



A fast cluster finder for the SDD

Simone Capodicasa

INFN - Turin

Offline Week

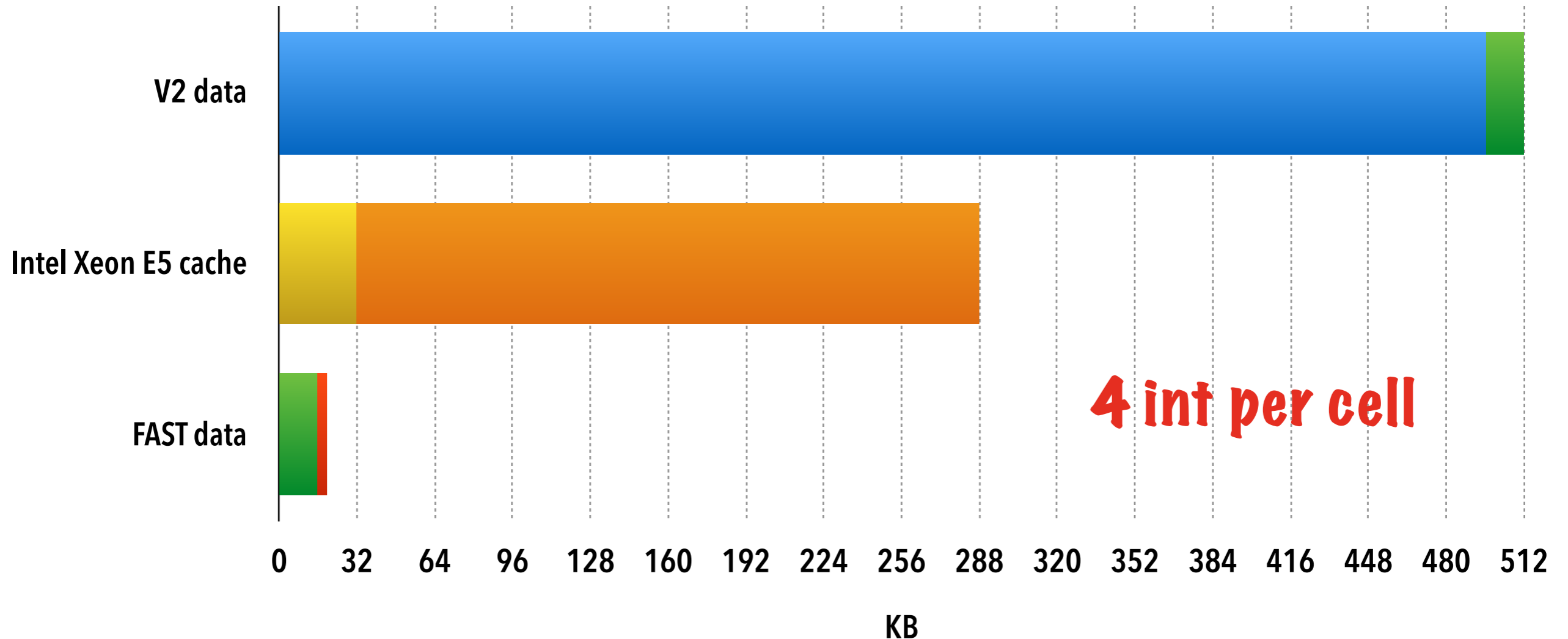
19/11/2014



The Algorithm

Data amount

Zero Real data L1 cache L2 cache Bitmask



