A Maiden Voyage into Open Source, Datatypes, and Scala

L Denney

About Me

- Finished my undergrad in May 2026
- Functional Programming <3
- Interested in Open Source development
- Typelevel

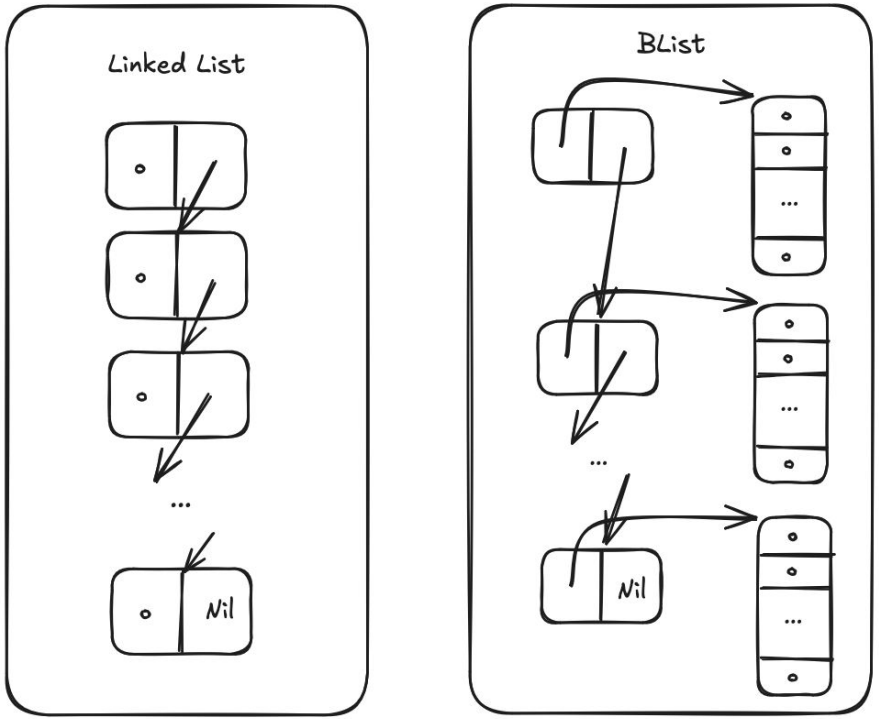


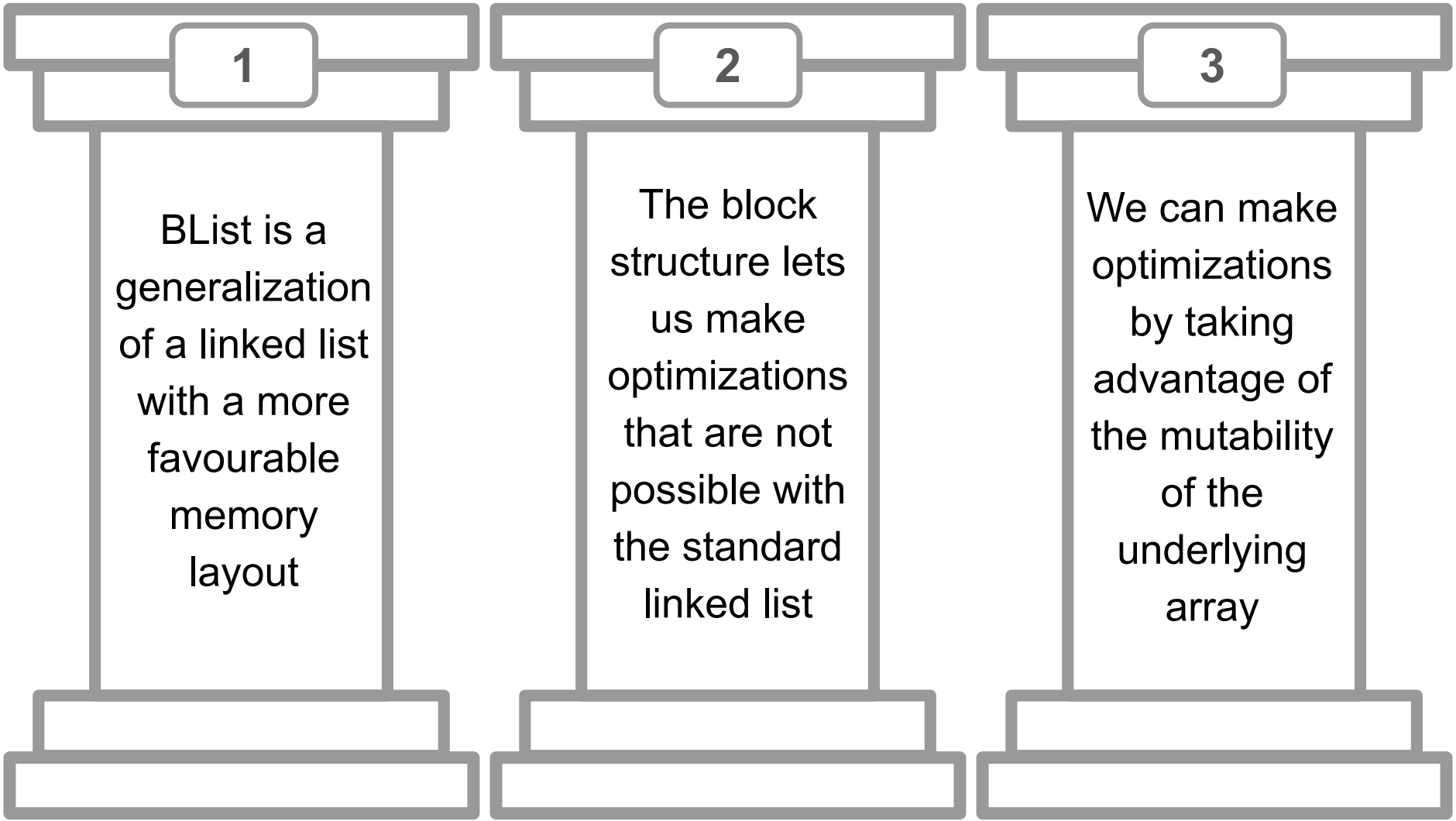
About Me

- Finished my undergrad in May 2026
- Functional Programming <3
- Interested in Open Source development
- Typelevel

