

facebook

HHVM on AArch64

Max Wang

Software Engineer



HHVM

Agenda

Four questions:

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

Agenda

Four questions:

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

Agenda

Four questions:

1

What is HHVM?

2

How did we get running on AArch64?

3

Will the demo work?

4

Where do we go from here?

Agenda

Four questions:

1

What is HHVM?

2

How did we get running on AArch64?

3

Will the demo work?

4

Where do we go from here?

Agenda

Four questions:

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

Agenda

i.e., *Why should you care?*

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

What is HHVM?

In a nutshell

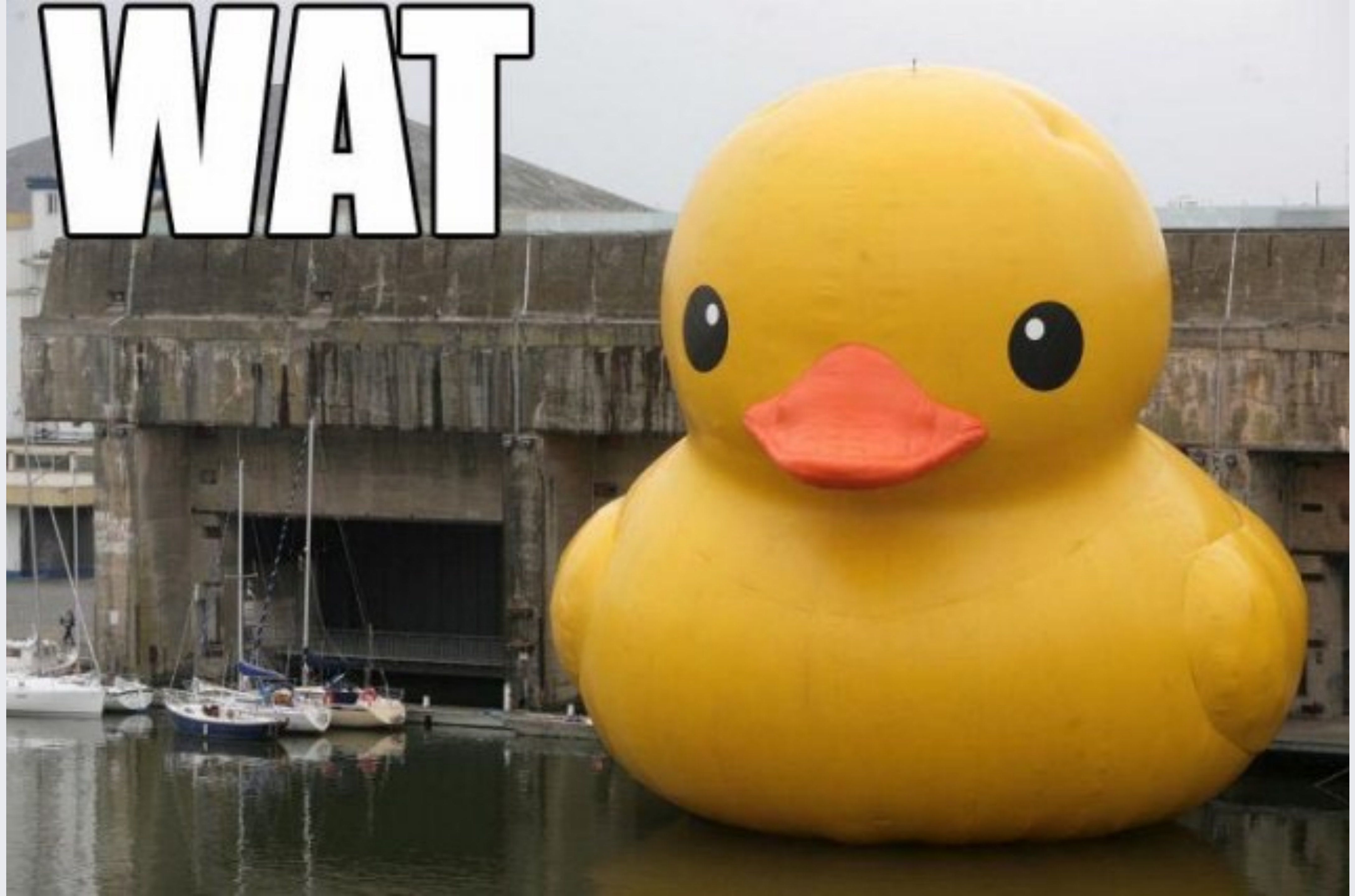
- Just-in-time compiler for PHP and Hack

```
"1" + "2" == 3
"1" + "2wo" == 3
"15" + "0xf" == 15
("15" == "0xf") == true

$str = "foo";
list($a, $b) = $str;
var_dump($a); // "f"

list($a, $b) = "foo";
var_dump($a); // NULL
```

WAT



What is HHVM?

In a nutshell

- Just-in-time compiler for PHP and Hack

```
<?hh

class Foo<T> {
    public async function getBar(
        dict<string,T> $ts
    ): Awaitable<Bar> {
        return await fetch_bar($this->priv, $ts);
    }
}
```

What is HHVM?

In a nutshell

- Serves production web traffic for Facebook
- HHVM is fast!
 - Orders of magnitude improvement
- Not just for FB!

What is HHVM?

In a nutshell

- Open source (<https://github.com/facebook/hhvm>)
- Used by 3 of the Alexa Top 5
 - Facebook, Baidu, Wikipedia
- Also: Box, Slack, Etsy, Wordpress, ...



- 1 [Google.com](#)
Enables users to search the world's info

- 2 [Youtube.com](#)
User-submitted videos with rating, com

- 3 [Facebook.com](#)
A social utility that connects people, to k

- 4 [Baidu.com](#)
The leading Chinese language search er

- 5 [Wikipedia.org](#)
A free encyclopedia built collaboratively

- 6 [Yahoo.com](#)
A major internet portal and service prov

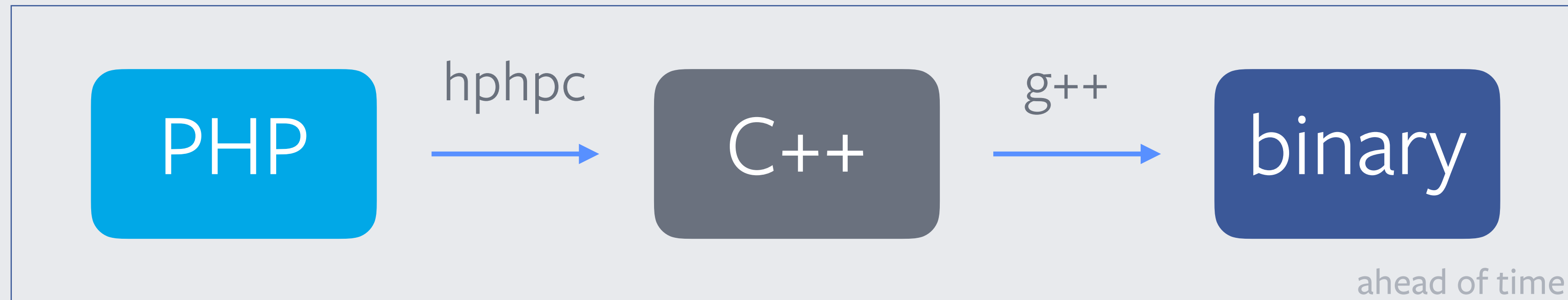
What is HHVM?