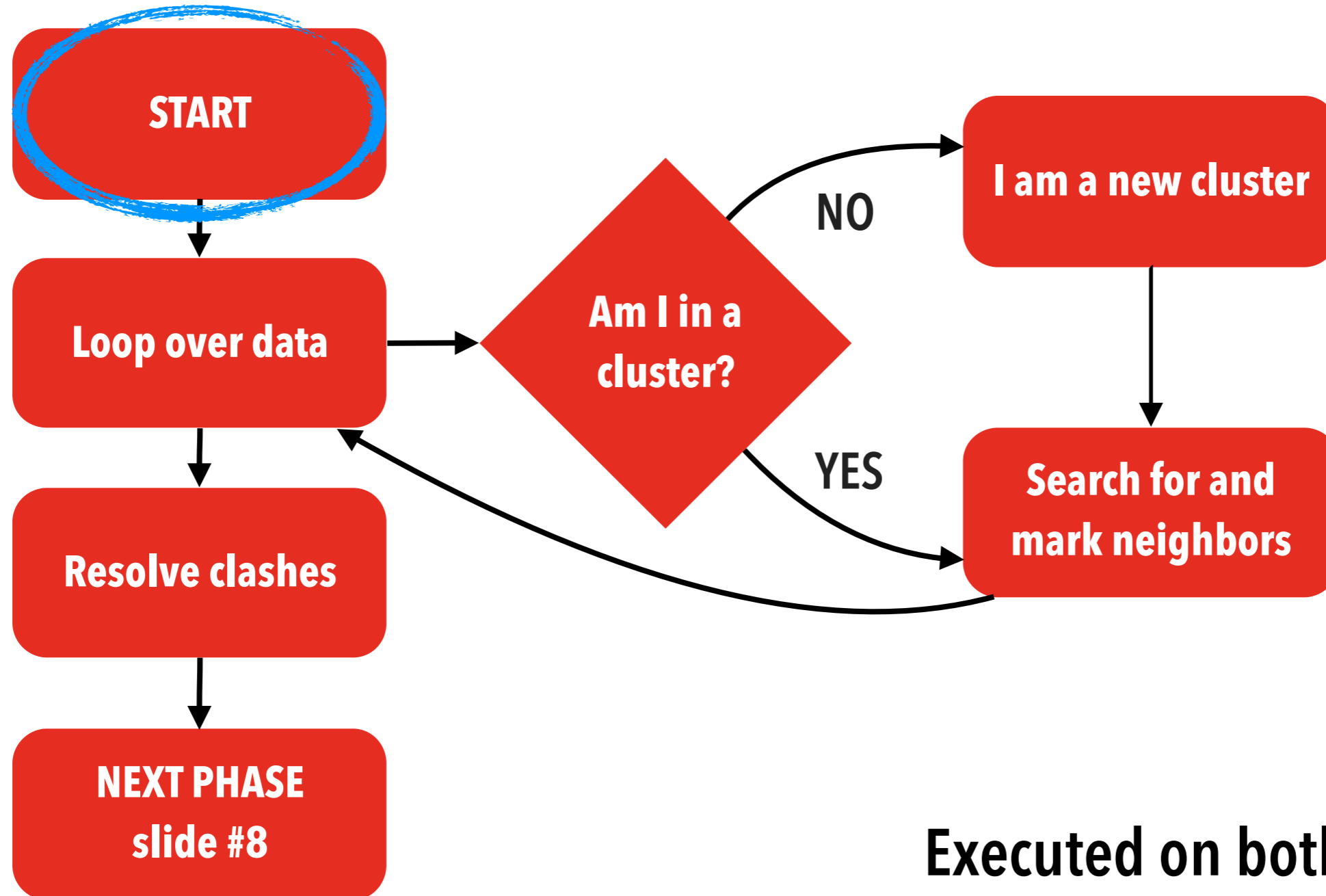
4 int per cell

○ Previous CF → v2

○ My (current) CF → fast

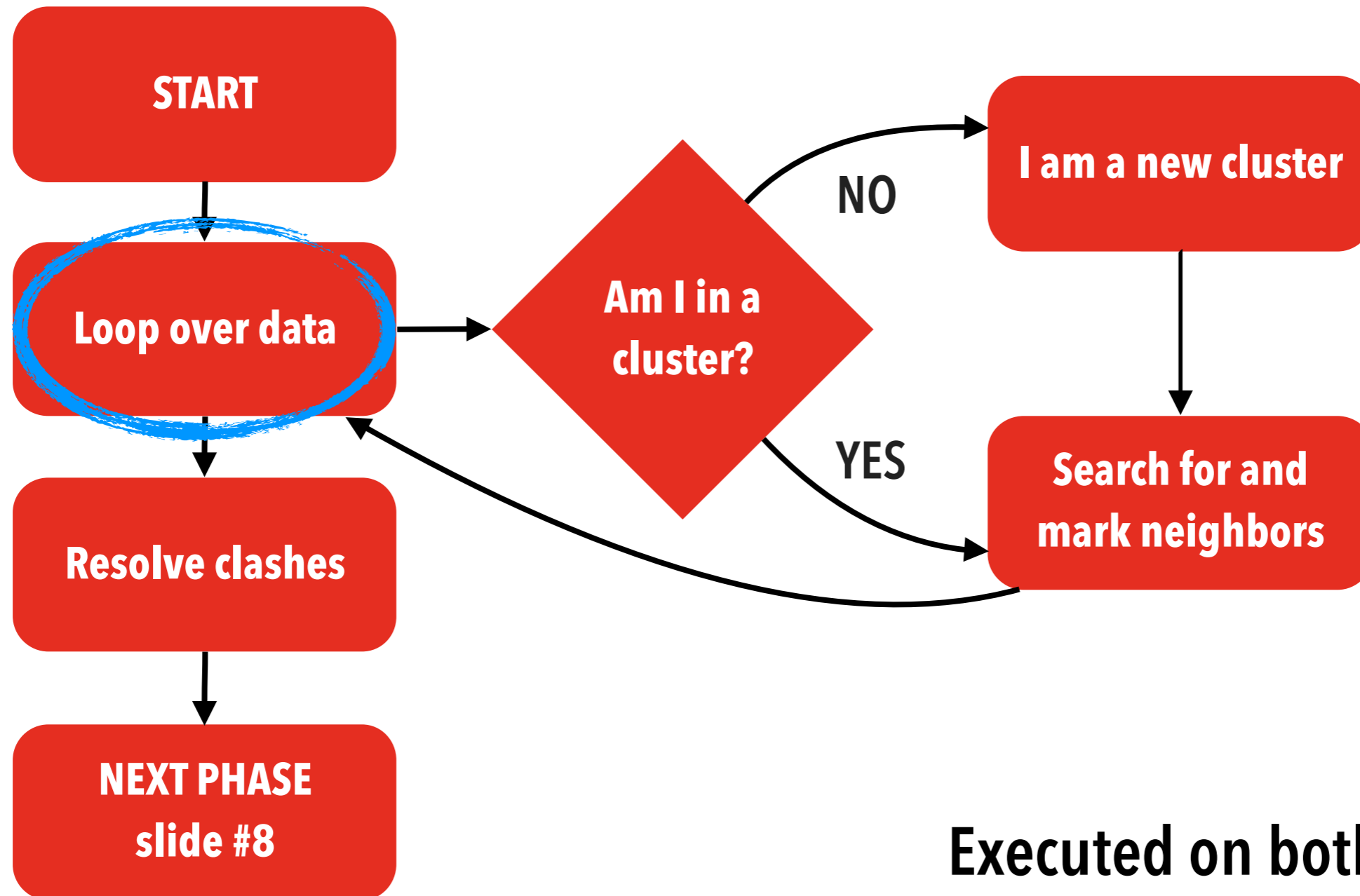
Data for each side (inner layer, max occupancy 2.5% central Pb-Pb event)