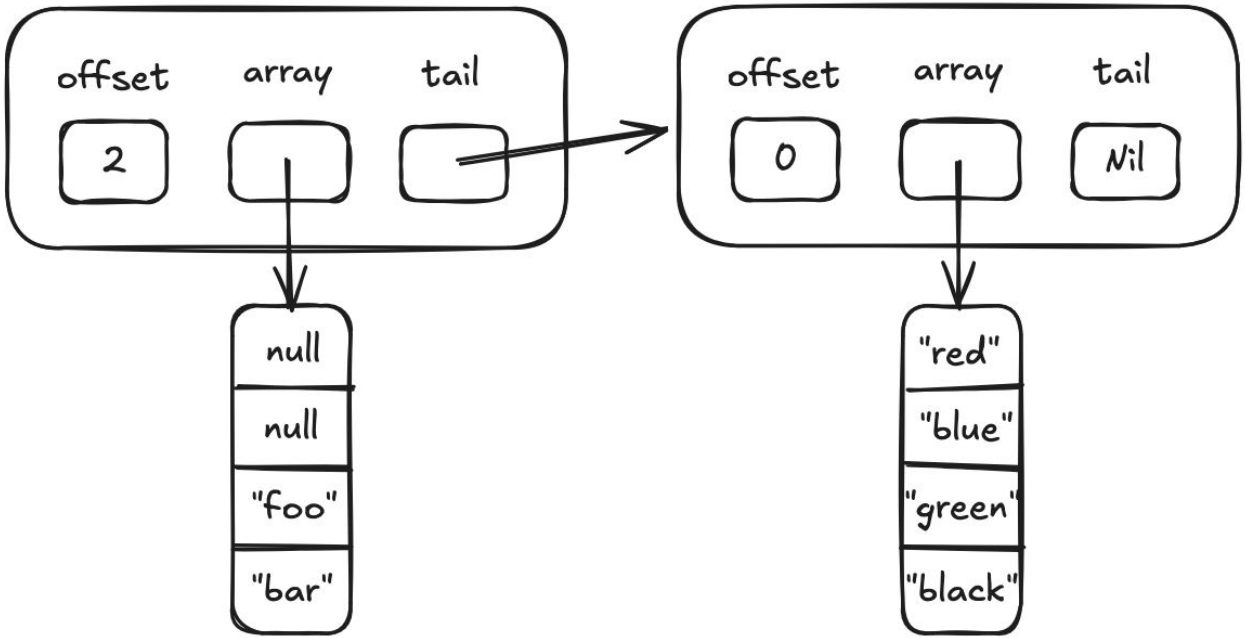


My Project: BList




1
BList is a generalization of a linked list with a more favourable memory layout
2
The block structure lets us make optimizations that are not possible with the standard linked list
3
We can make optimizations by taking advantage of the mutability of the underlying array

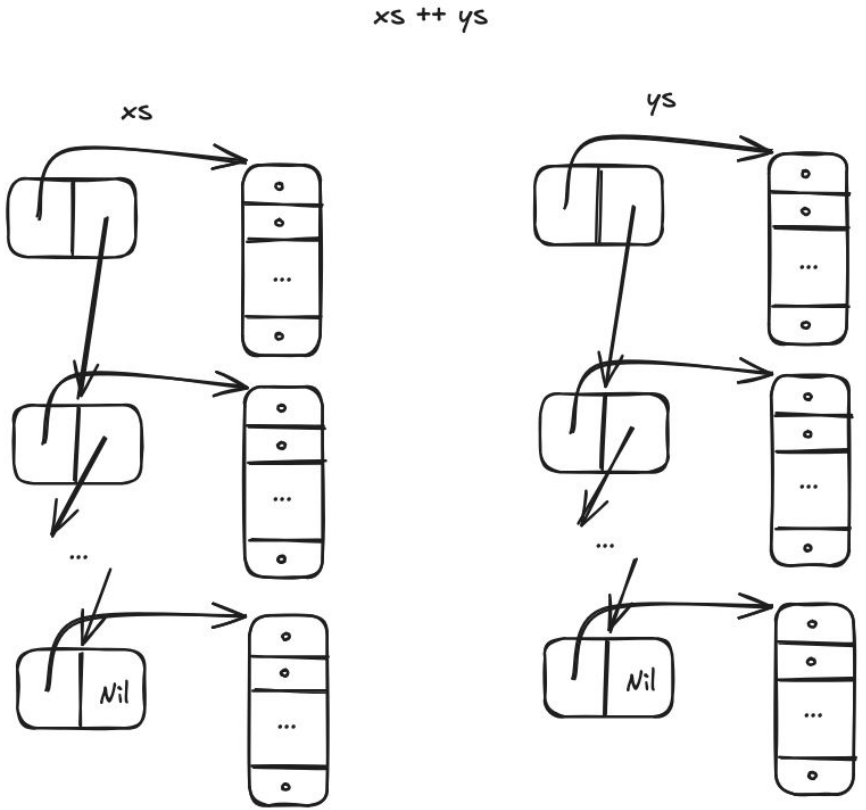
BList example with a block size of 4



“BList is a generalization of linked list with a more favourable memory layout”