In a nutshell

- Fast!
- Open source!
- Just-in-time compiler for PHP and Hack

Compilation pipeline

HipHop for PHP (HPHPc)



- Major performance improvement over PHP5
- But:
 - slow ahead-of-time compilation
 - massive binary size
 - static type inference on a dynamic language

Compilation pipeline

High-level PHP bytecode



Compilation pipeline

High-level PHP bytecode

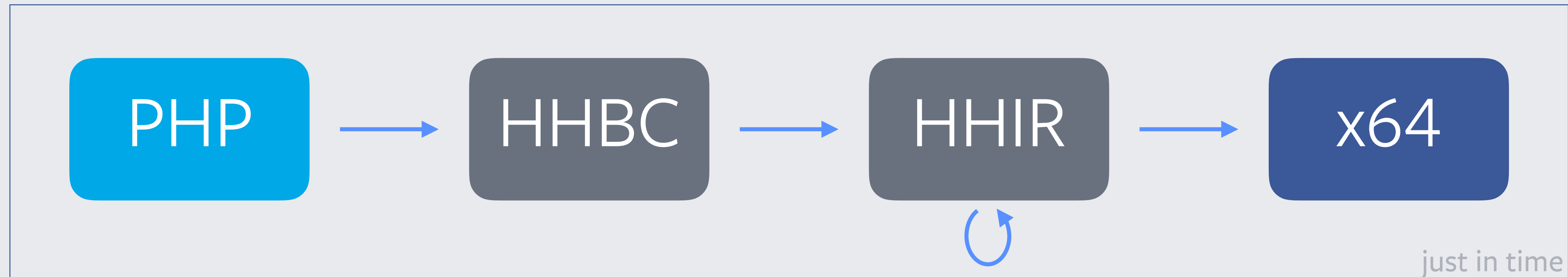


```
$elem = ...;  
if ($elem > 0) {  
    ...  
}
```

```
63  SetL L:4  
65  PopC  
66  Int 0  
75  CGetL2 L:4  
77  Gt  
78  JmpZ 13 (91)
```

Compilation pipeline

HHVM intermediate representation

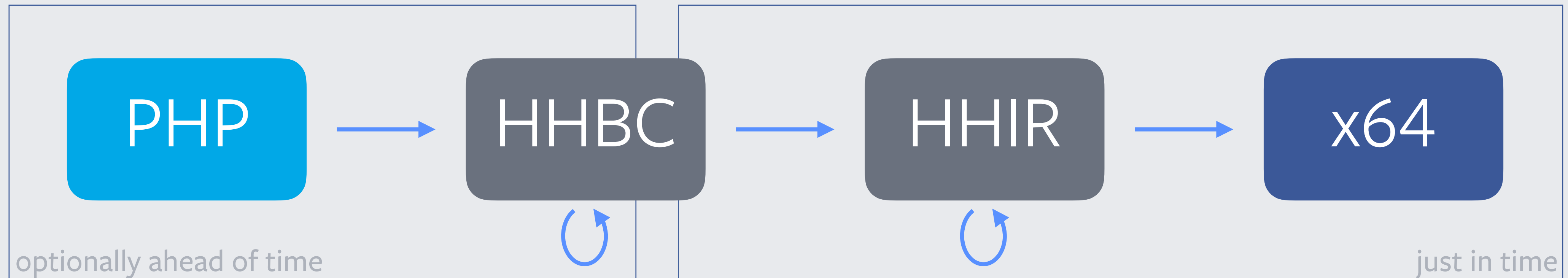


```
63  SetL L:4
65  PopC
66  Int 0
75  CGetL2 L:4
77  Gt
78  JmpZ 13 (91)
```

```
63: SetL L:4
    (12) t3:Int = LdStk<Int,IRSP0ff 0> t1:StkPtr
    (14) StLoc<4> t0:FramePtr, t3:Int
66: Int 0
    (21) StStk<IRSP0ff 0> t1:StkPtr, 0
75: CGetL2 L:4
    (24) StStk<IRSP0ff 0> t1:StkPtr, t3:Int
    (25) StStk<IRSP0ff -1> t1:StkPtr, 0
```

Compilation pipeline

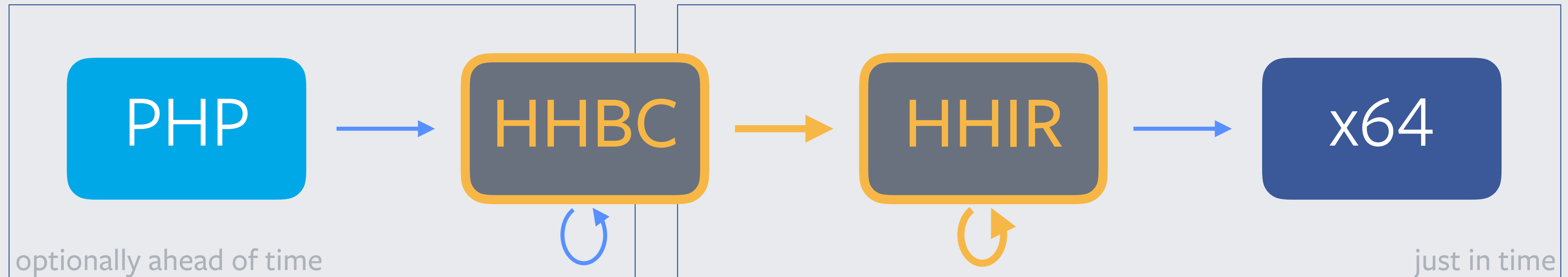
Bytecode-to-bytecode transformation



```
63 SetL L:4
65 PopC
66 Int 0
75 CGetL2 L:4
77 Gt
78 JmpZ 13 (91)
```

```
63: SetL L:4
    (12) t3:Int = LdStk<Int,IRSP0ff 0> t1:StkPtr
    (14) StLoc<4> t0:FramePtr, t3:Int
66: Int 0
    (21) StStk<IRSP0ff 0> t1:StkPtr, 0
75: CGetL2 L:4
    (24) StStk<IRSP0ff 0> t1:StkPtr, t3:Int
    (25) StStk<IRSP0ff -1> t1:StkPtr, 0
```