Fast algorithm



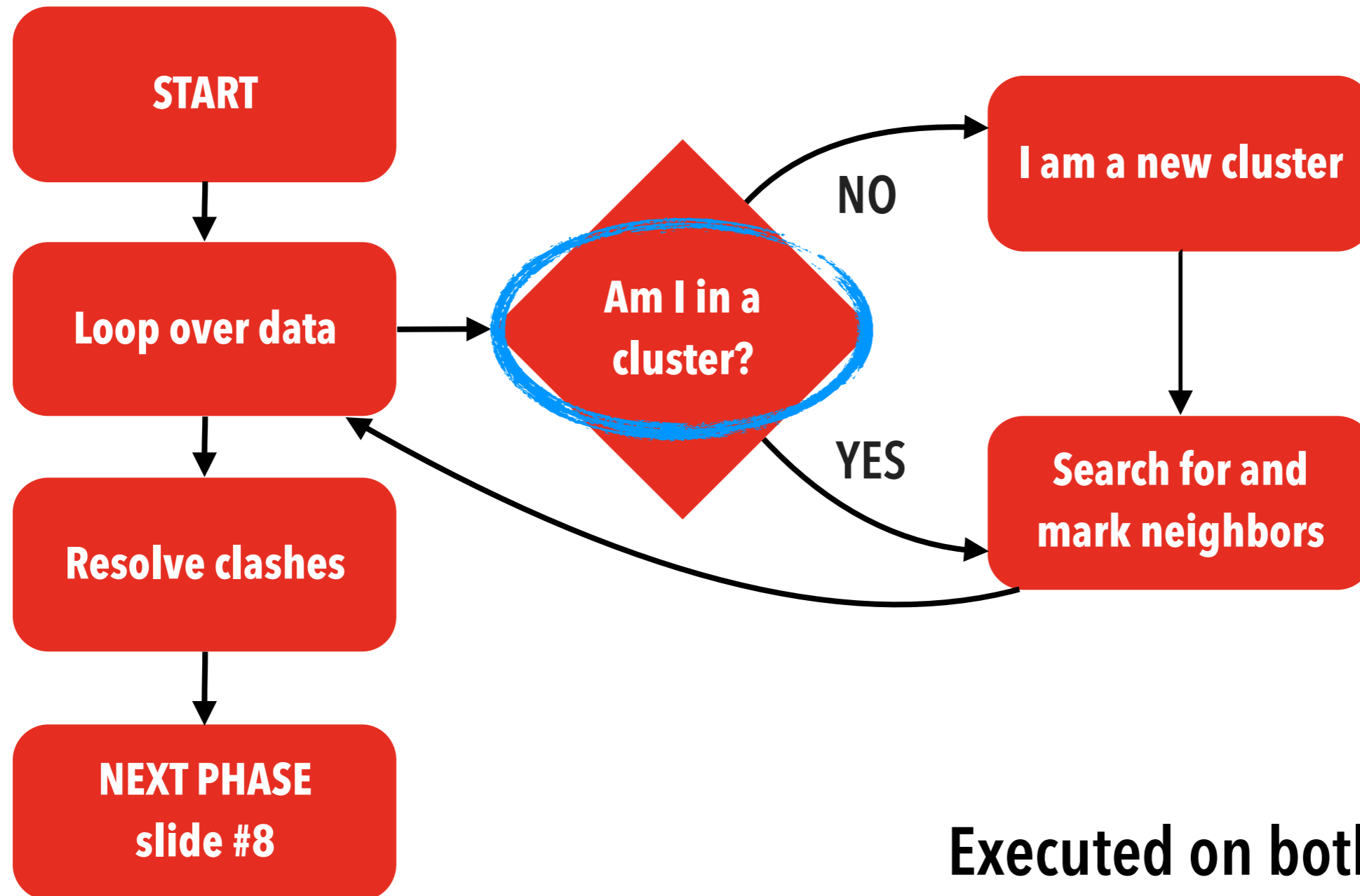
Executed on both sides of
each SDD module

Fast algorithm



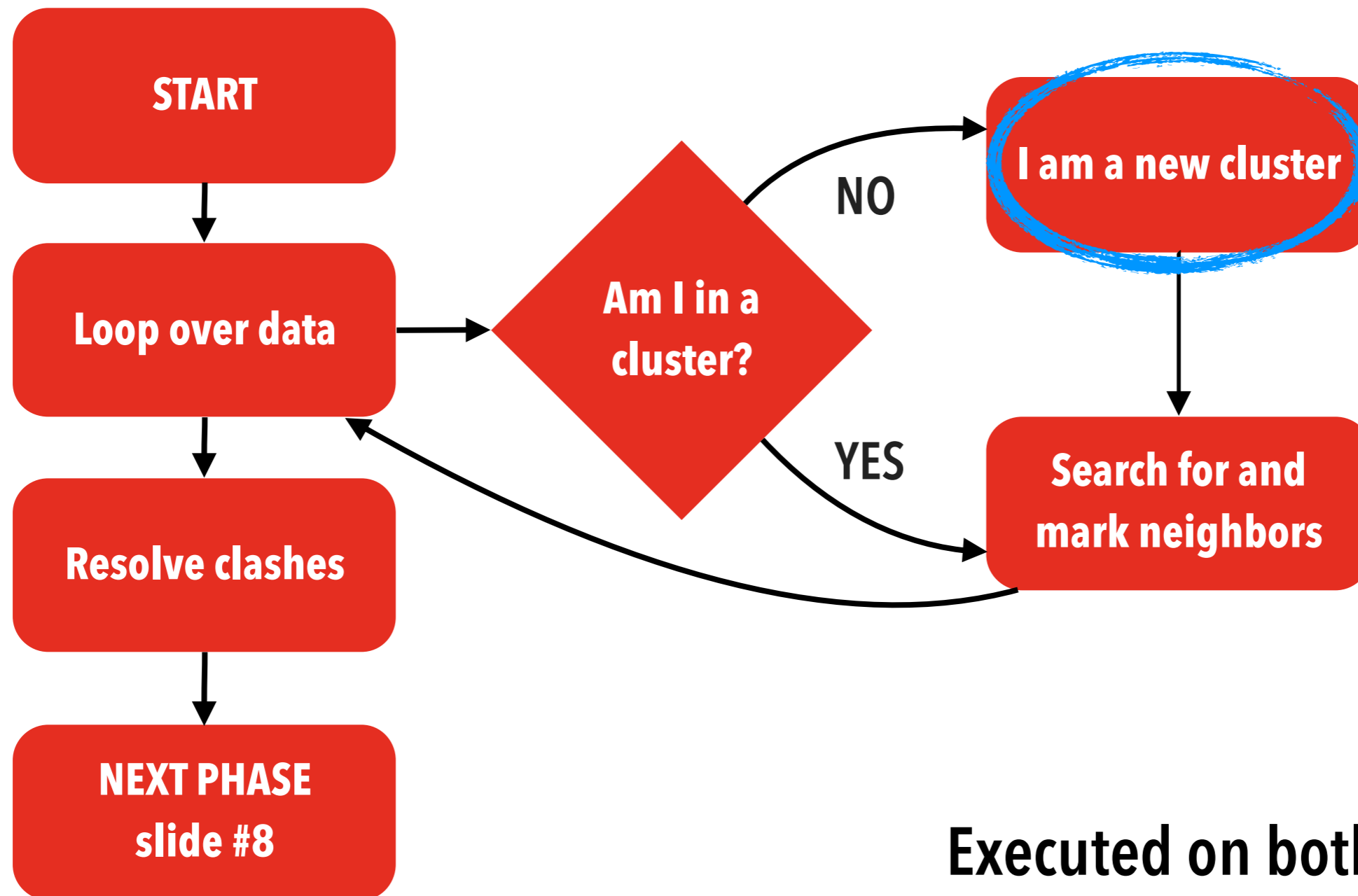
Executed on both sides of
each SDD module

