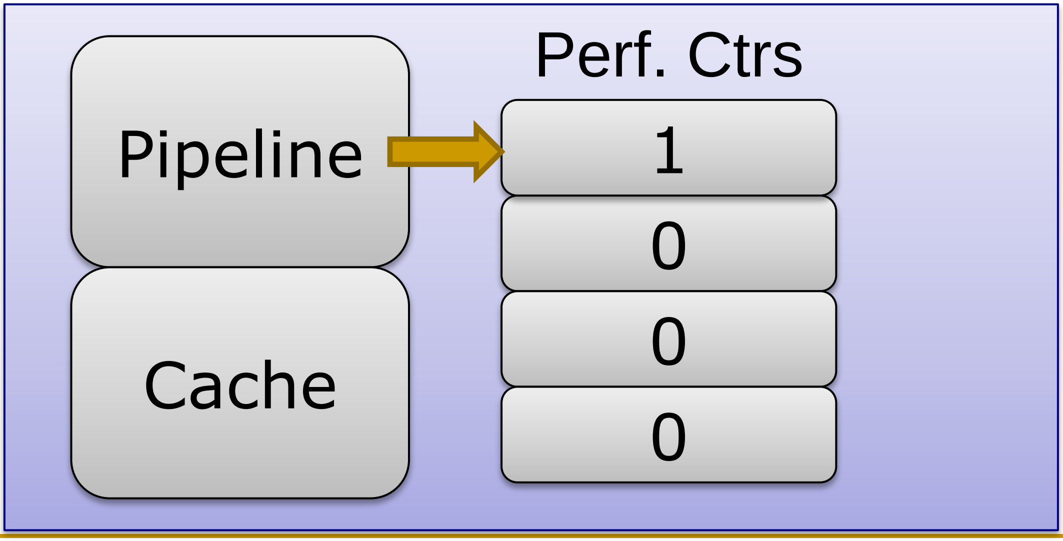


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

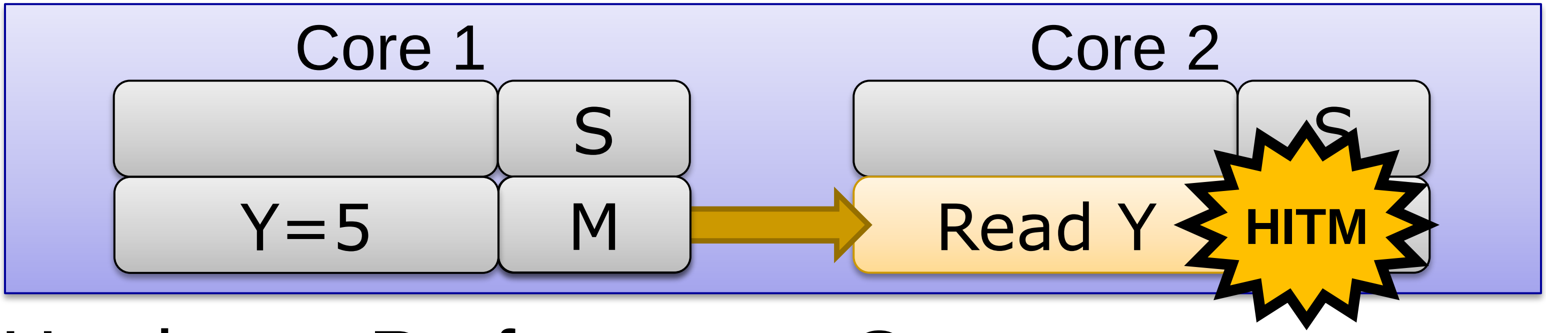


- Hardware Performance Counters

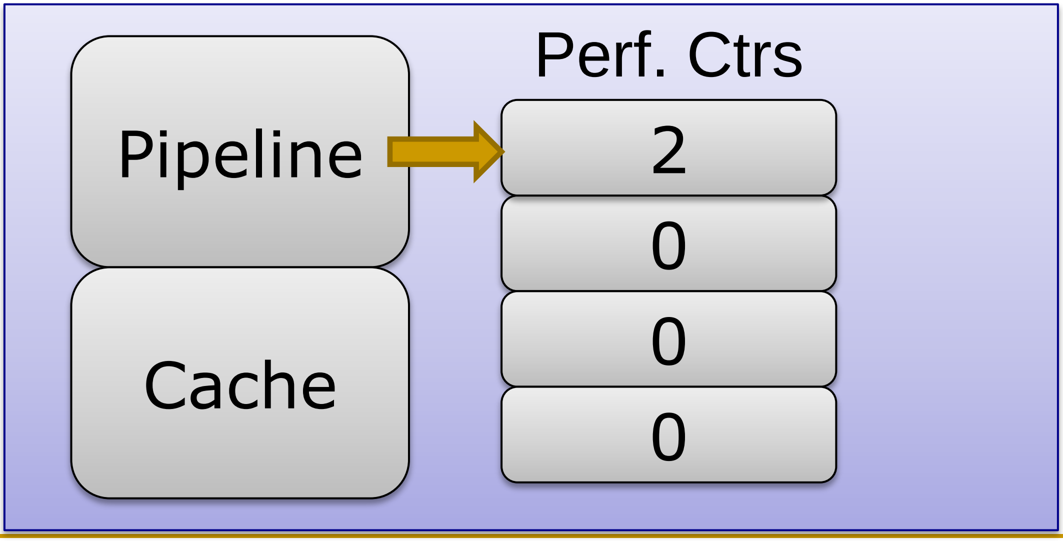


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

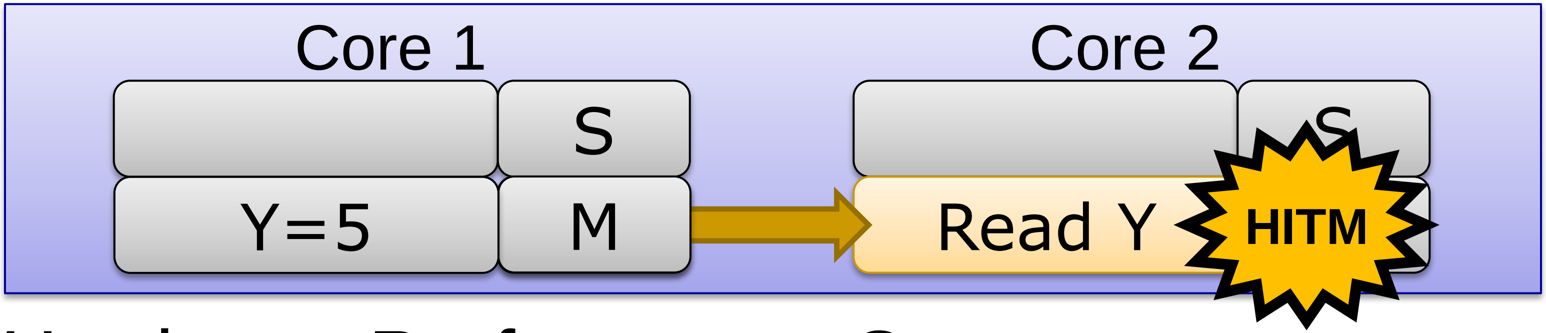


- Hardware Performance Counters

