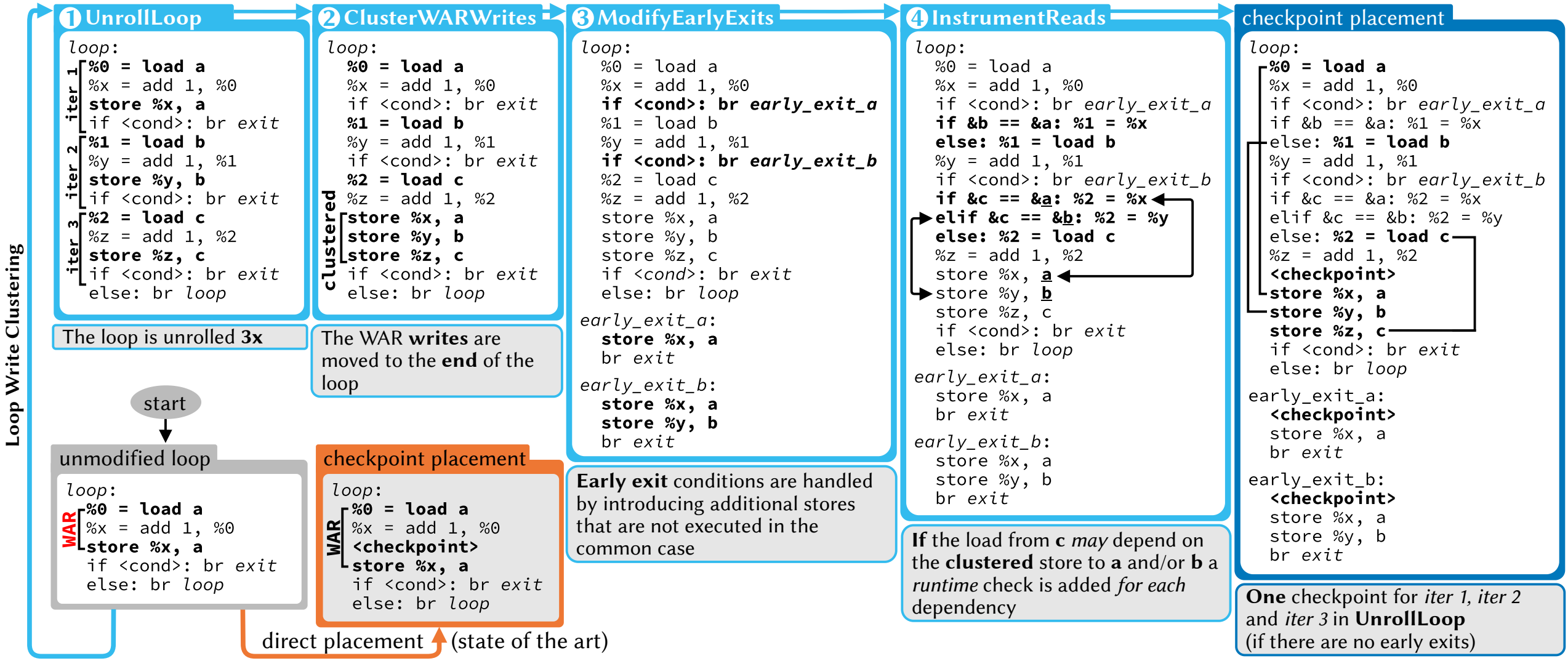
Loop Write Clustering
1 UnrollLoop
loop:
iter 1
%0 = load a
%x = add 1, %0
store %x, a
if <cond>: br exit
iter 2
%1 = load b
%y = add 1, %1
store %y, b
if <cond>: br exit
iter 3
%2 = load c
%z = add 1, %2
store %z, c
if <cond>: br exit
else: br loop
The loop is unrolled 3x
2 ClusterWARWrites
loop:
%0 = load a
%x = add 1, %0
if <cond>: br exit
%1 = load b
%y = add 1, %1
if <cond>: br exit
%2 = load c
%z = add 1, %2
clustered
store %x, a
store %y, b
store %z, c
if <cond>: br exit
else: br loop
The WAR writes are moved to the end of the loop
3 ModifyEarlyExits
loop:
%0 = load a
%x = add 1, %0
if <cond>: br early_exit_a
%1 = load b
%y = add 1, %1
if <cond>: br early_exit_b
%2 = load c
%z = add 1, %2
store %x, a
store %y, b
store %z, c
if <cond>: br exit
else: br loop
early_exit_a:
store %x, a
br exit
early_exit_b:
store %x, a
store %y, b
br exit
Early exit conditions are handled by introducing additional stores that are not executed in the common case
4 InstrumentReads
loop:
%0 = load a
%x = add 1, %0
if <cond>: br early_exit_a
if &b == &a: %1 = %x
else: %1 = load b
%y = add 1, %1
if <cond>: br early_exit_b
if &c == &a: %2 = %x
elif &c == &b: %2 = %y
else: %2 = load c
%z = add 1, %2
store %x, a
store %y, b
store %z, c
if <cond>: br exit
else: br loop
early_exit_a:
store %x, a
br exit
early_exit_b:
store %x, a
store %y, b
br exit
If the load from c may depend on the clustered store to a and/or b a runtime check is added for each dependency
checkpoint placement
loop:
%0 = load a
%x = add 1, %0
if <cond>: br early_exit_a
if &b == &a: %1 = %x
else: %1 = load b
%y = add 1, %1
if <cond>: br early_exit_b
if &c == &a: %2 = %x
elif &c == &b: %2 = %y
else: %2 = load c
%z = add 1, %2
<checkpoint>
store %x, a
store %y, b
store %z, c
if <cond>: br exit
else: br loop
early_exit_a:
<checkpoint>
store %x, a
br exit
early_exit_b:
<checkpoint>
store %x, a
store %y, b
br exit
One checkpoint for iter 1, iter 2 and iter 3 in UnrollLoop (if there are no early exits)
start
unmodified loop
loop:
WAR
%0 = load a
%x = add 1, %0
store %x, a
if <cond>: br exit
else: br loop
checkpoint placement
loop:
WAR
%0 = load a
%x = add 1, %0
<checkpoint>
store %x, a
if <cond>: br exit
else: br loop
direct placement
(state of the art)