Compilation pipeline

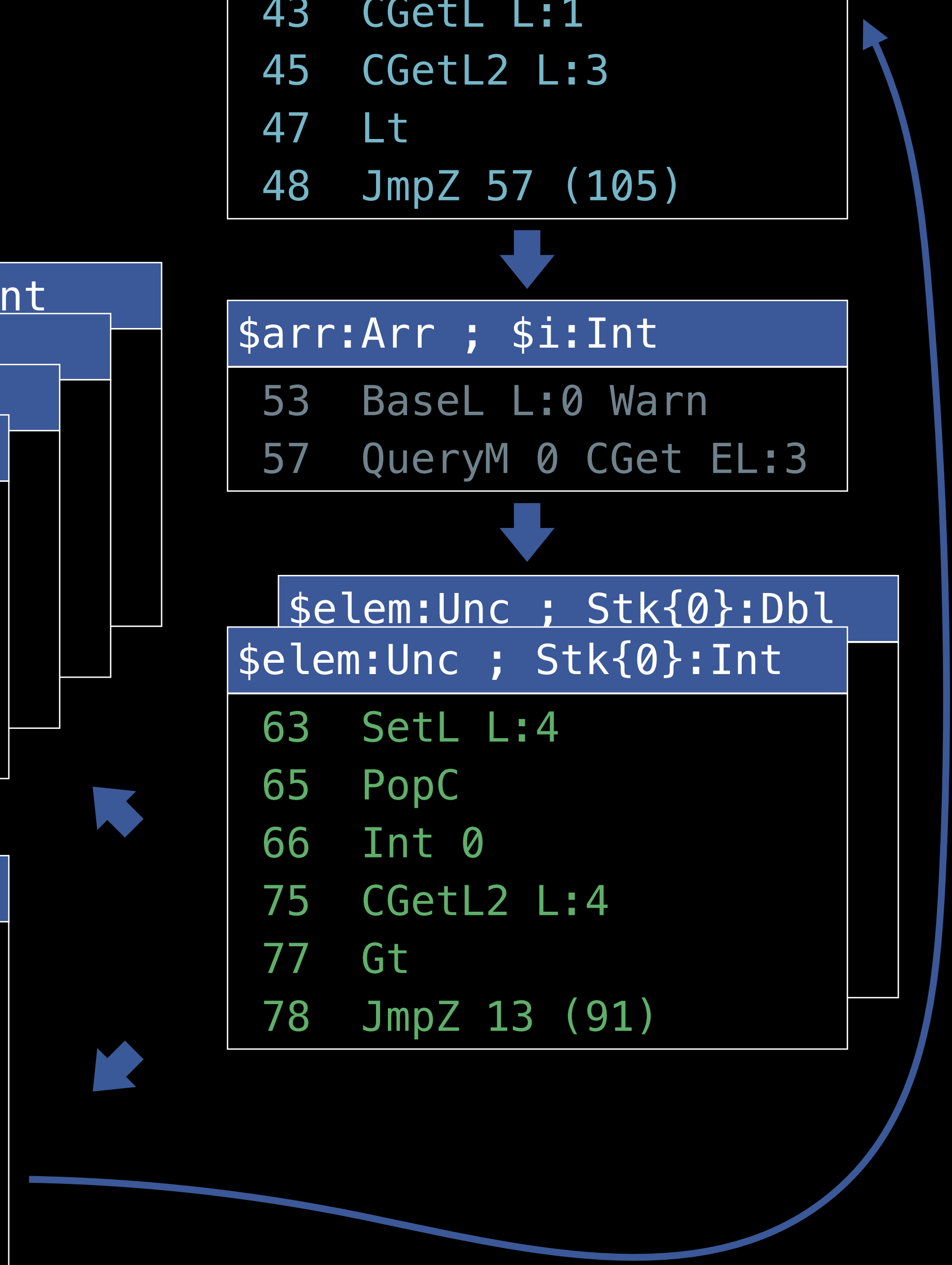
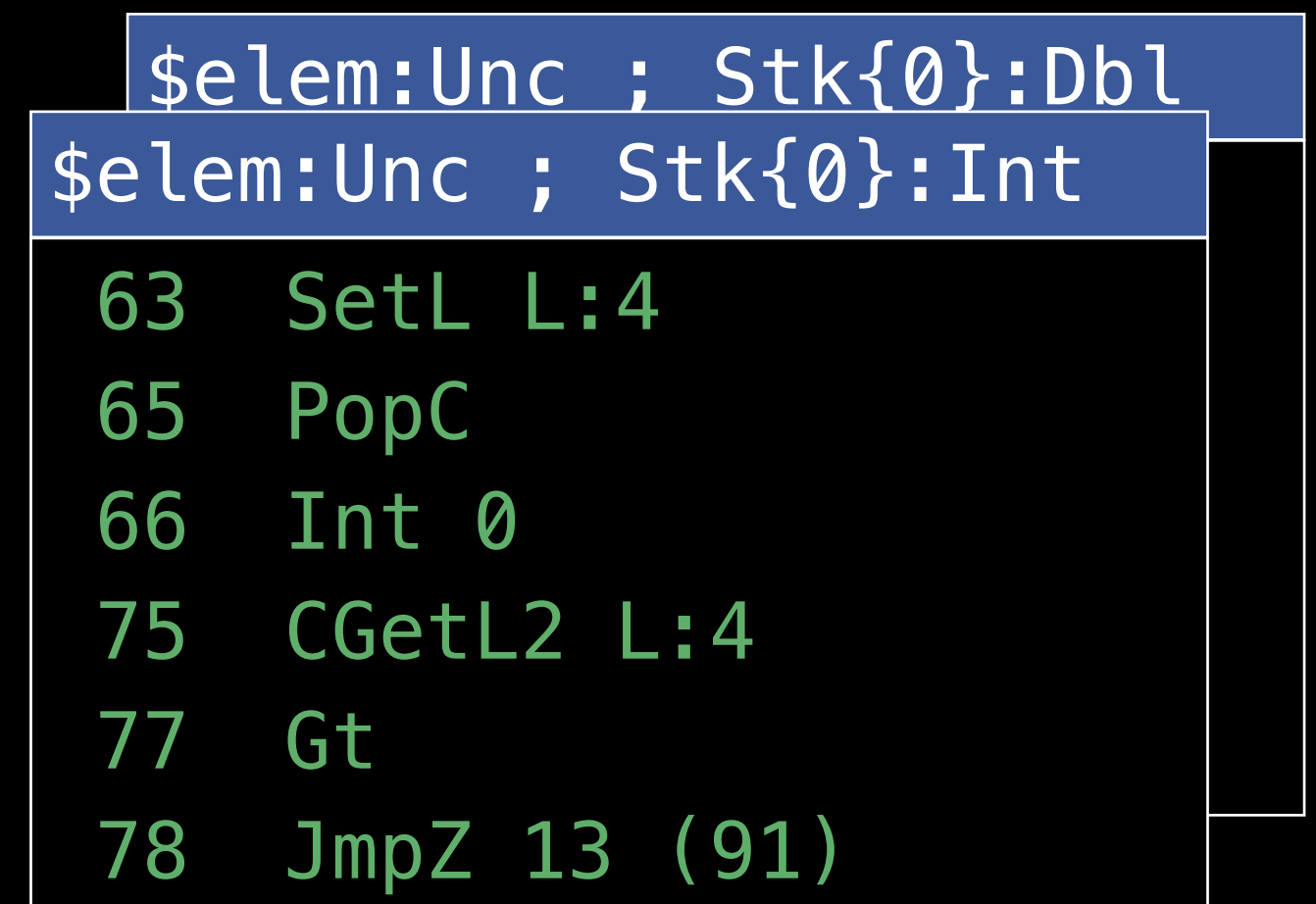
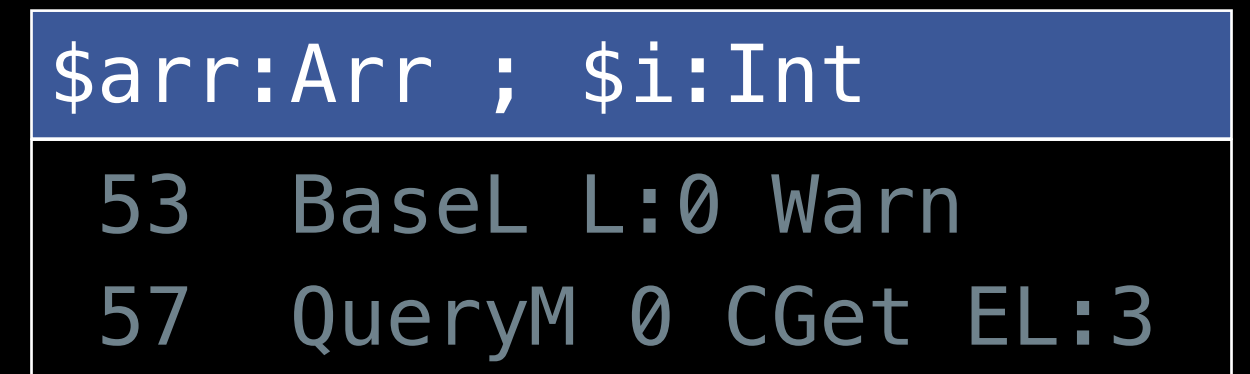