Fast algorithm



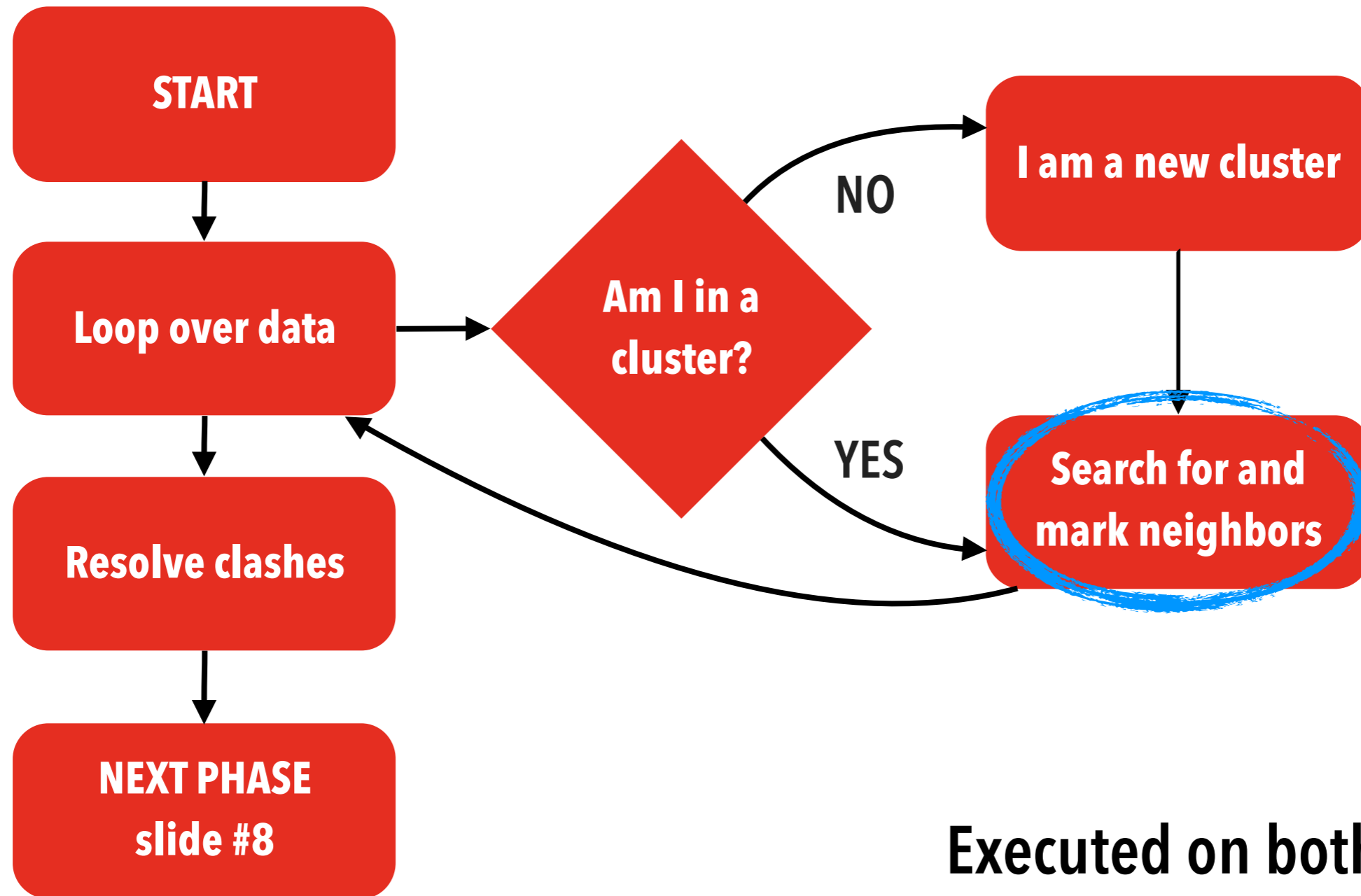
Executed on both sides of
each SDD module

Fast algorithm



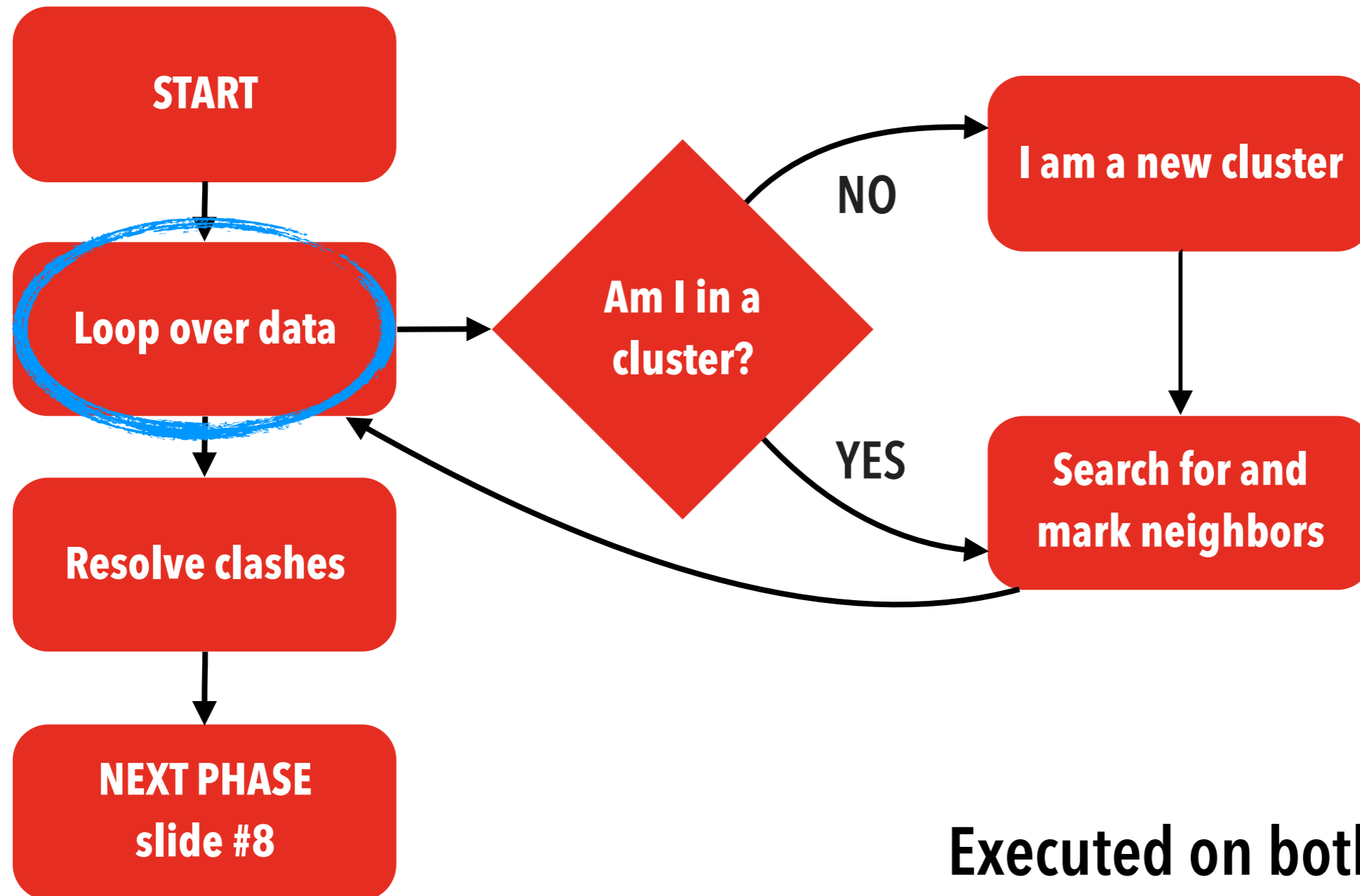
Executed on both sides of
each SDD module