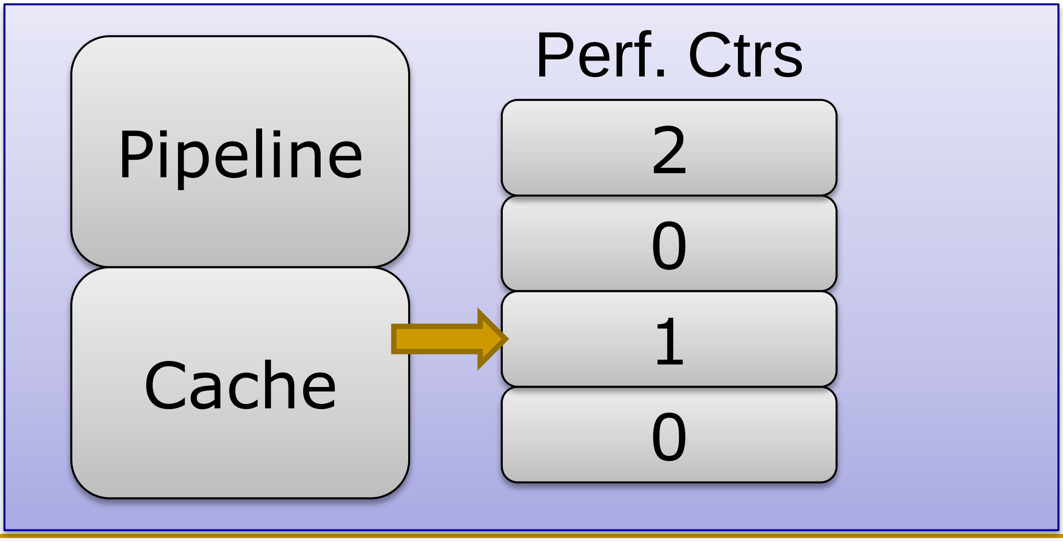


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

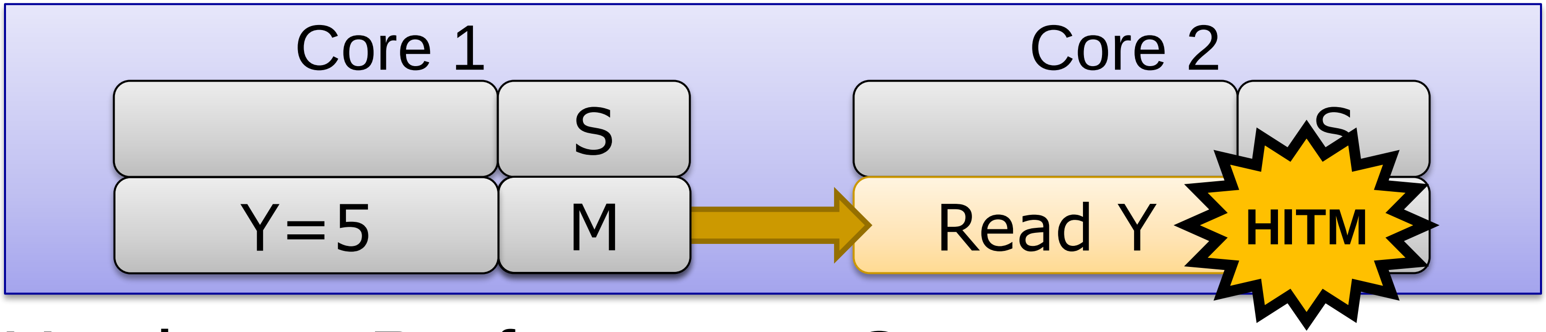


- Hardware Performance Counters

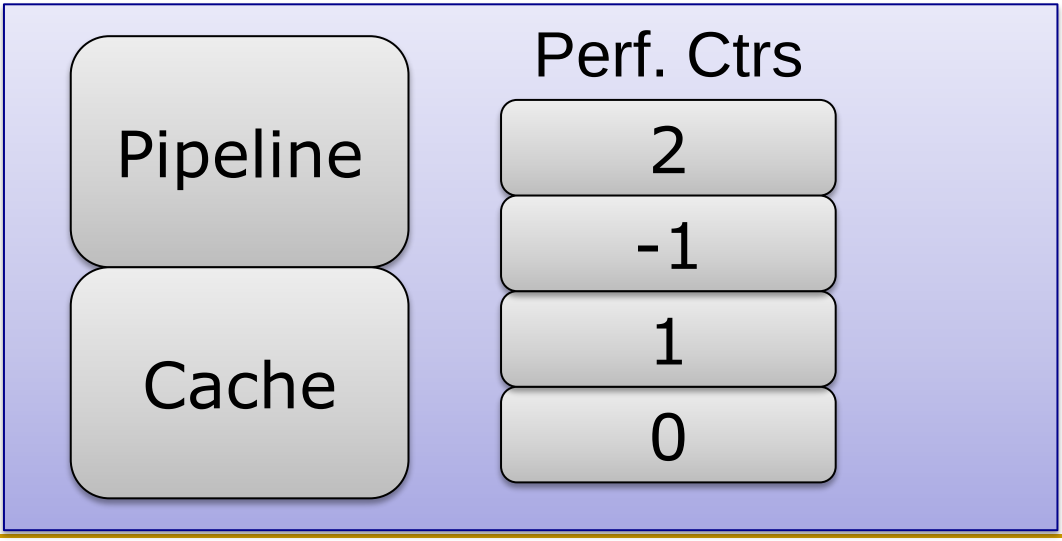


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing

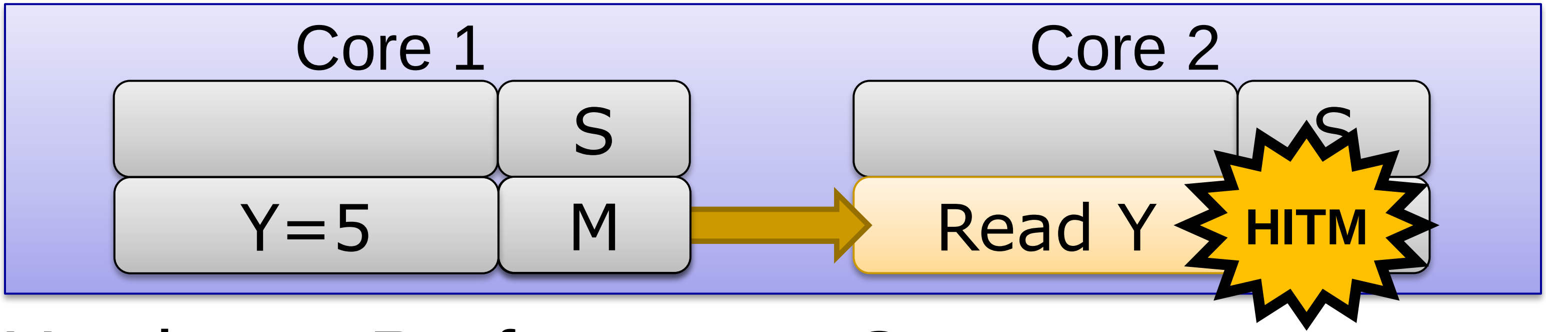


- Hardware Performance Counters