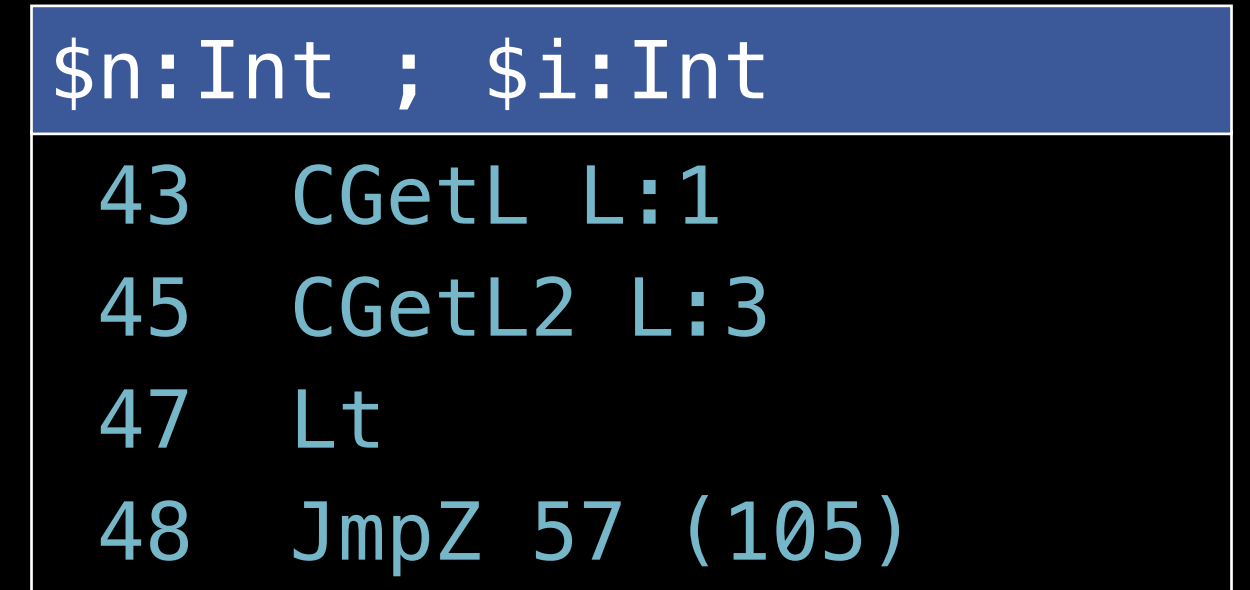
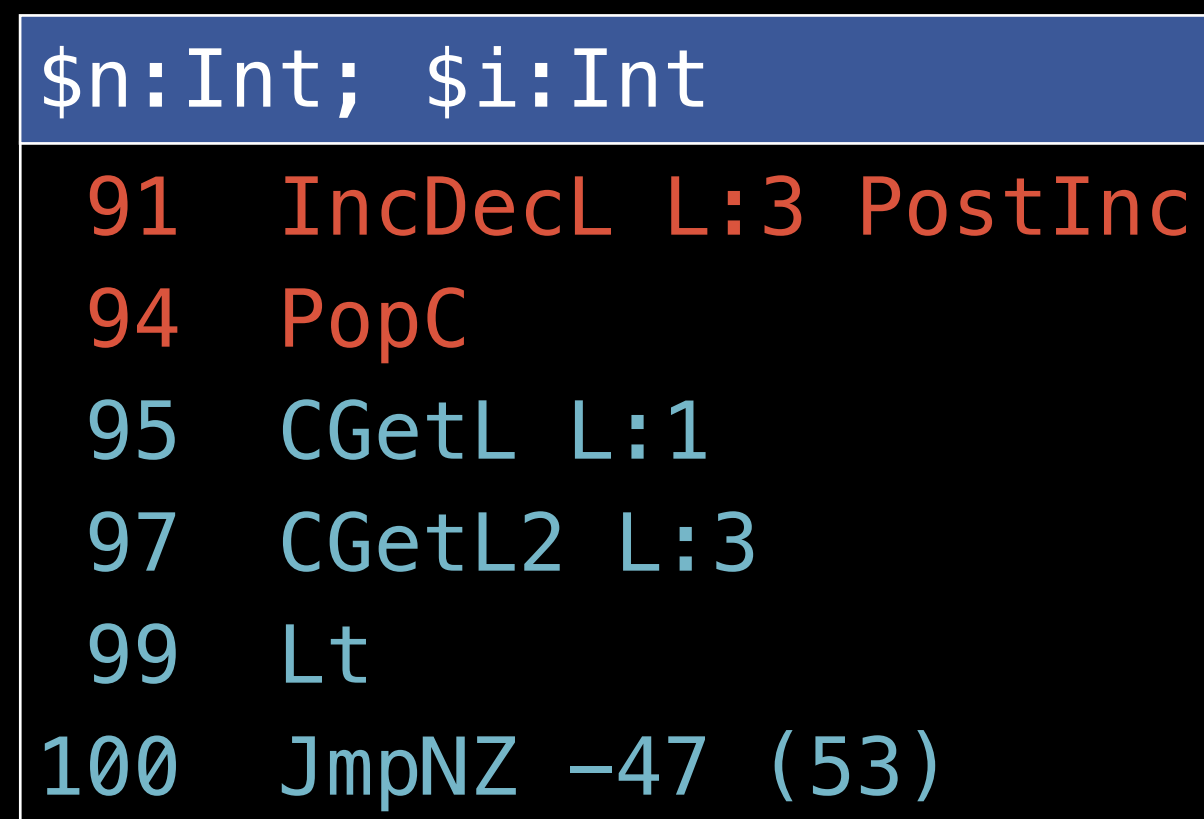
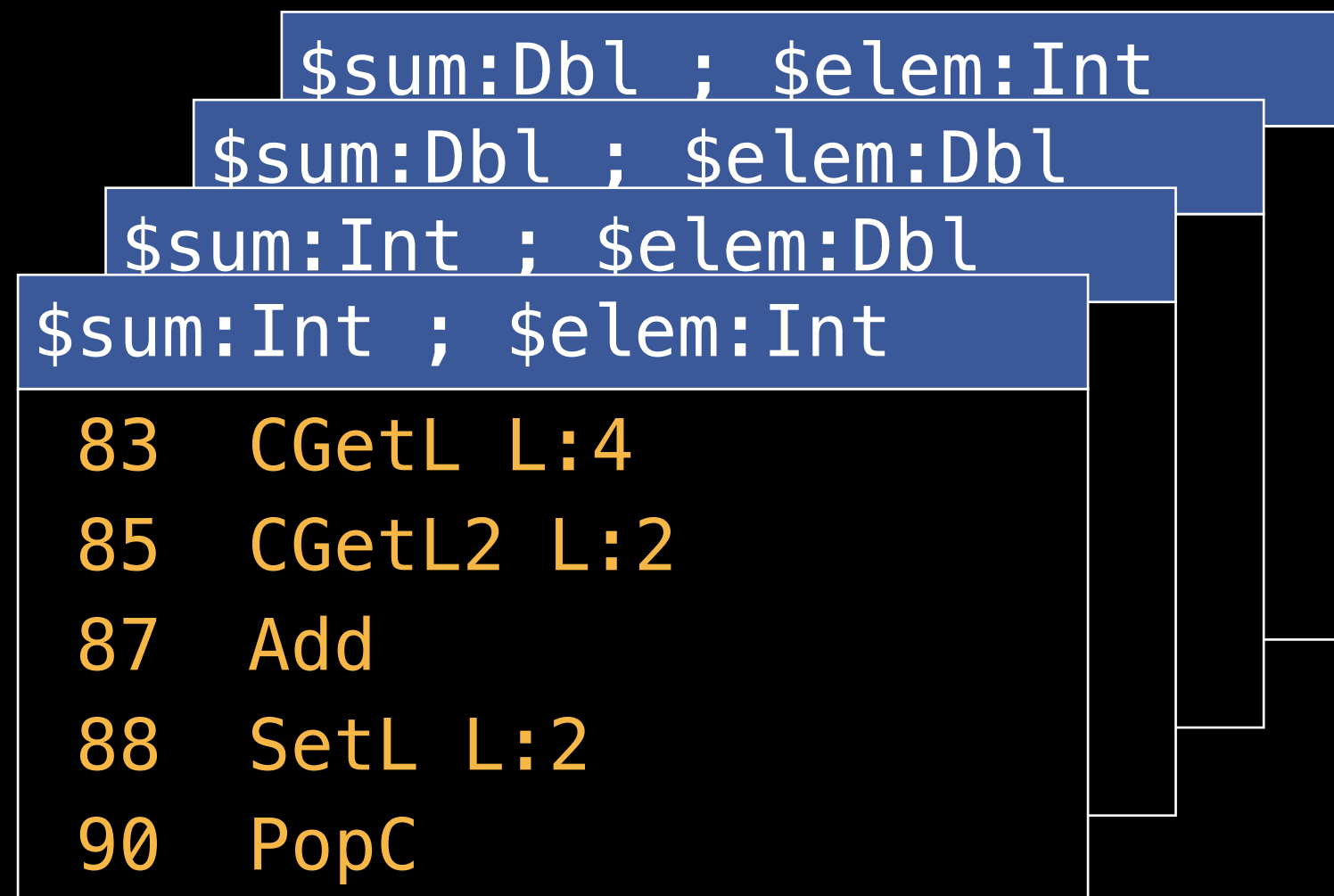