Methods

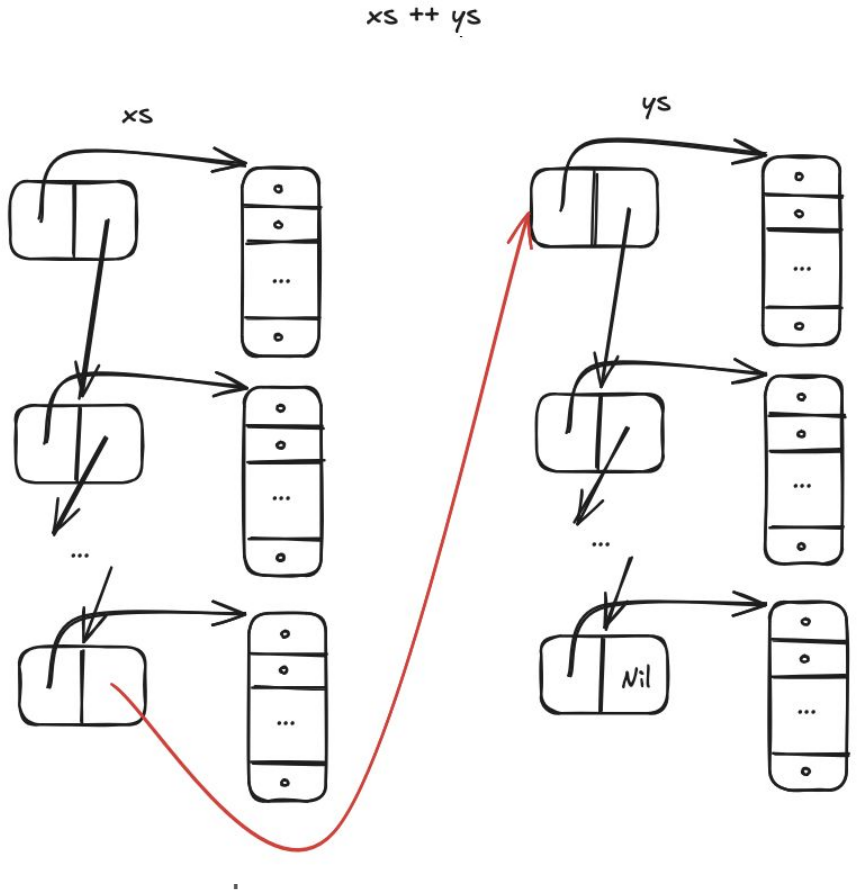
- prepend, head, tail, get, etc.
- map, filter, folds, etc.
- splitAt, reverse, concat
- drop, take, etc.
- flatMap, forEach
- toList, fromList
- iterator



“The block structure lets us make optimizations that are not possible with the standard linked list”

Methods

- prepend, head, tail, get, etc.
- map, filter, folds, etc.
- splitAt, reverse, concat
- drop, take, etc.
- flatMap, forEach
- toList, fromList
- iterator