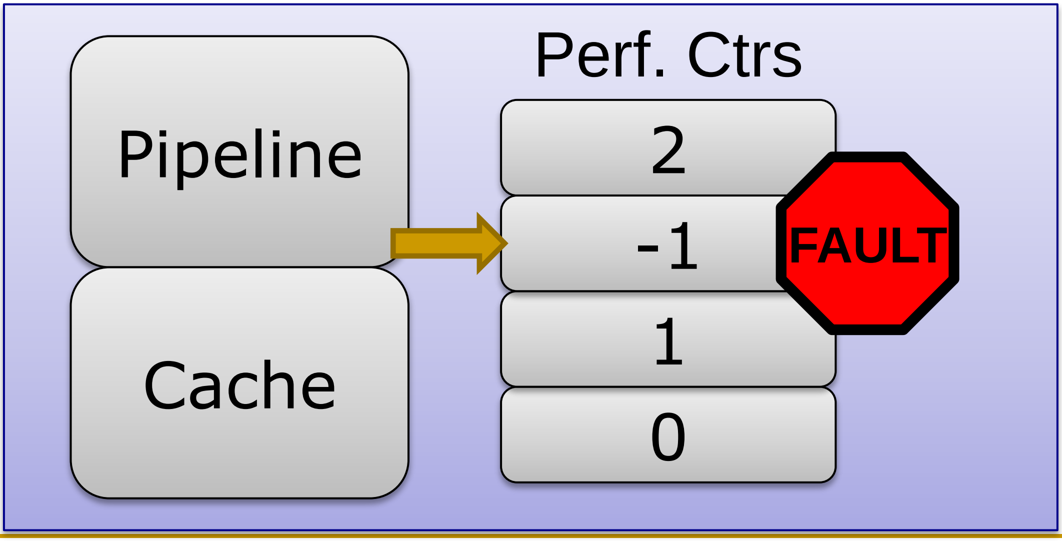


Hardware Sharing Detector

- HITM in Cache Memory: W→R Data Sharing



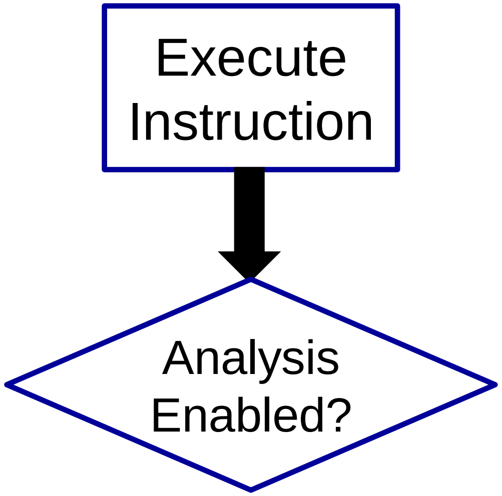
- Hardware Performance Counters



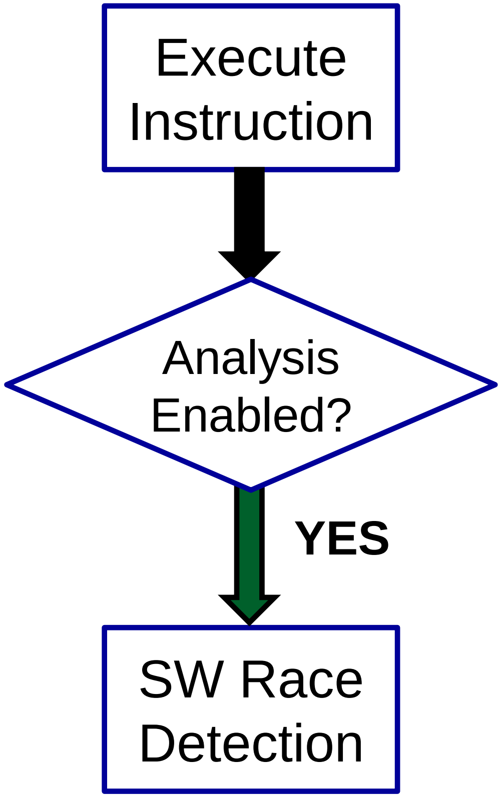
On-Demand Analysis on Real HW

Execute Instruction

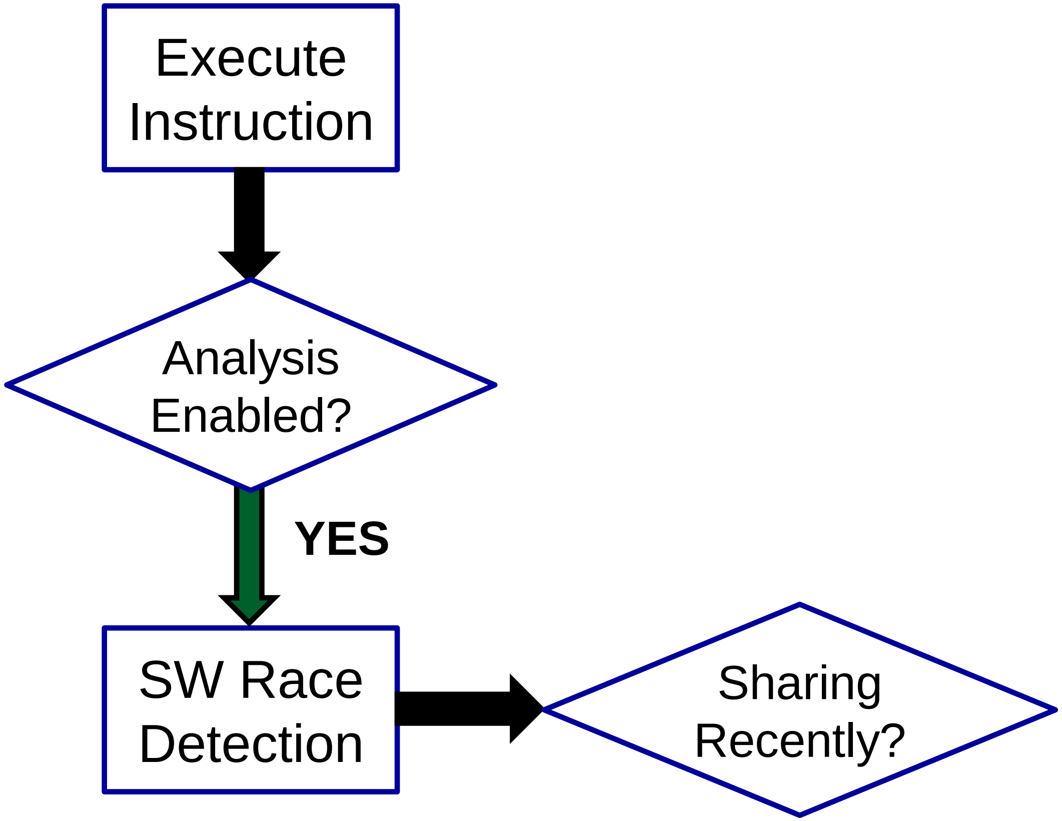
On-Demand Analysis on Real HW