- This is where the magic happens

JIT optimizations

Dynamic type specialization

```
function addPositive($arr, $n) {  
    $sum = 0;  
    for ($i = 0; $i < $n; $i++) {  
        $elem = $arr[$i];  
        if ($elem > 0) {  
            $sum = $sum + $elem;  
        }  
    }  
    return $sum;  
}
```



JIT optimizations

Profile-guided optimization

```
function addPositive($arr, $n) {  
    $sum = 0;  
    for ($i = 0; $i < $n; $i++) {  
        $elem = $arr[$i];  
        if ($elem > 0) {  
            $sum = $sum + $elem;  
        }  
    }  
    return $sum;  
}
```

\$sum:Dbl ; \$elem:Int
\$sum:Dbl ; \$elem:Dbl
\$sum:Int ; \$elem:Dbl
\$sum:Int ; \$elem:Int

```
83 CGetL L:4  
85 CGetL2 L:2  
87 Add  
88 SetL L:2  
90 PopC
```

\$n:Int; \$i:Int

```
91 IncDecL L:3 PostInc  
94 PopC  
95 CGetL L:1  
97 CGetL2 L:3  
99 Lt  
100 JmpNZ -47 (53)
```

\$n:Int ; \$i:Int

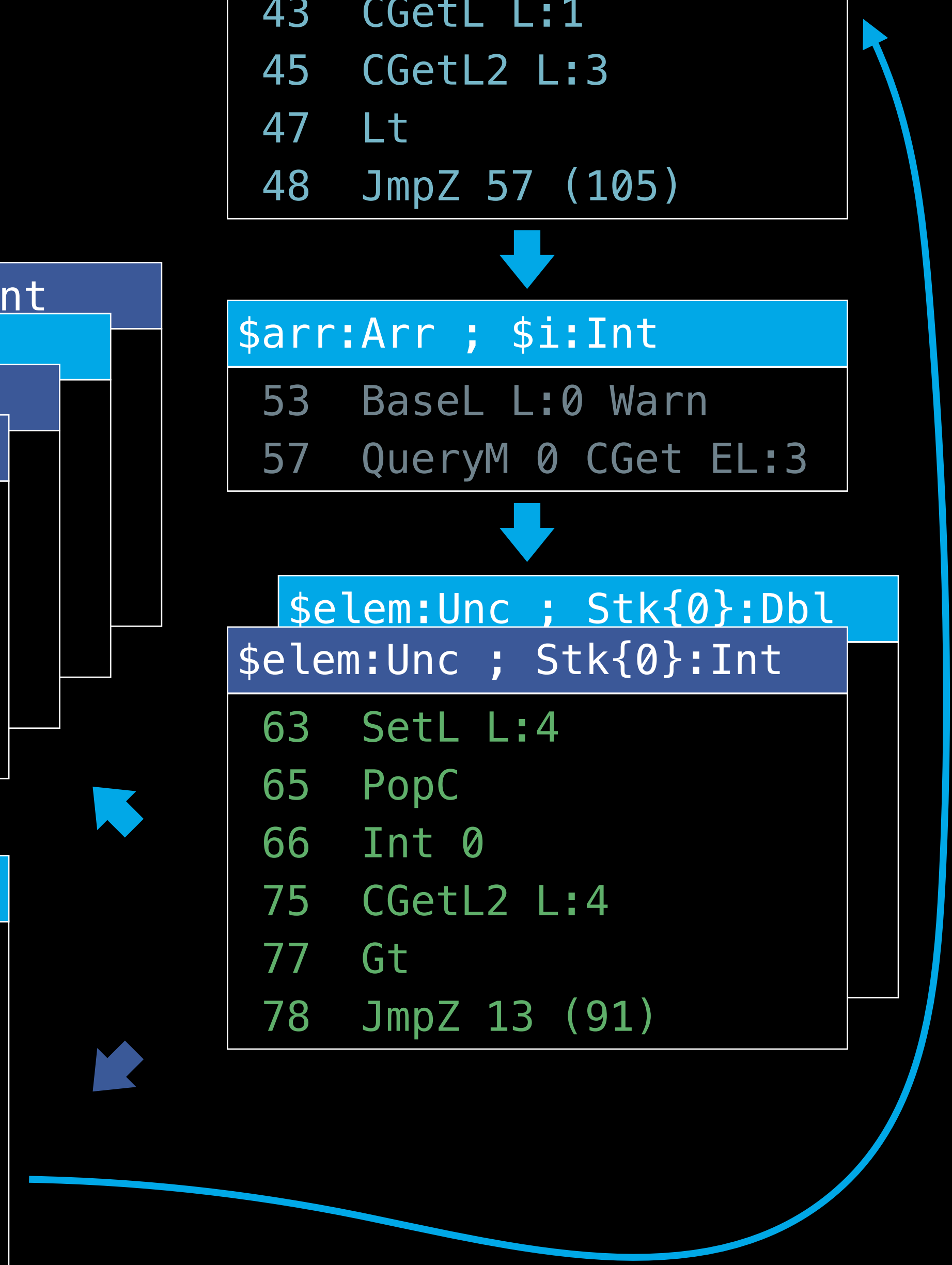
```
43 CGetL L:1  
45 CGetL2 L:3  
47 Lt  
48 JmpZ 57 (105)
```