Fast algorithm



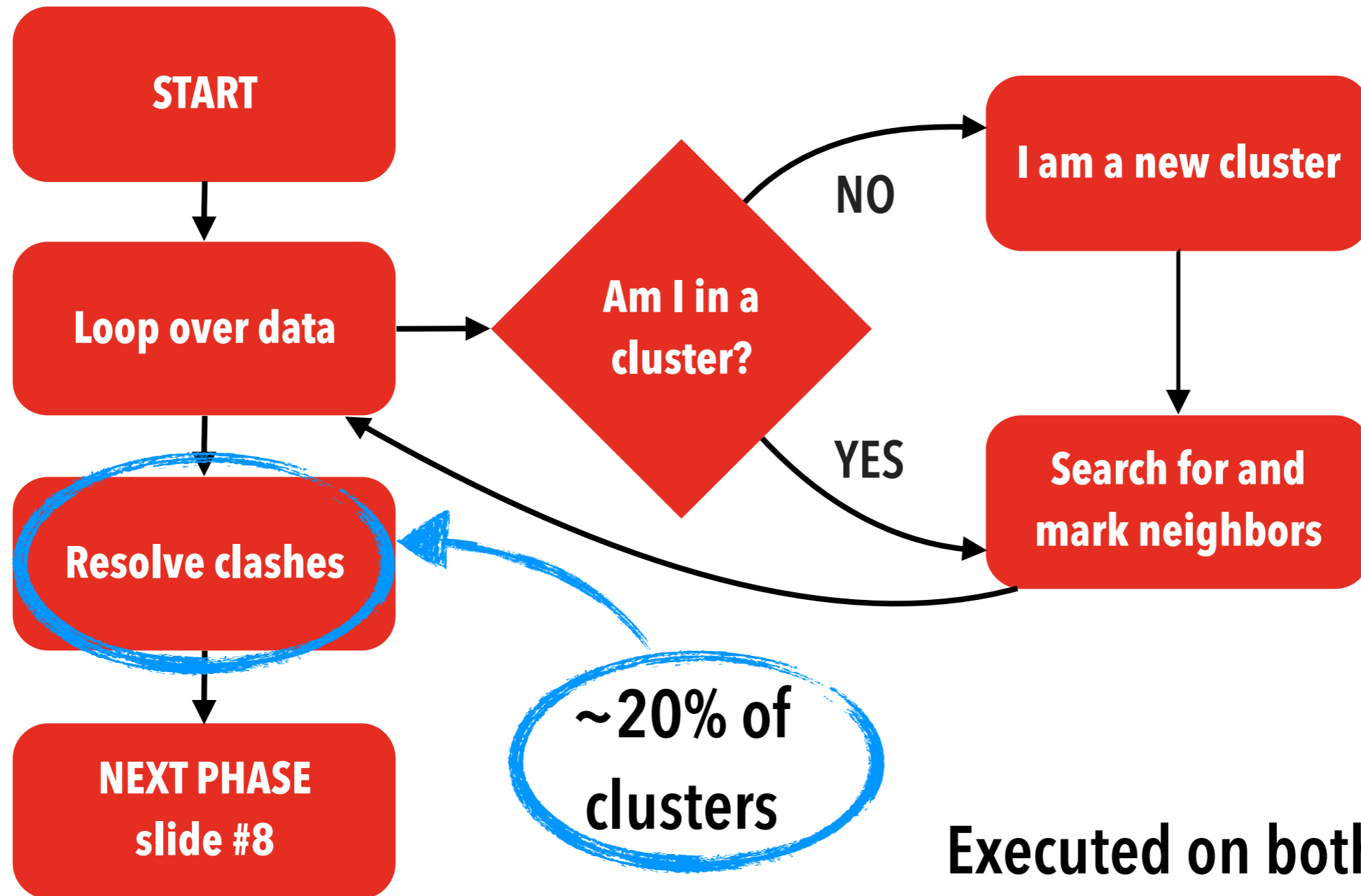
Executed on both sides of
each SDD module