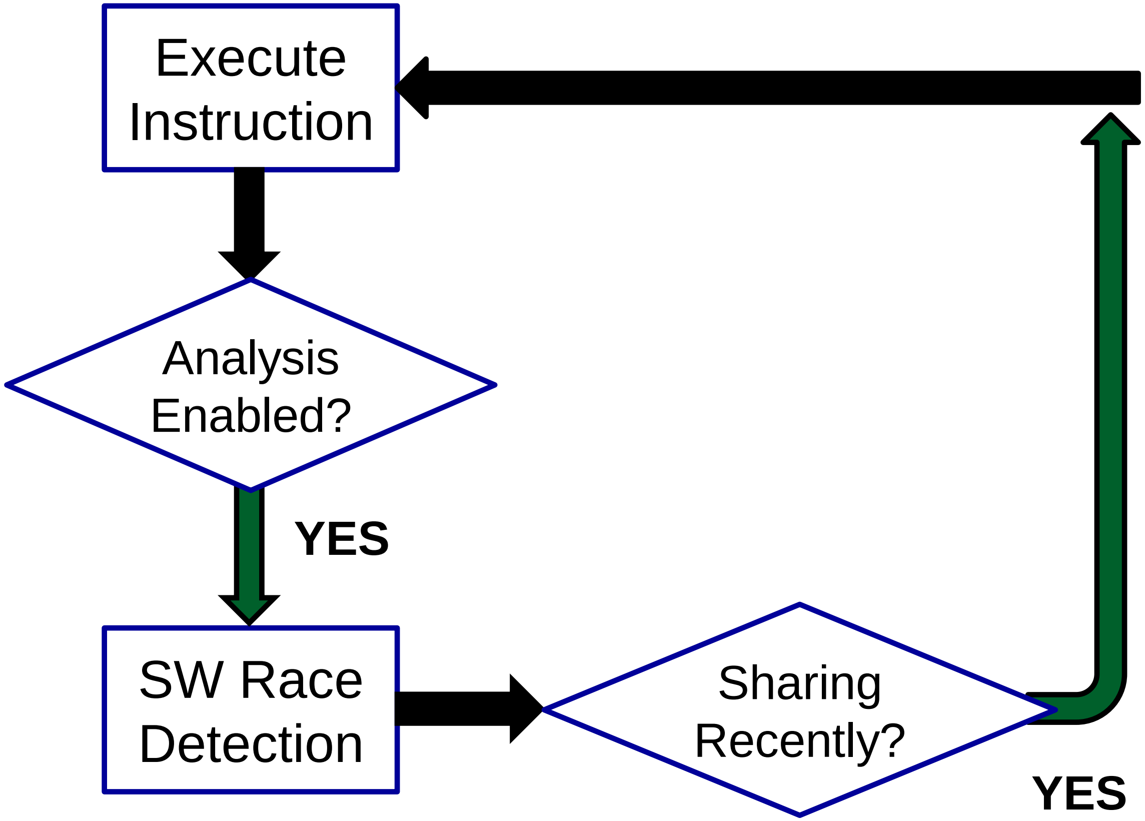
On-Demand Analysis on Real HW



On-Demand Analysis on Real HW



On-Demand Analysis on Real HW



Execute Instruction

Analysis Enabled?

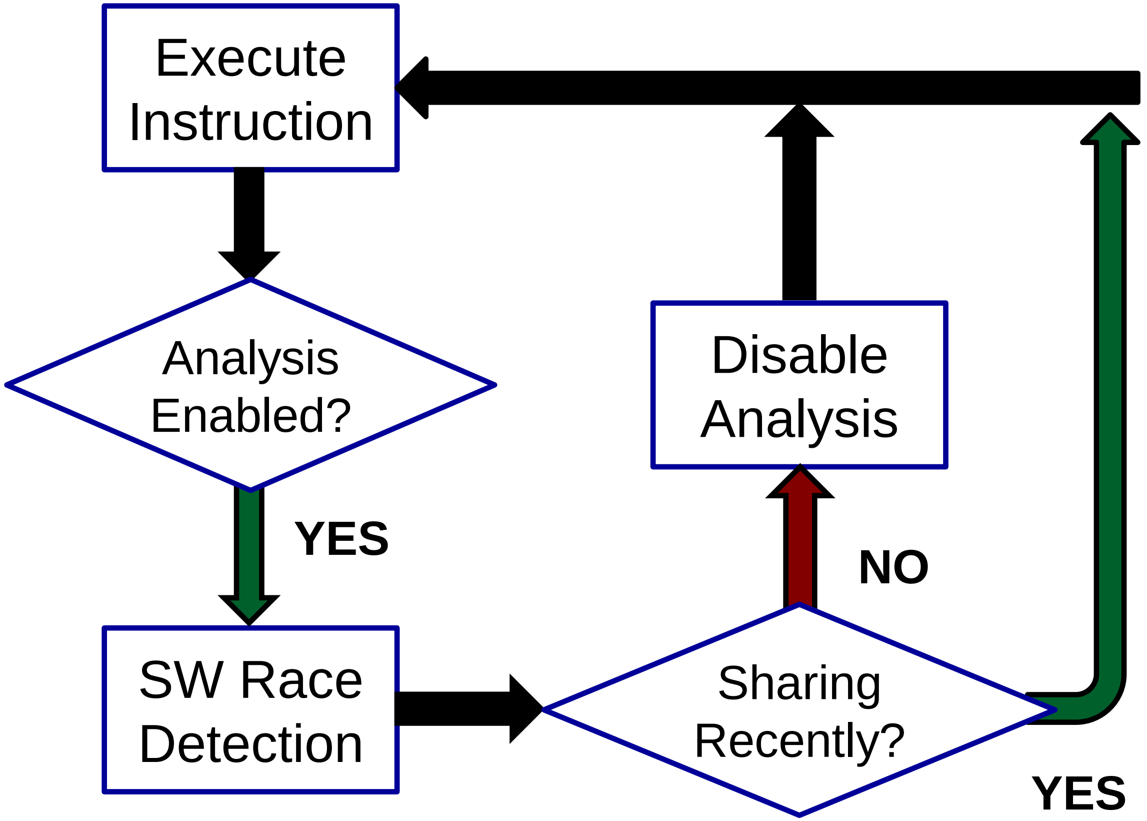
YES

SW Race Detection

Sharing Recently?

YES

On-Demand Analysis on Real HW



Execute Instruction

Analysis Enabled?