\$arr:Arr ; \$i:Int

```
53 BaseL L:0 Warn  
57 QueryM 0 CGet EL:3
```

\$elem:Unc ; Stk{0}:Dbl
\$elem:Unc ; Stk{0}:Int

```
63 SetL L:4  
65 PopC  
66 Int 0  
75 CGetL2 L:4  
77 Gt  
78 JmpZ 13 (91)
```



JIT optimizations

Profile-guided optimization

```
function addPositive($arr, $n) {  
    $sum = 0;  
    for ($i = 0; $i < $n; $i++) {  
        $elem = $arr[$i];  
        if ($elem > 0) {  
            $sum = $sum + $elem;  
        }  
    }  
    return $sum;  
}
```

\$sum:Db1 ; \$elem:Db1	
83	CGetL L:4
85	CGetL2 L:2
87	Add
88	SetL L:2
90	PopC
91	IncDecL L:3 PostInc
94	PopC
95	CGetL L:1
97	CGetL2 L:3
99	Lt
100	JmpNZ -47 (53)

\$n:Int ; \$i:Int	
\$arr:Arr ; \$i:Int	
43	CGetL L:1
45	CGetL2 L:3
47	Lt
48	JmpZ 57 (105)
53	BaseL L:0 Warn
57	QueryM 0 CGet EL:3
\$elem:Unc ; Stk{0}:Db1	
63	SetL L:4
65	PopC
66	Int 0
75	CGetL2 L:4
77	Gt
78	JmpZ 13 (91)

JIT optimizations

HHIR optimization passes

```
$c = $a . $b;  
$len = strlen($c);
```

```
69: CGetL L:1  
    (12) t3:Str = LdLoc<Str,1> t0:FramePtr  
    (13) IncRef t3:Str  
71: CGetL2 L:0  
    (16) t4:Str = LdLoc<Str,0> t0:FramePtr  
    (17) IncRef t4:Str  
73: Concat  
    (22) t5:Str = ConcatStrStr t4:Str, t3:Str  
    (24) DecRef<-> t3:Str  
74: SetL L:2  
    (27) StLoc<2> t0:FramePtr, t5:Str  
    (28) IncRef t5:Str  
76: PopC  
    (31) DecRef<-> t5:Str  
77: CGetL L:2  
    (33) IncRef t5:Str  
79: FCallBuiltin 1 1 "strlen"  
    (35) t7:Int = LdStrLen t5:Str  
    (36) DecRef<-> t5:Str
```


JIT optimizations

HHIR optimization passes

```
$c = $a . $b;  
$len = strlen($c);
```

```
69: CGetL L:1  
    (12) t3:Str = LdLoc<Str,1> t0:FramePtr  
    (13) IncRef t3:Str  
71: CGetL2 L:0  
    (16) t4:Str = LdLoc<Str,0> t0:FramePtr  
    (17) IncRef t4:Str  
73: Concat  
    (22) t5:Str = ConcatStrStr t4:Str, t3:Str  
    (24) DecRef<-> t3:Str  
74: SetL L:2  
    (27) StLoc<2> t0:FramePtr, t5:Str  
    (28) IncRef t5:Str  
76: PopC  
    (31) DecRef<-> t5:Str  
77: CGetL L:2  
    (33) IncRef t5:Str  
79: FCallBuiltin 1 1 "strlen"  
    (35) t7:Int = LdStrLen t5:Str  
    (36) DecRef<-> t5:Str
```

JIT optimizations

HHIR optimization passes

```
$c = $a . $b;  
$len = strlen($c);
```

```
69: CGetL L:1  
    (12) t3:Str = LdLoc<Str,1> t0:FramePtr  
    (13) Nop  
71: CGetL2 L:0  
    (16) t4:Str = LdLoc<Str,0> t0:FramePtr  
    (17) IncRef t4:Str  
73: Concat  
    (22) t5:Str = ConcatStrStr t4:Str, t3:Str  
    (24) Nop  
74: SetL L:2  
    (27) StLoc<2> t0:FramePtr, t5:Str  
    (28) Nop  
76: PopC  
    (31) Nop  
77: CGetL L:2  
    (33) Nop  
79: FCallBuiltin 1 1 "strlen"  
    (35) t7:Int = LdStrLen t5:Str  
    (36) Nop
```

JIT optimizations

HHIR optimization passes

```
$c = $a . $b;  
$len = strlen($c);
```

```
69: CGetL L:1  
    (12) t3:Str = LdLoc<Str,1> t0:FramePtr  
71: CGetL2 L:0  
    (16) t4:Str = LdLoc<Str,0> t0:FramePtr  
    (17) IncRef t4:Str  
73: Concat  
    (22) t5:Str = ConcatStrStr t4:Str, t3:Str  
74: SetL L:2  
    (27) StLoc<2> t0:FramePtr, t5:Str  
79: FCallBuiltin 1 1 "strlen"  
    (35) t7:Int = LdStrLen t5:Str
```

JIT optimizations

HHIR optimization passes

`/* ... */`

Agenda

i.e., Who did all the hard work?

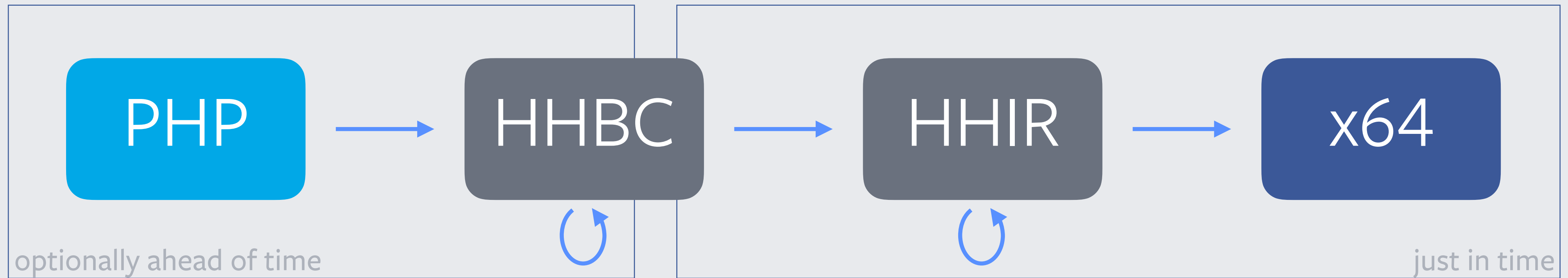
1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