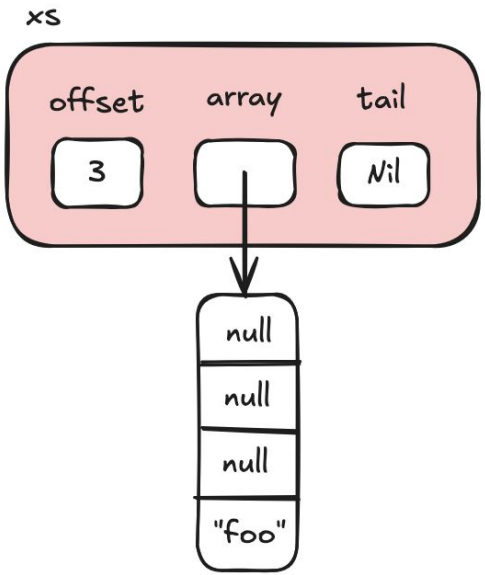


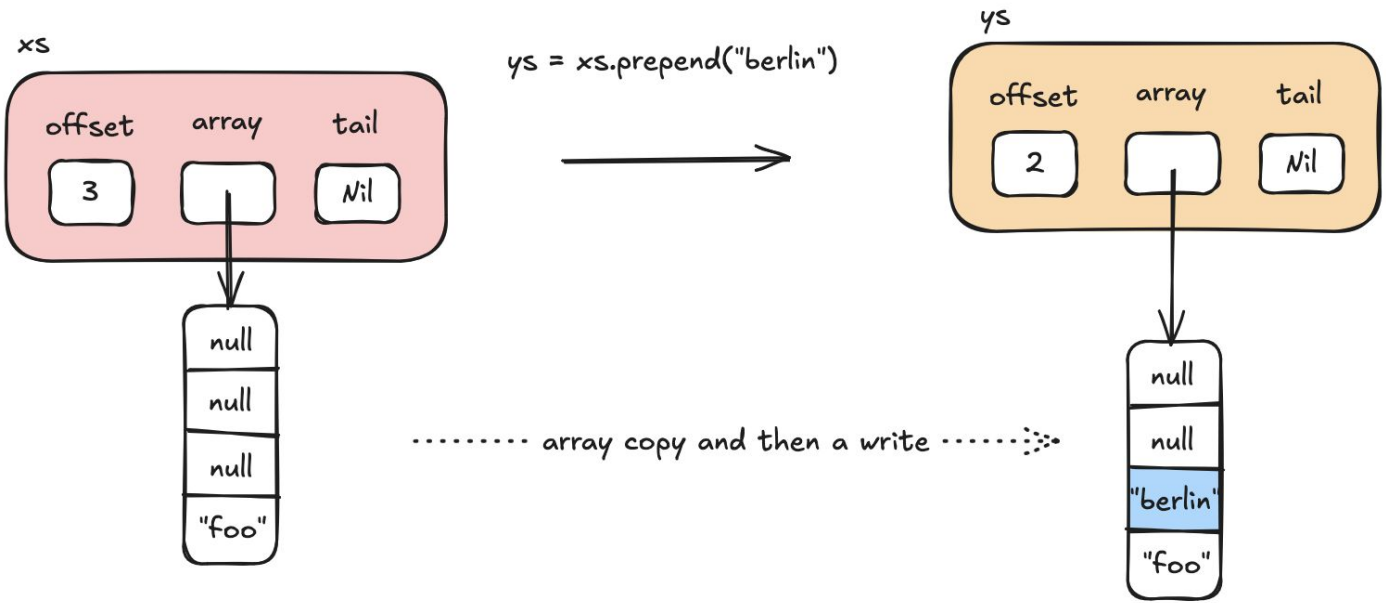
“The block structure lets us make optimizatic