YES

SW Race Detection

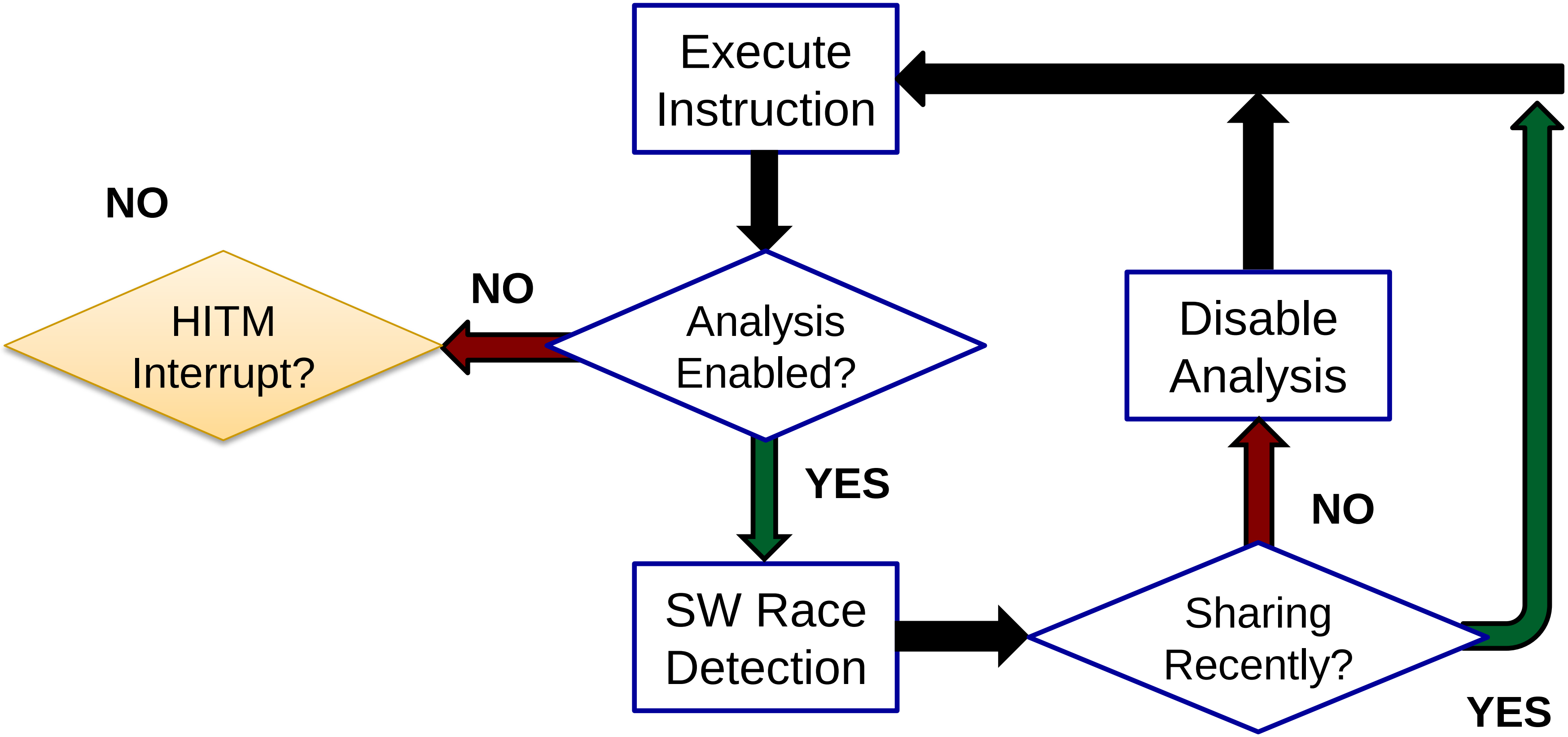
Sharing Recently?