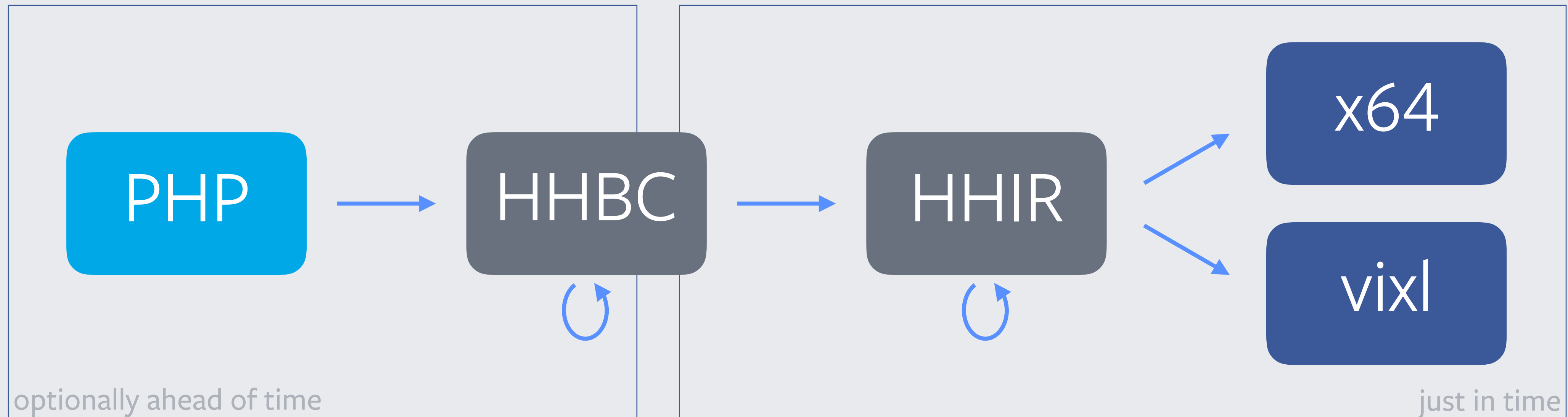
4 Where do we go from here?

Compilation pipeline



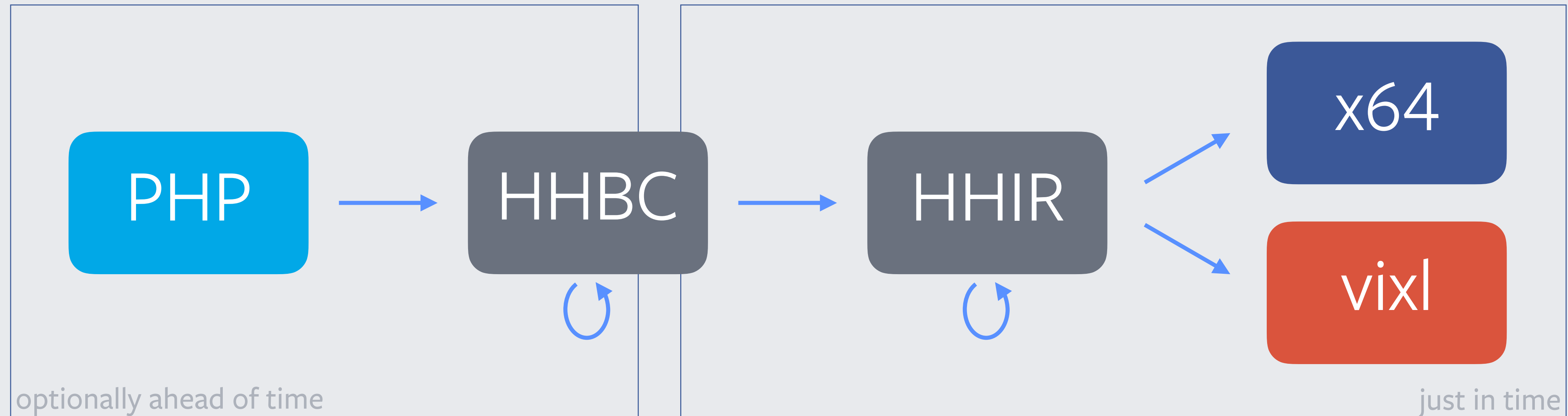
Compilation pipeline

ARM simulator



Compilation pipeline

ARM simulator

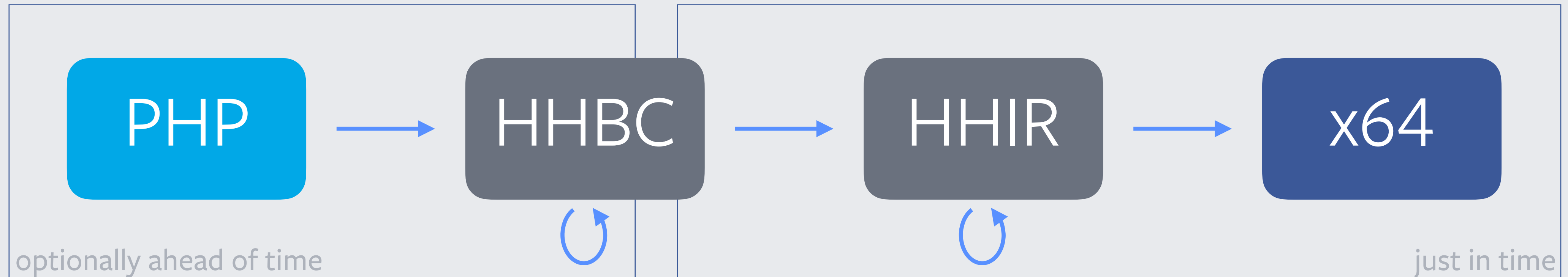


- Maintenance nightmare

- > 600 HHIR ops: 🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝🍝

- We aren't ARM experts

Compilation pipeline



```
63  SetL L:4
65  PopC
66  Int 0
75  CGetL2 L:4
77  Gt
78  JmpZ 13 (91)
```

```
63: SetL L:4
    (12) t3:Int = LdStk<Int,IRSP0ff 0> t1:StkPtr
    (14) StLoc<4> t0:FramePtr, t3:Int
66: Int 0
    (21) StStk<IRSP0ff 0> t1:StkPtr, 0
75: CGetL2 L:4
    (24) StStk<IRSP0ff 0> t1:StkPtr, t3:Int
    (25) StStk<IRSP0ff -1> t1:StkPtr, 0
```

Compilation pipeline

Virtual assembly



```
63: SetL L:4
(12) t3:Int = LdStk<Int,IRSP0ff 0> t1:StkPtr
(14) StLoc<4> t0:FramePtr, t3:Int
66: Int 0
(21) StStk<IRSP0ff 0> t1:StkPtr, 0
75: CGetL2 L:4
(24) StStk<IRSP0ff 0> t1:StkPtr, t3:Int
(25) StStk<IRSP0ff -1> t1:StkPtr, 0
```

```
load [%128] => %129
storeb %136(17b), [%rbp - 0x48]
store %129, [%rbp - 0x50]
storeb %136(17b), [%128 + 0x8]
store %129, [%128]
```

Compilation pipeline

Virtual assembly



- Uncanny resemblance to x64
- Spiritual sibling of WebKit's Bare Bones Backend

Compilation pipeline

LLVM? Have you heard of it?