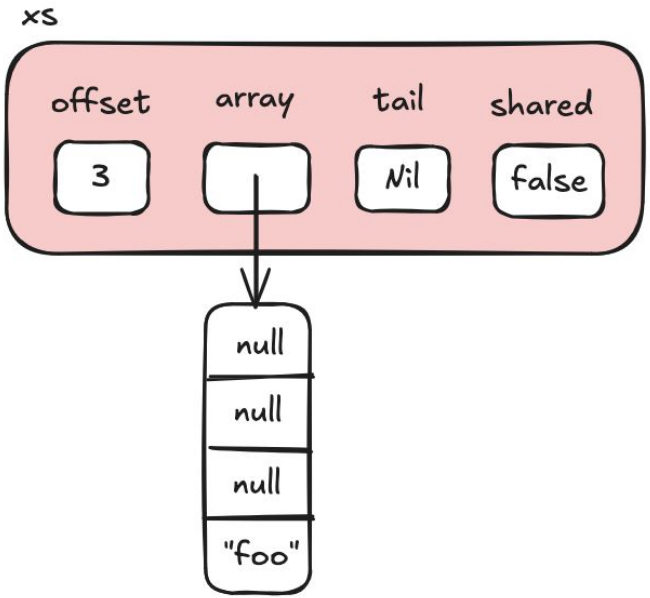
prepend

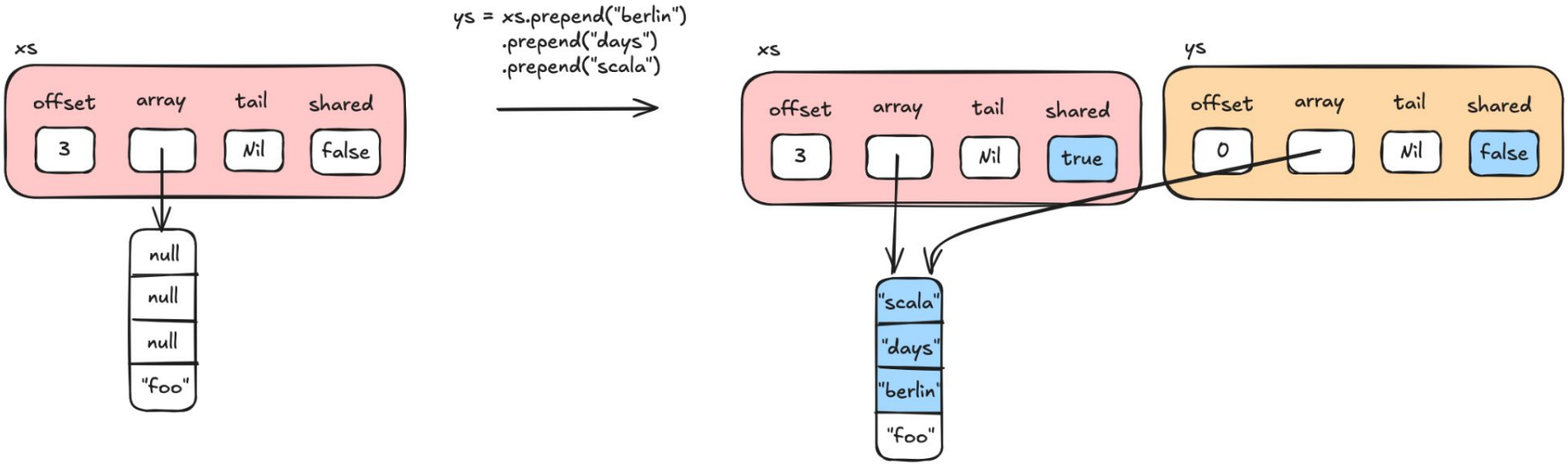
share and fork