NO

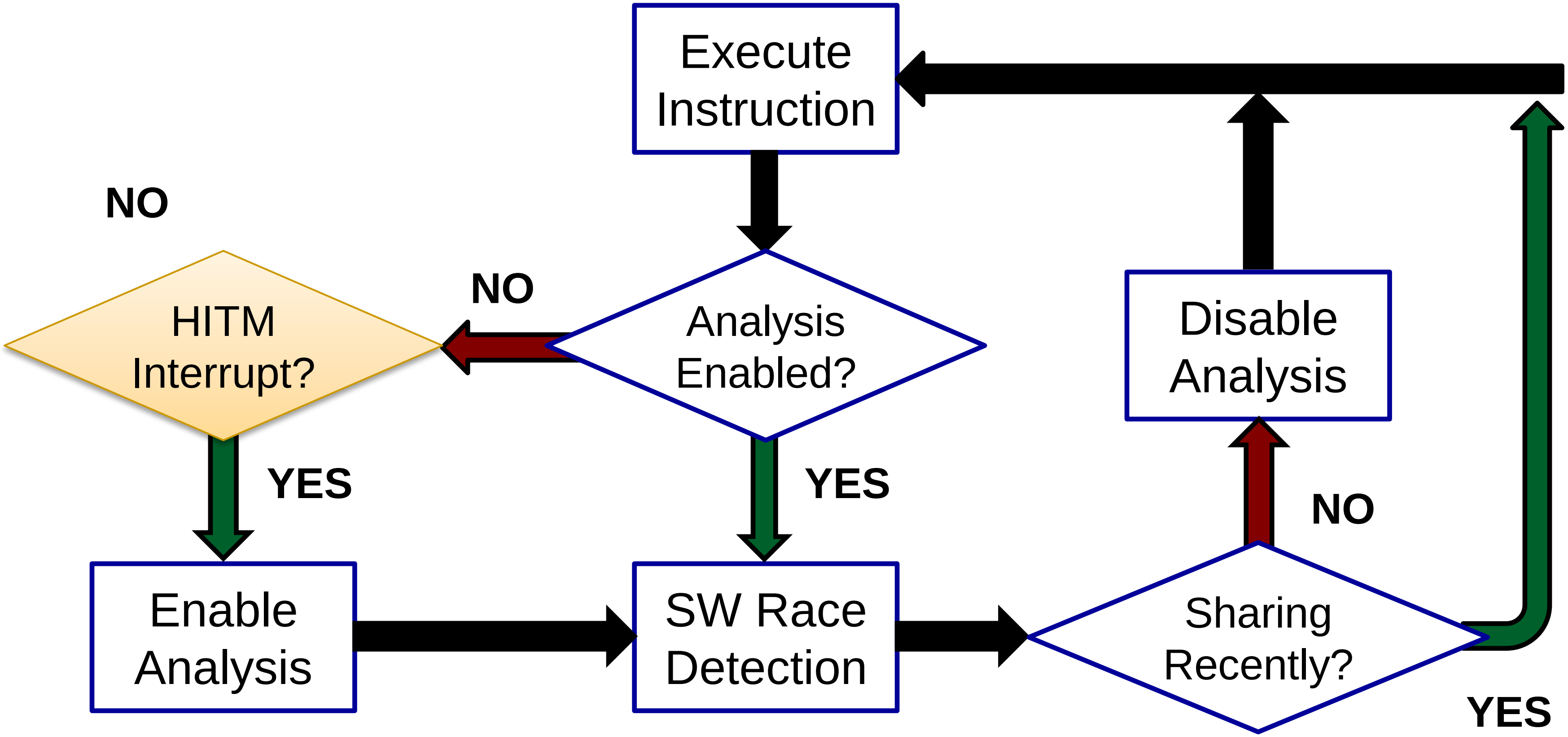
Disable Analysis

YES

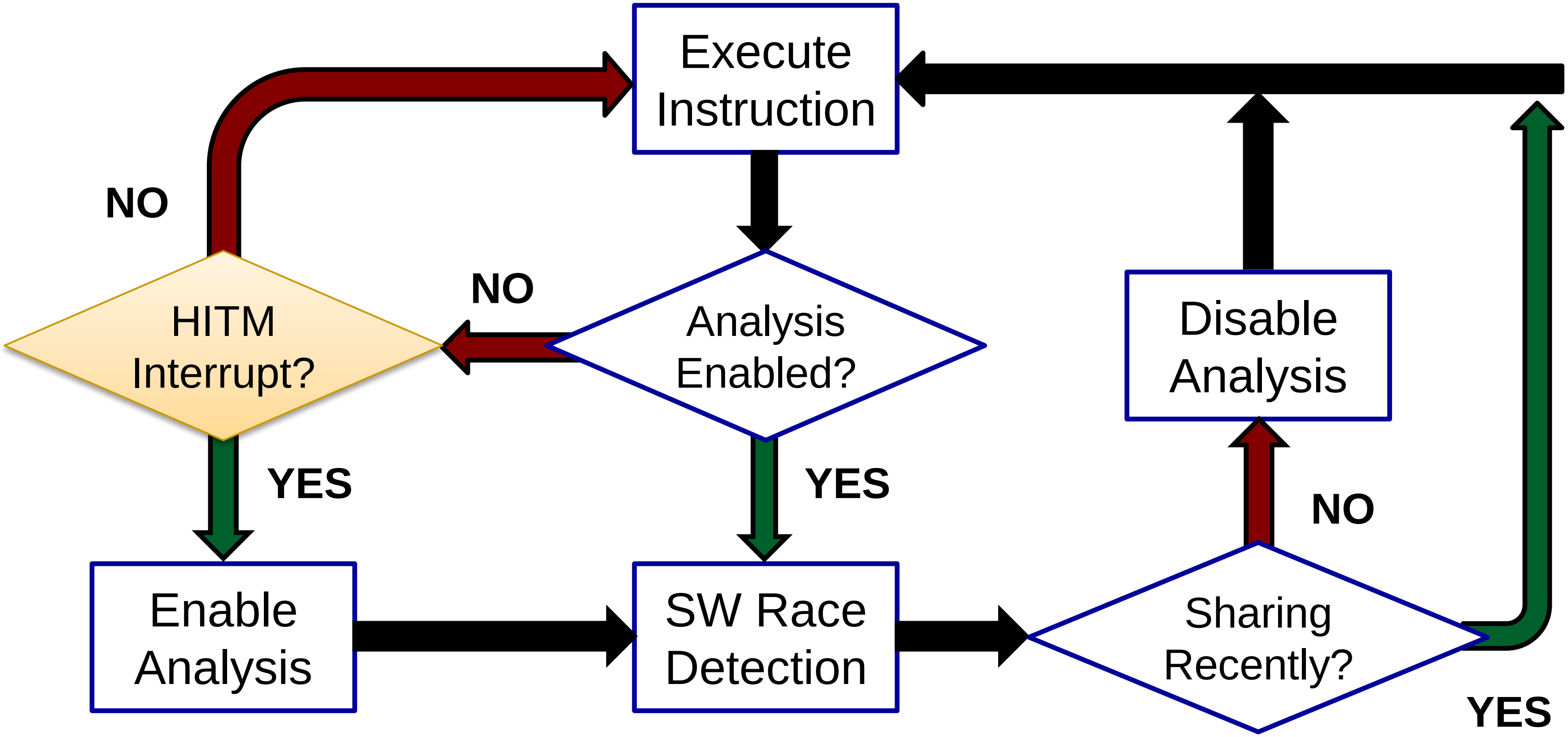
On-Demand Analysis on Real HW



On-Demand Analysis on Real HW



On-Demand Analysis on Real HW



Execute Instruction

Analysis Enabled?

HITM Interrupt?

Enable Analysis

SW Race Detection

Sharing Recently?