- “Why don’t you just use LLVM?” 🤔
- We tried it:
 - No noticeable performance gains
 - LLVM’s MCJIT is too heavyweight

Compilation pipeline

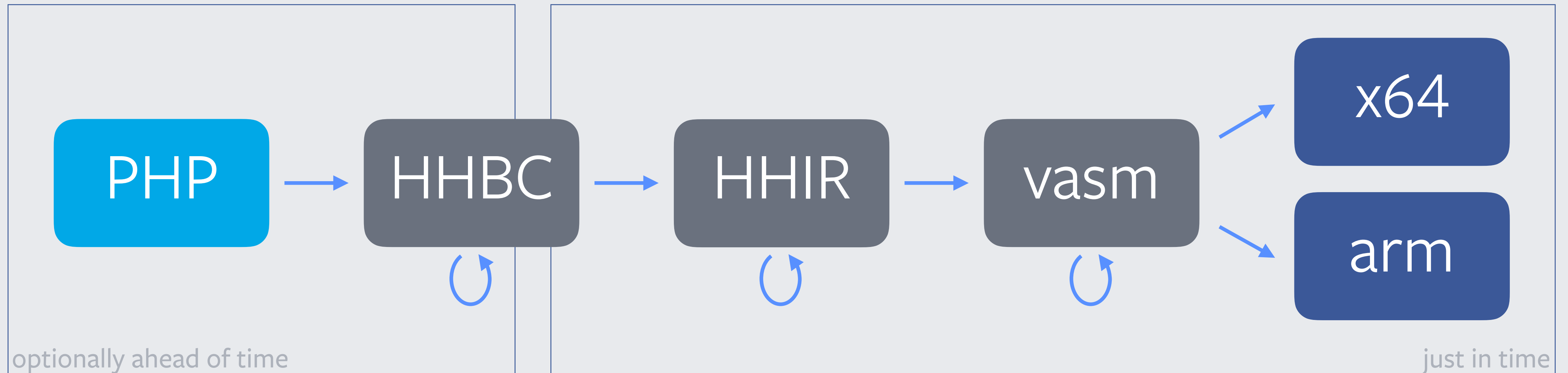
LLVM? Have you heard of it?



- Experimental LLVM backend stress-tested vasm
 - Calling conventions
 - Register widths
 - ...

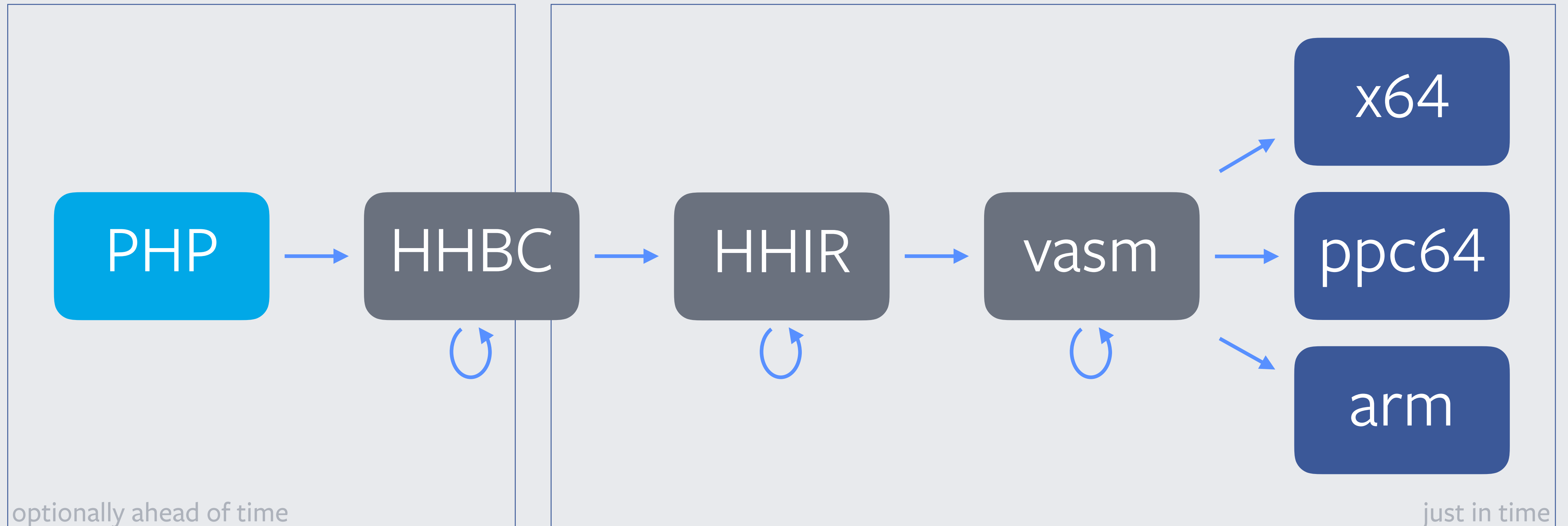
Compilation pipeline

ARM backend



Compilation pipeline

Backends for everyone!



Contributors

The most important slide in this talk

- Lakshmi Pathy — @lpathy
- Dave Estes — @dave-estes
- Jim Saxman — @jim-saxman
- Christoph Müllner — @cmuellner
- Steve Walk — @swalk-cavium
- Andrew Pinski — @apinski-cavium
- ...

ARM backend

Baseline functionality

- Final vasm-to-AArch64 lowering pass
- Code smashing
- Boundary-crossing b/w C++ and jitted code
- Bonus: Continuous integration testing!

ARM backend

Optimizations

- Strength reduction on flag-setting instructions
- 64-bit immediate lifting
- Branch offset optimizations
- ...

Agenda

Hold onto your backends

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

Agenda

i.e., How can you get involved?

1 What is HHVM?

2 How did we get running on AArch64?

3 Will the demo work?

4 Where do we go from here?

Future work

Code size and layout

- Cache is king!
- Instruction sequences generally larger on AArch64
- HHVM is sensitive to code layout
 - Huge pages
 - Indirect branch rewriting
 - Locality tuning
 - ...

Future work

More ARM-specific optimizations

- Profile OSS workloads on ARM using perf
 - <https://github.com/hhvm/oss-performance>
- Make some measurements

Resources

Feel free to contribute!

- Website: <http://hhvm.com/>
- GitHub: <https://github.com/facebook/hhvm>
- IRC: #hphp-dev on Freenode

- Mailing list: <https://groups.google.com/d/forum/hhvm-arm>
- My email: mwang@fb.com

Quick recap

HHVM on AArch64

Quick recap

HHVM on AArch64

1

HHVM

Quick recap

HHVM on AArch64

1

HHVM

2

It runs on AArch64 (thanks to the community)

Quick recap

HHVM on AArch64

1

HHVM

2

It runs on AArch64 (thanks to the community)

3

Seriously, the demo worked and everything

Any questions?

Any questions?

1

HHVM

2

It runs on AArch64 (thanks to the community)

3

Seriously, the demo worked and everything

4

Any questions?

facebook