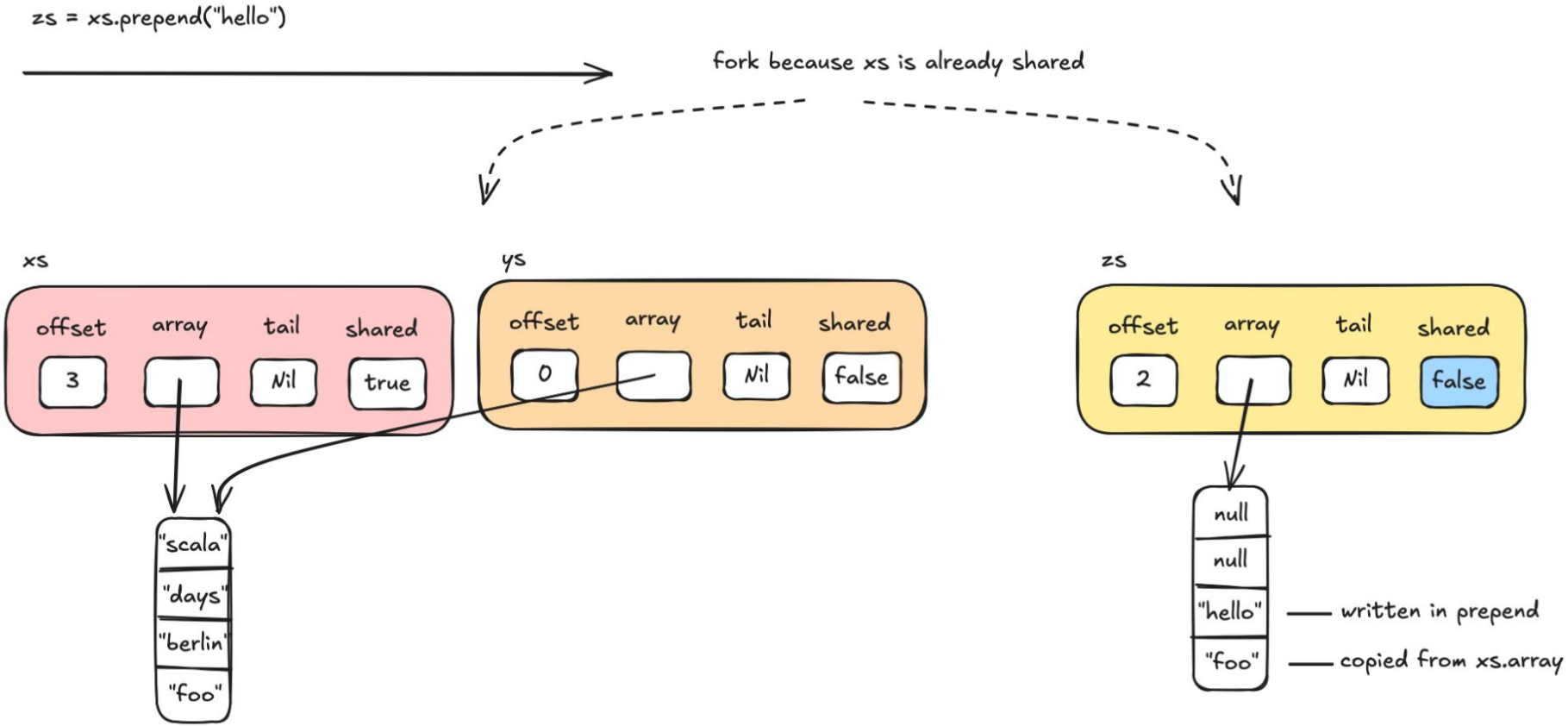
“We can make optimizations by taking advantage of the mutability of the underlying array”