Disable Analysis

NO

NO

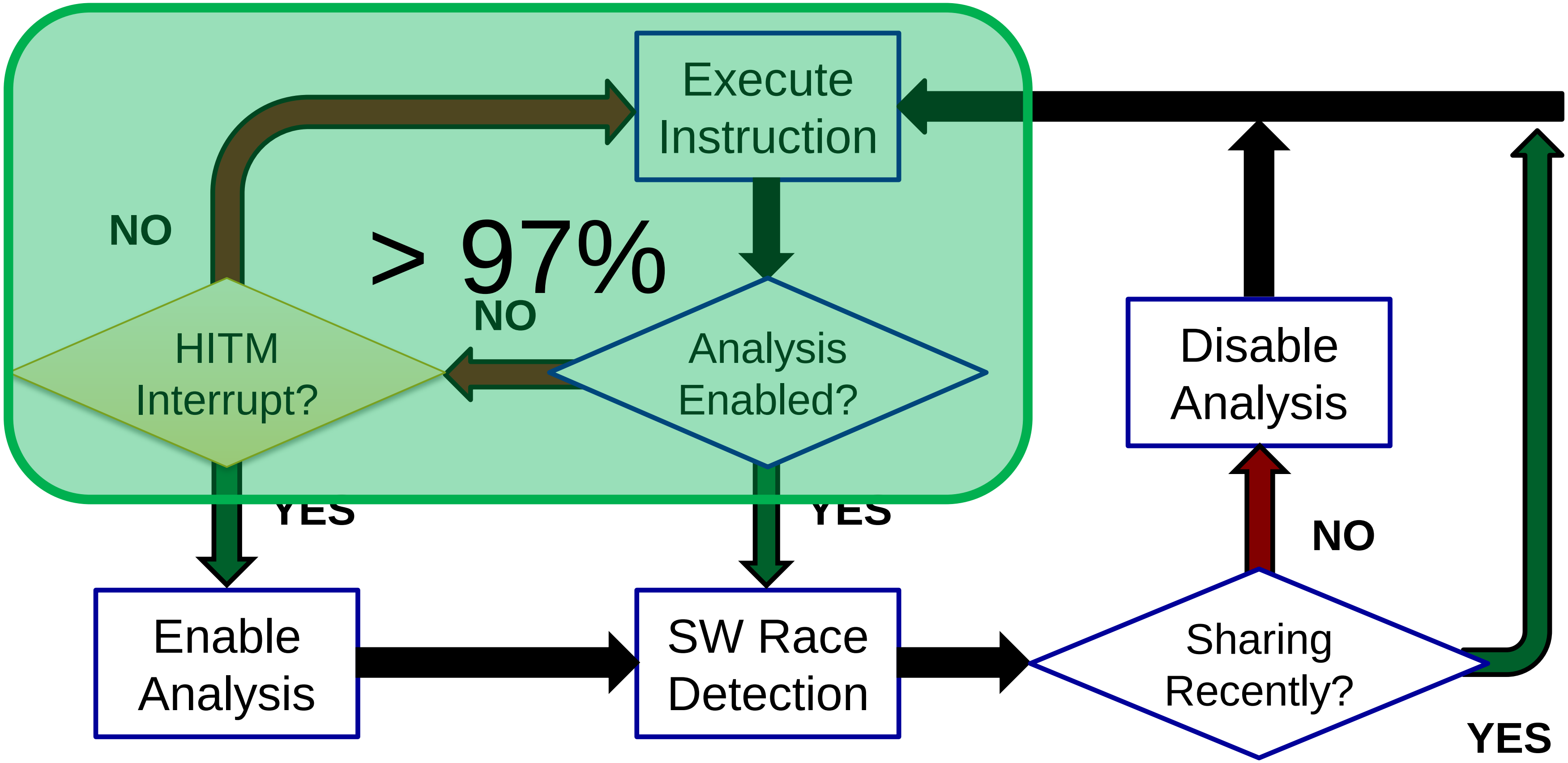
YES

YES

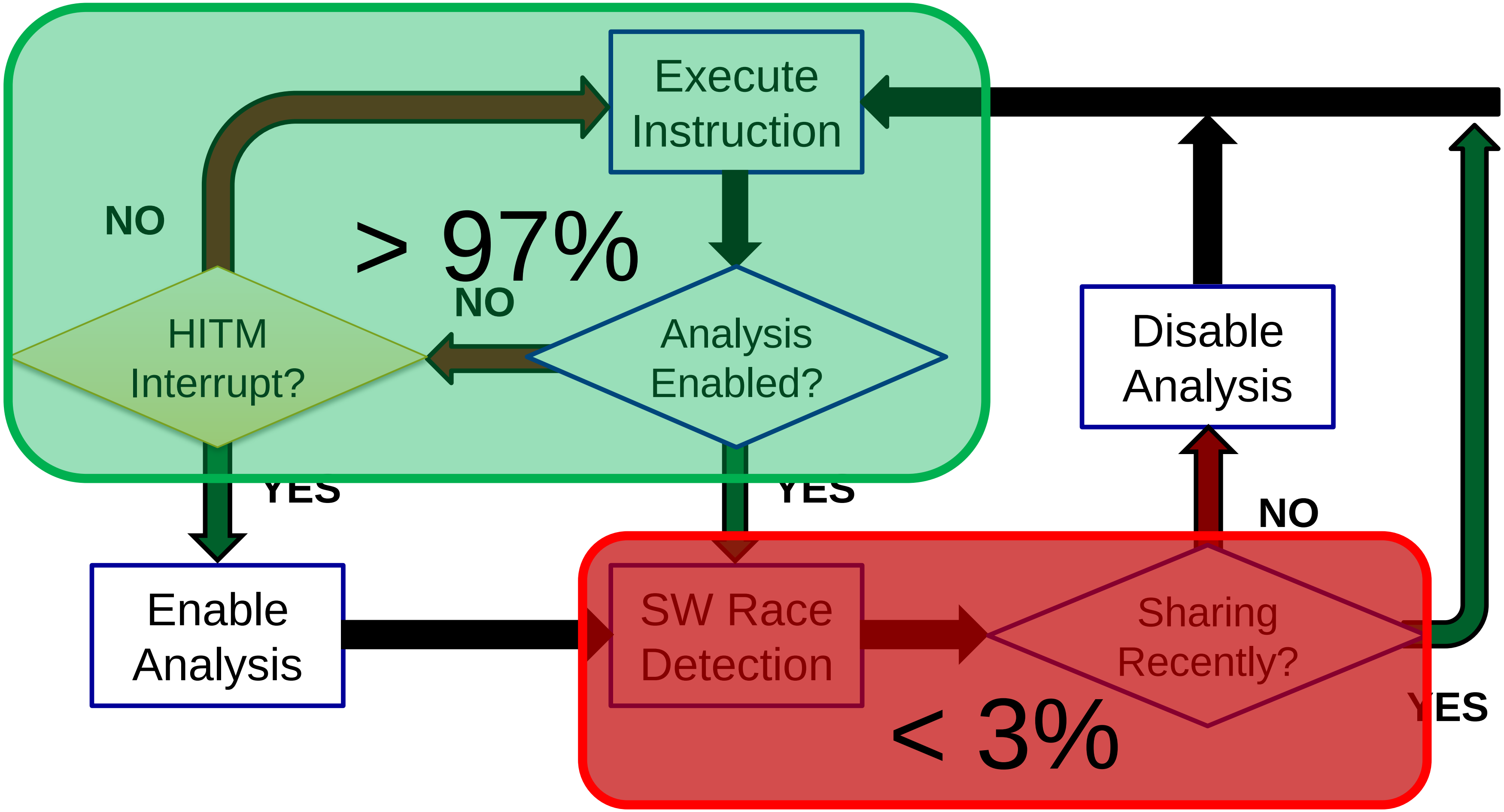
NO

YES

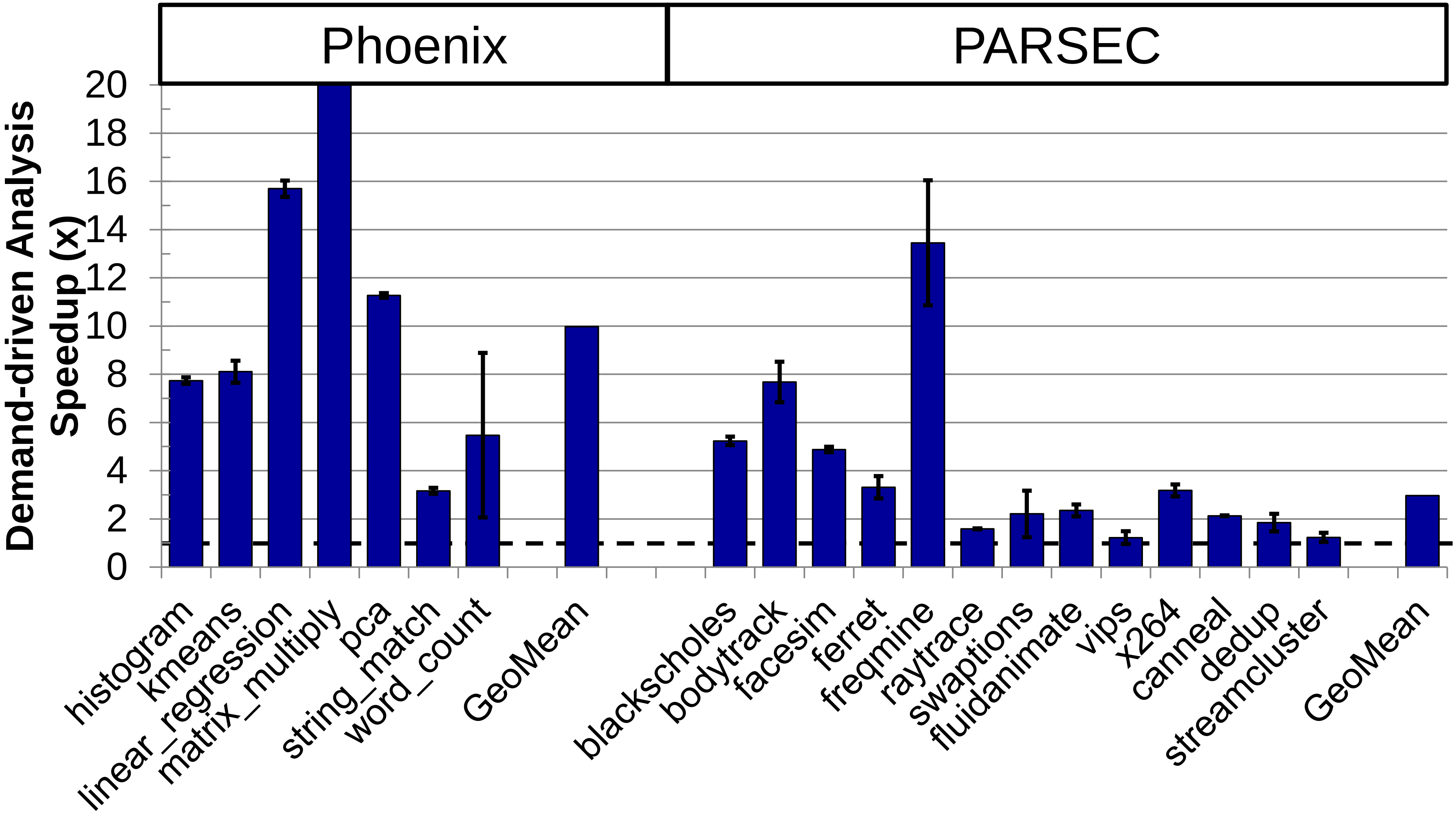
On-Demand Analysis on Real HW



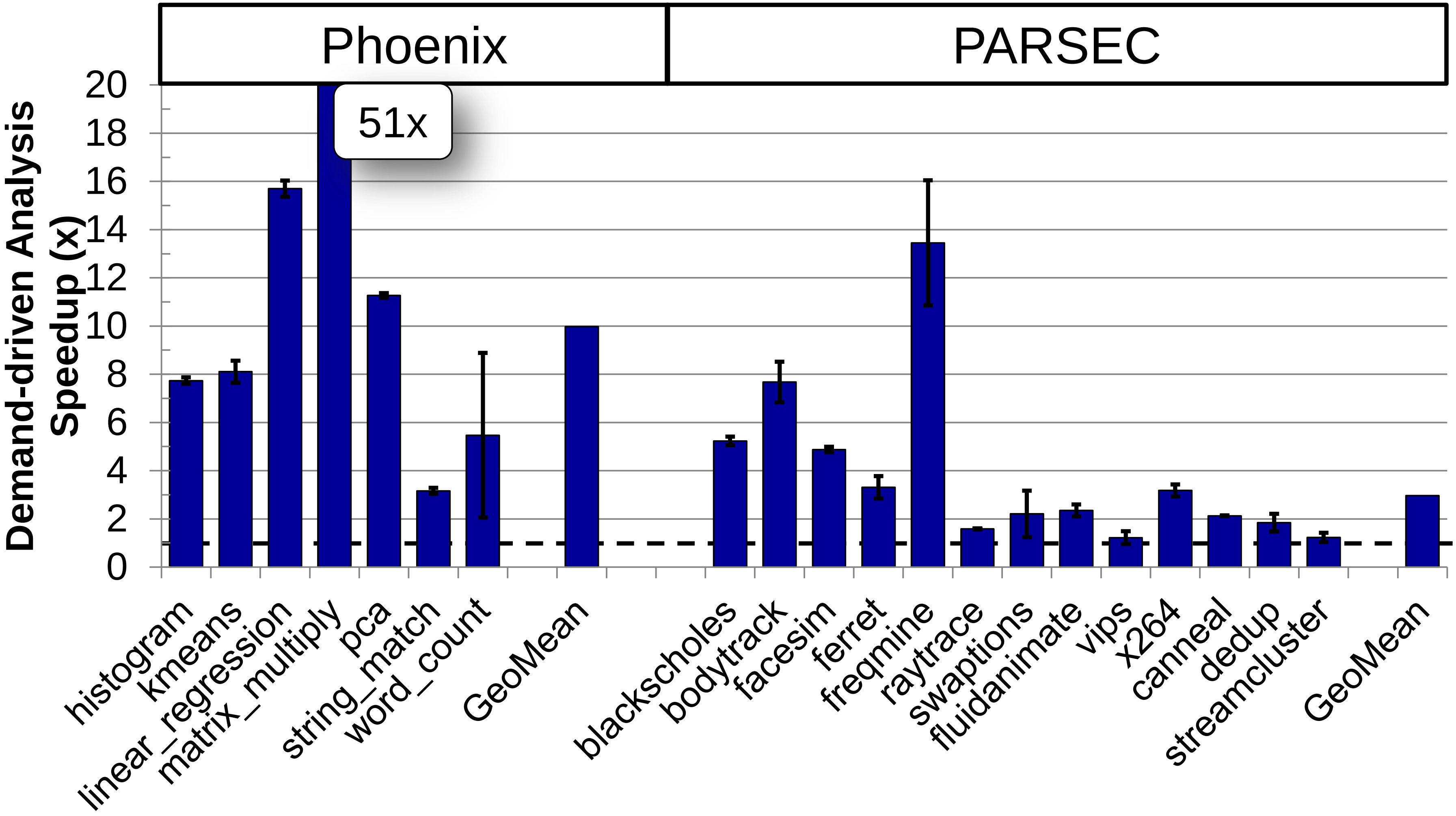
On-Demand Analysis on Real HW