Fast algorithm



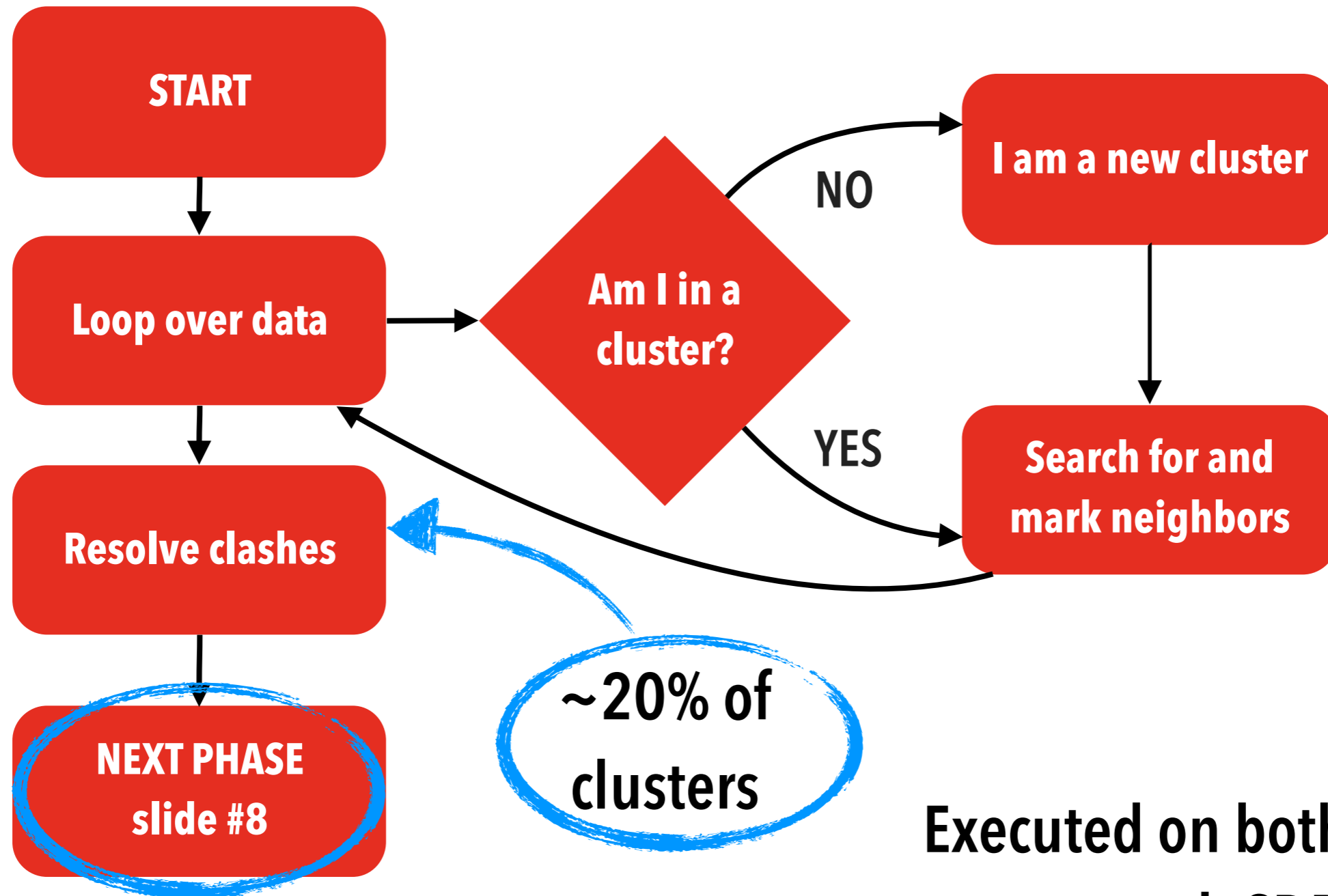
Executed on both sides of
each SDD module

Fast algorithm



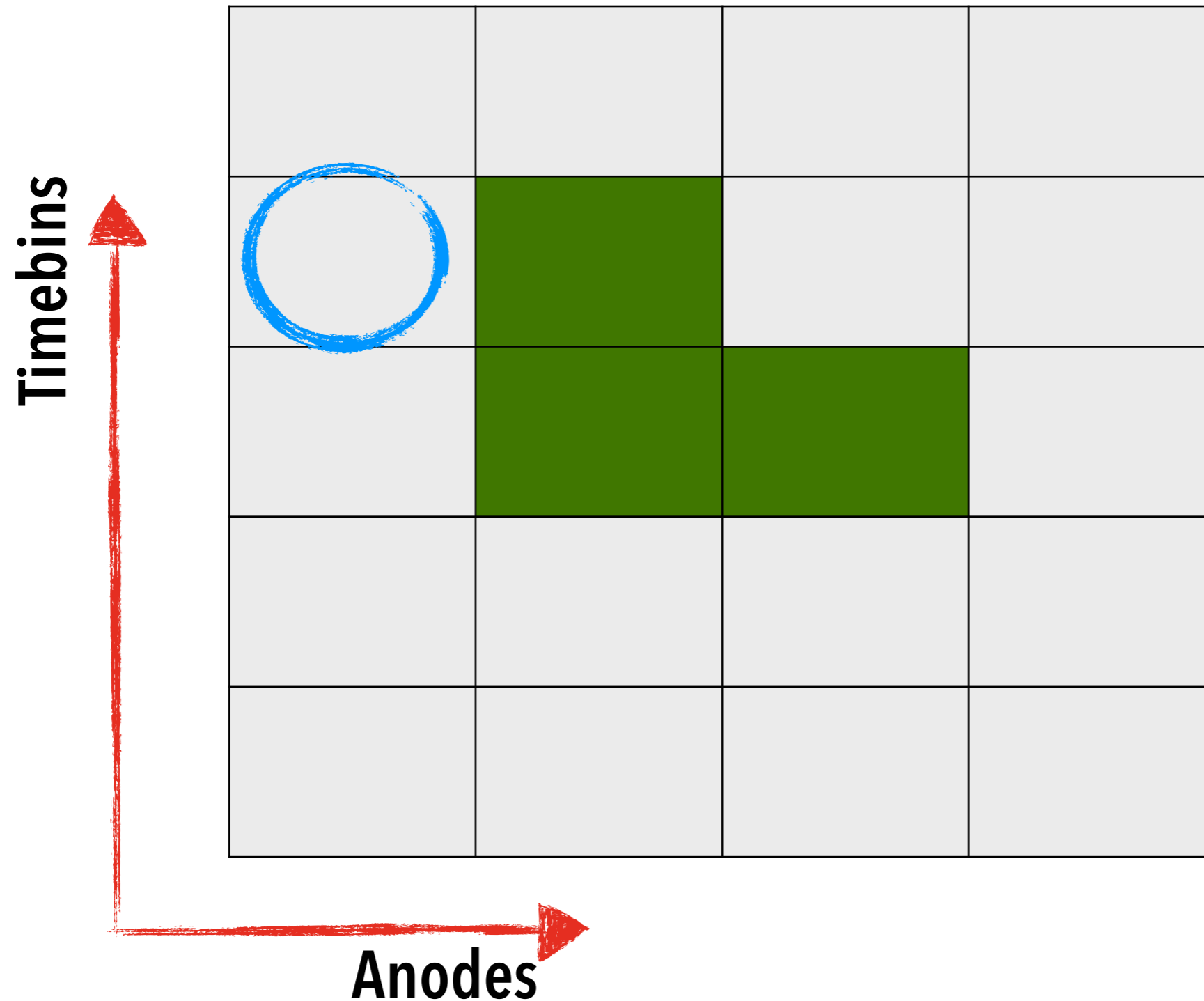
Executed on both sides of
each SDD module