xs
offset
array
tail
3
Nil
null
null
null
"foo"

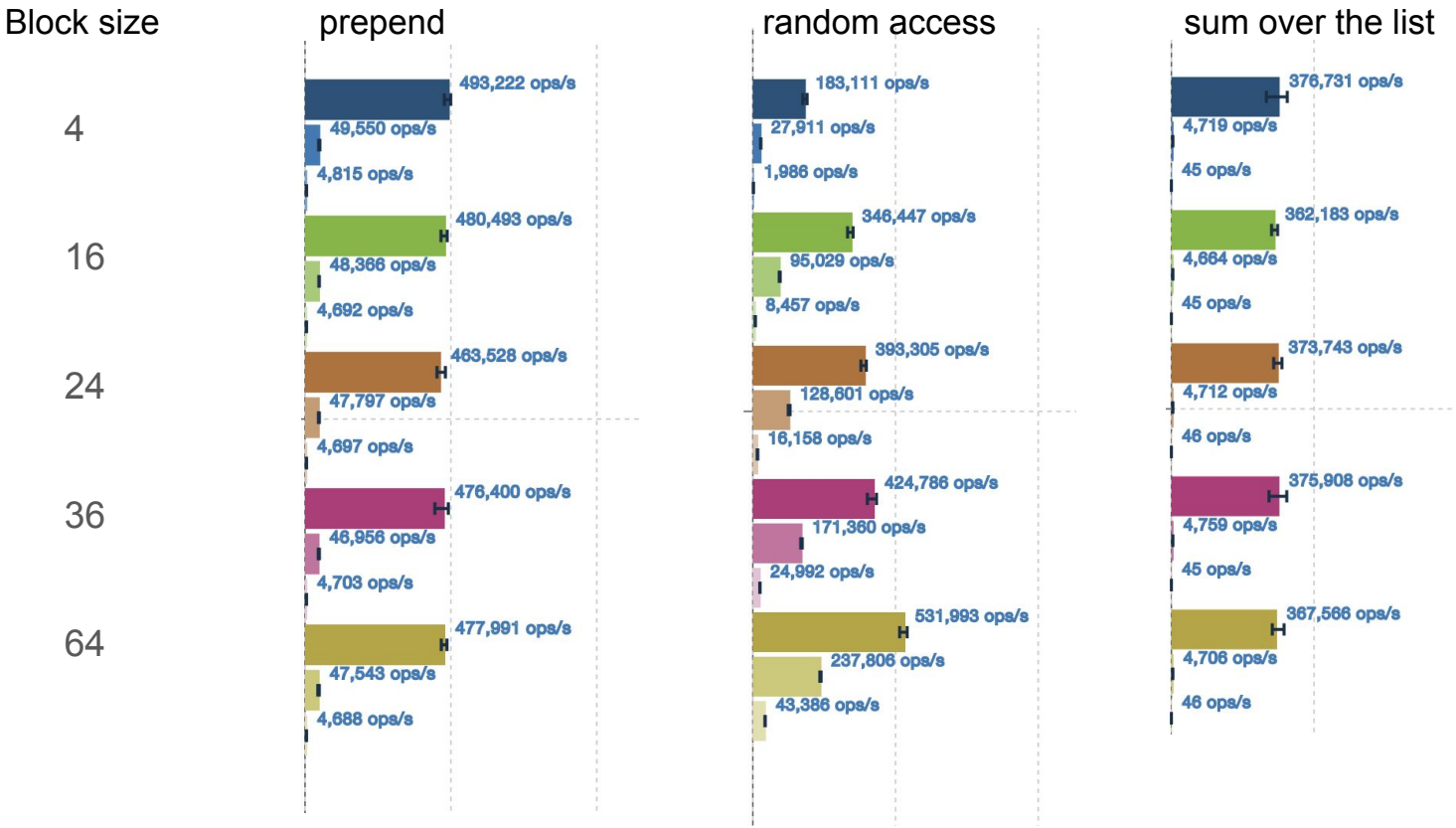

xs
offset
array
tail
3
Nil
null
null
null
"foo"
ys = xs.prepend("berlin")
array copy and then a write
ys
offset
array
tail
2
Nil
null
null
"berlin"
"foo"


xs
offset
array
tail
shared
3
Nil
false
null
null
null
"foo"

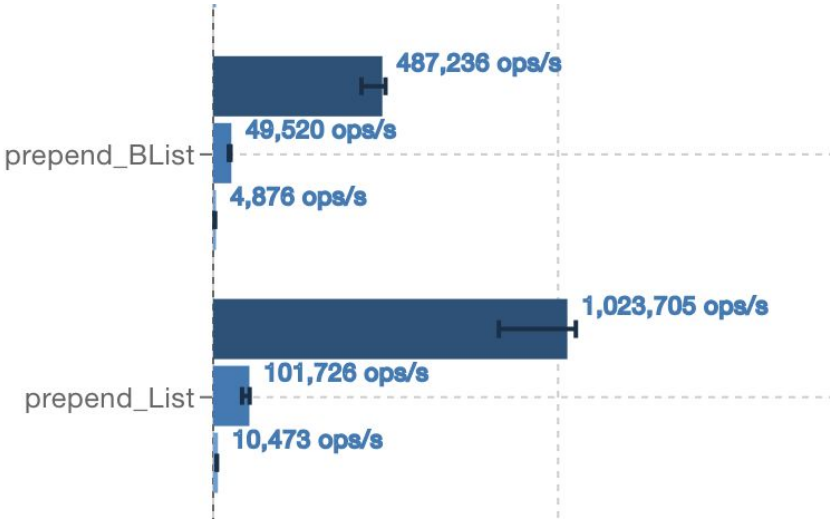

xs
offset
array
tail
shared
3
Nil
false
null
null
null
"foo"
ys = xs.prepend("berlin")
.prepend("days")
.prepend("scala")
xs
offset
array
tail
shared
3
Nil
true
ys
offset