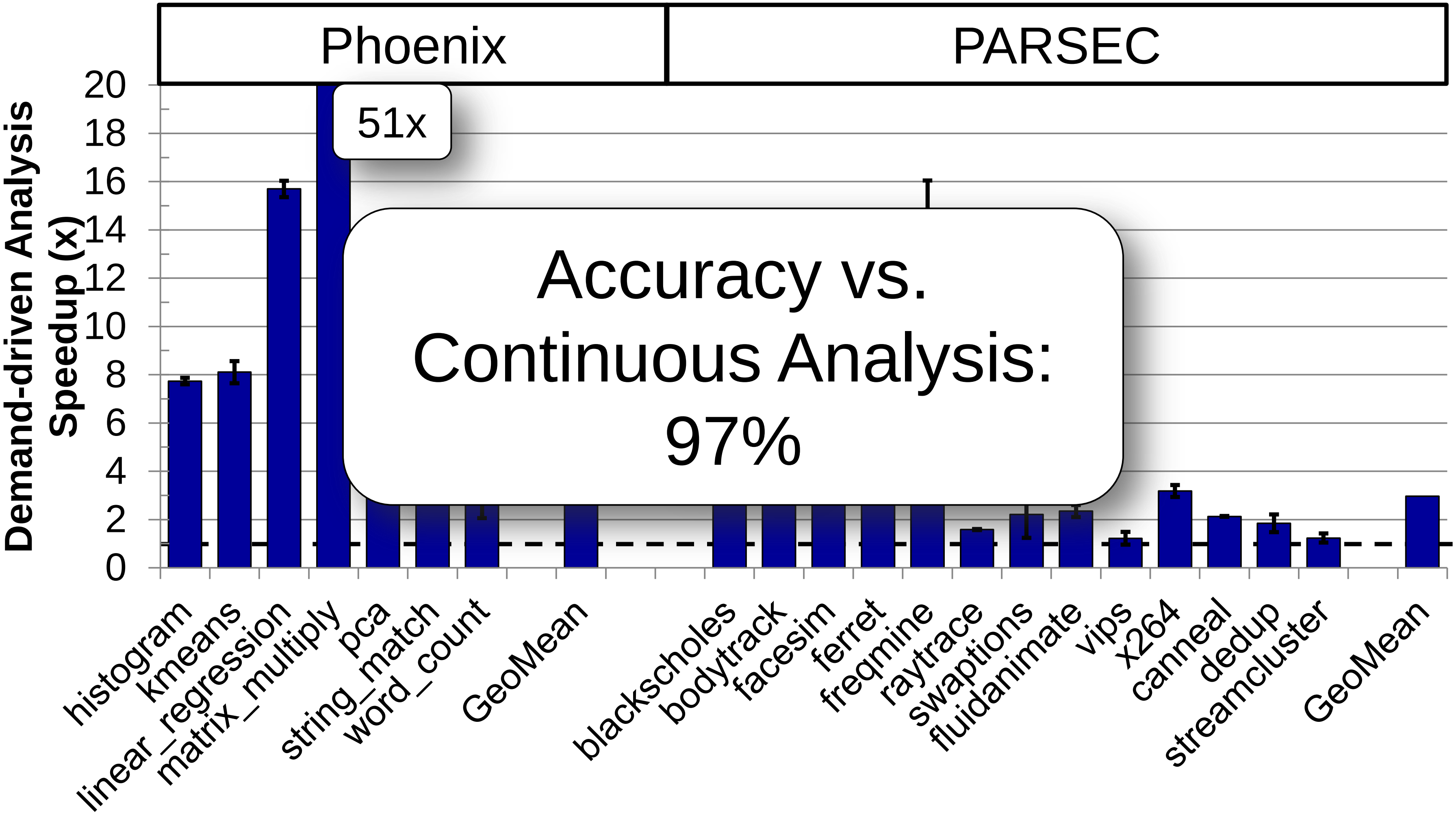
Performance Increases



Performance Increases



Performance Increases



Accuracy vs. Continuous Analysis: 97%

In Summary

Hardware makes constructing software difficult.

Tools make software better.

Hardware can (and should!) help these tools.

BACKUP SLIDES

Width Test