Fast algorithm

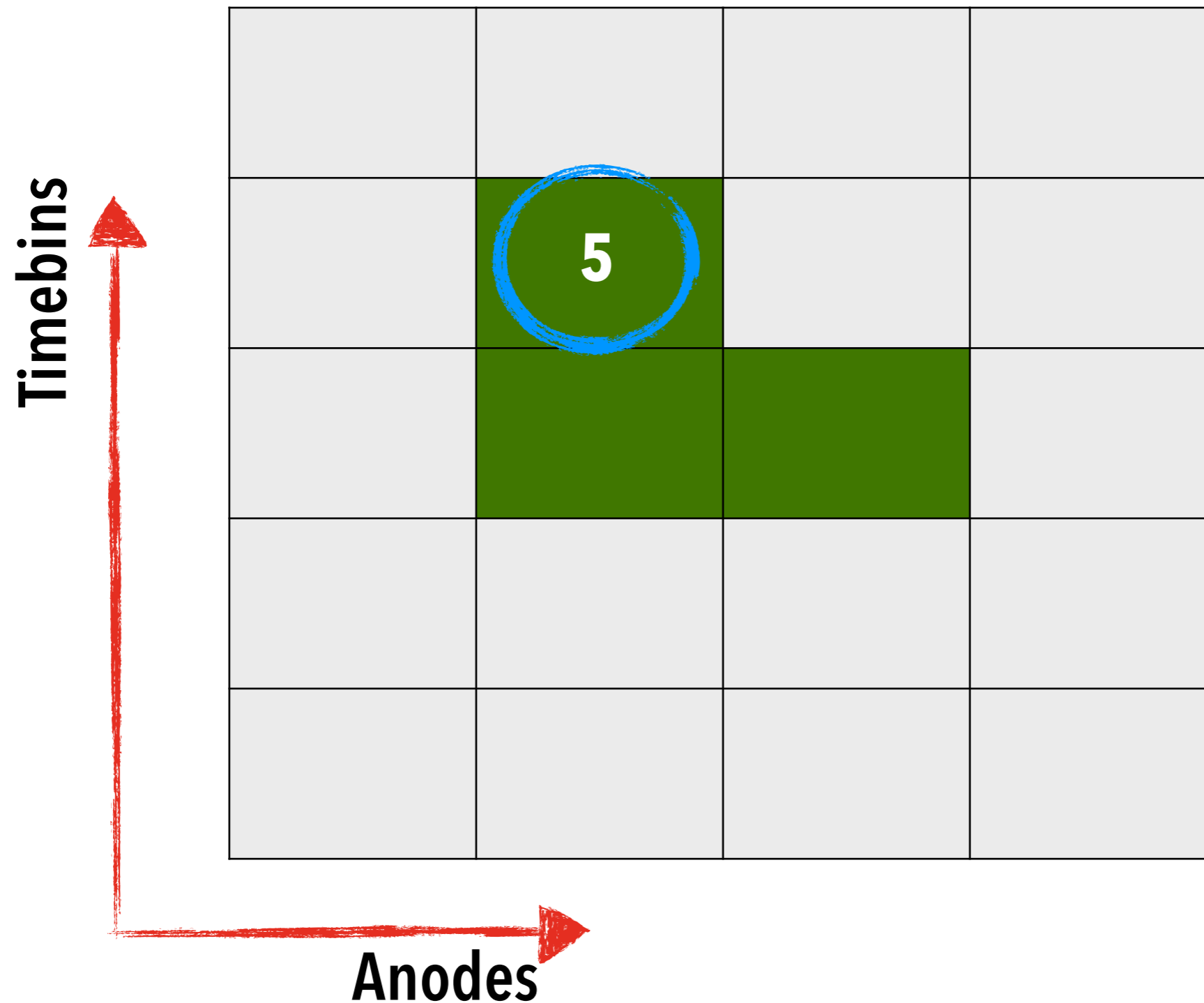


Executed on both sides of each SDD module