array
tail
shared
0
Nil
false
"scala"
"days"
"berlin"
"foo"

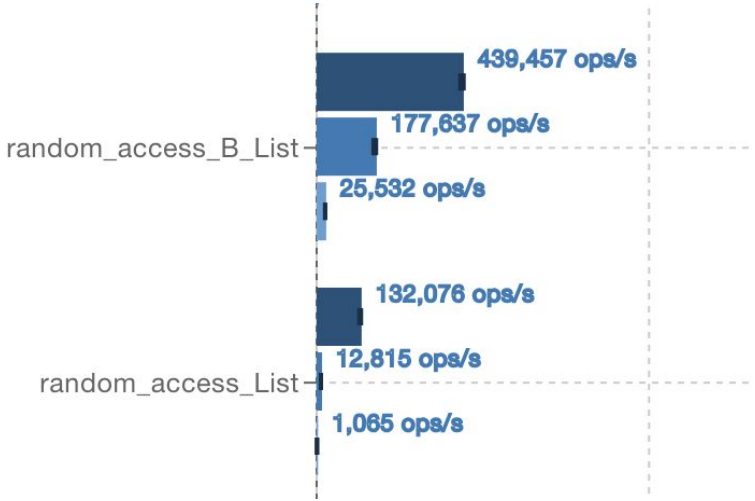

zs = xs.prepend("hello")
fork because xs is already shared
xs
offset
array
tail
shared
3
Nil
true
ys
offset
array
tail
shared
0
Nil
false
zs
offset
array
tail
shared
2
Nil
false
"scala"
"days"
"berlin"
"foo"
null
null
"hello"
written in prepend
"foo"
copied from xs.array

Design decisions: determining the block size



Benchmarking





Thank you

This work can be found in PR #837 in Typelevel's cats-collections