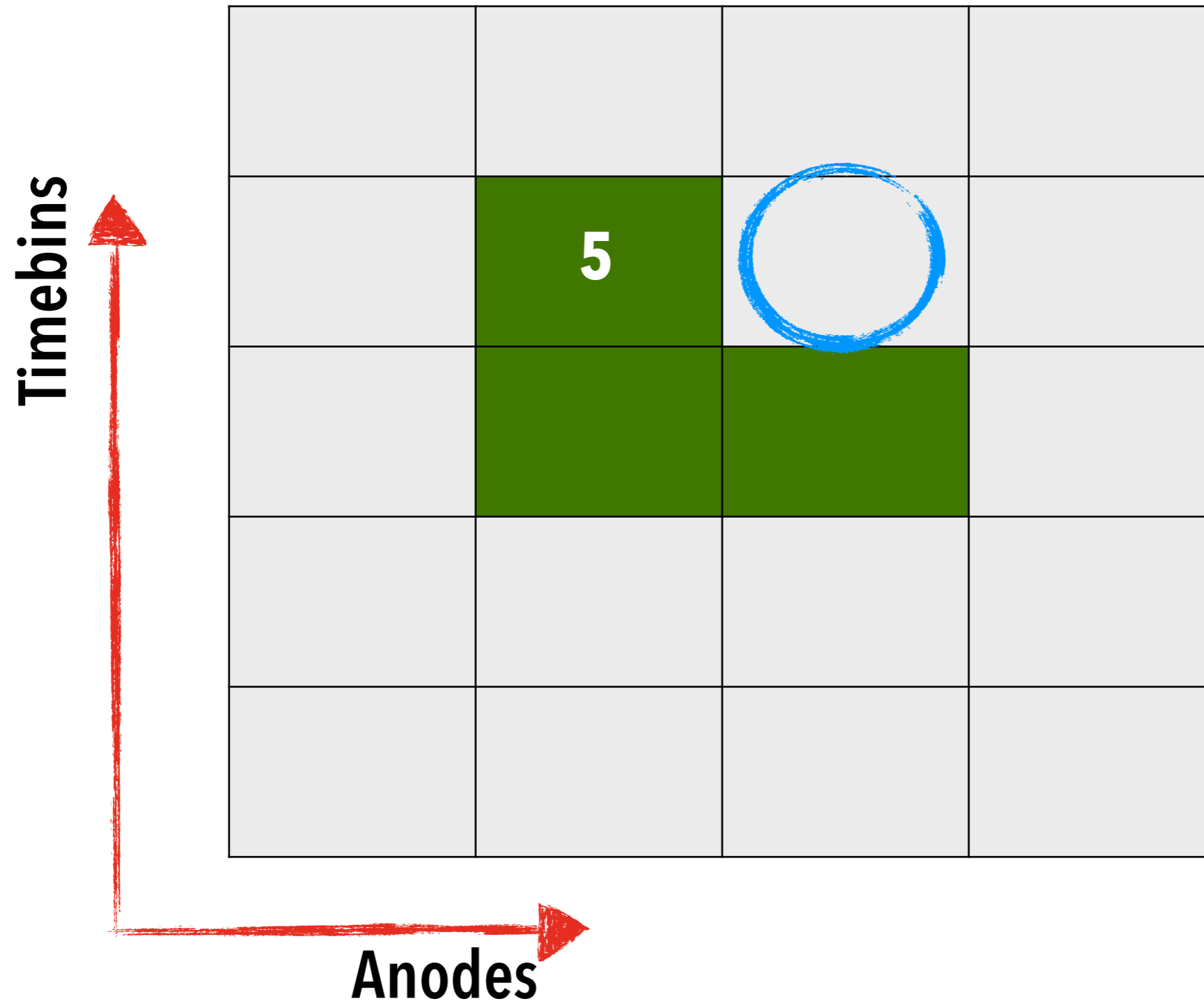
Fast algorithm



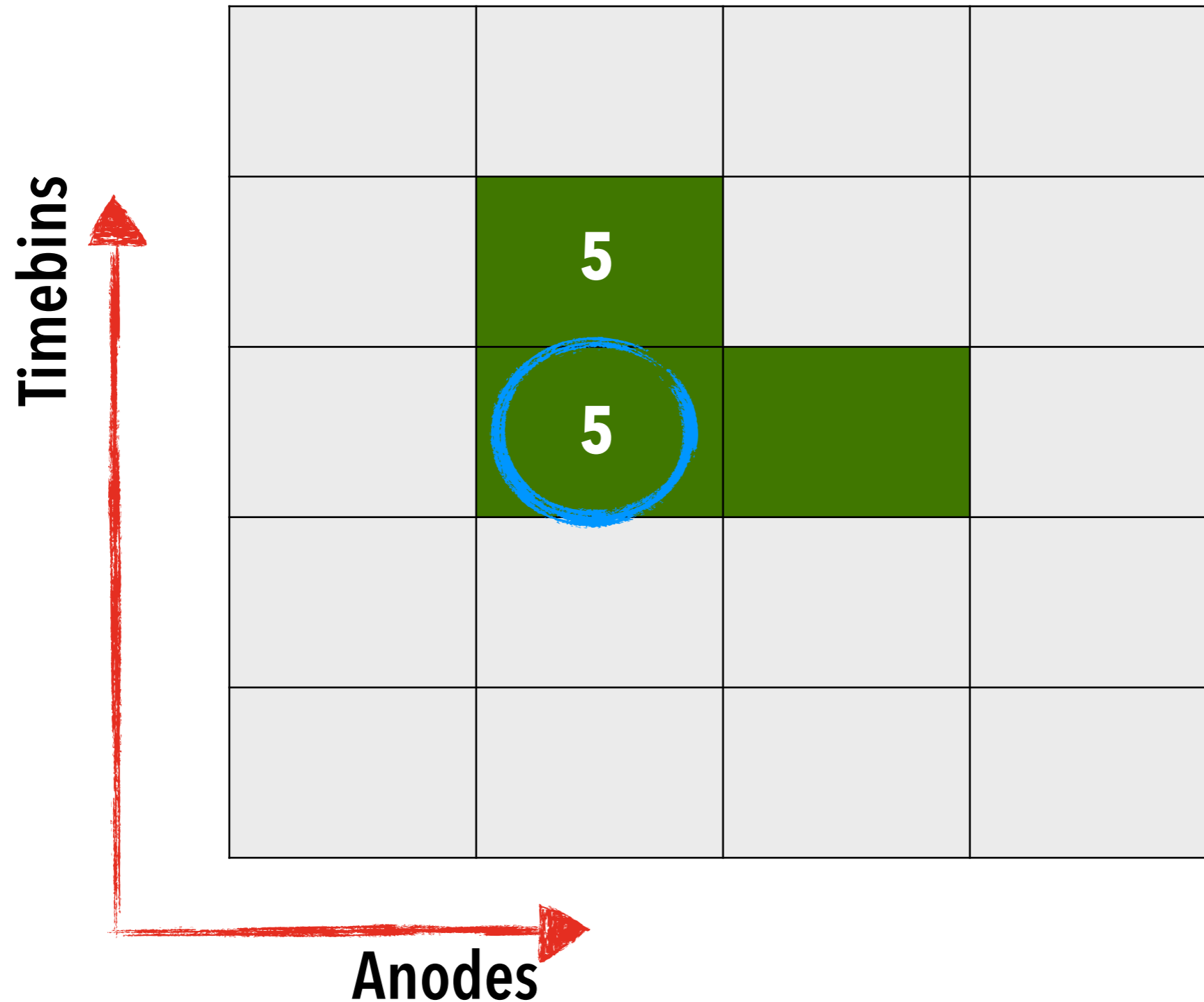
Fast algorithm



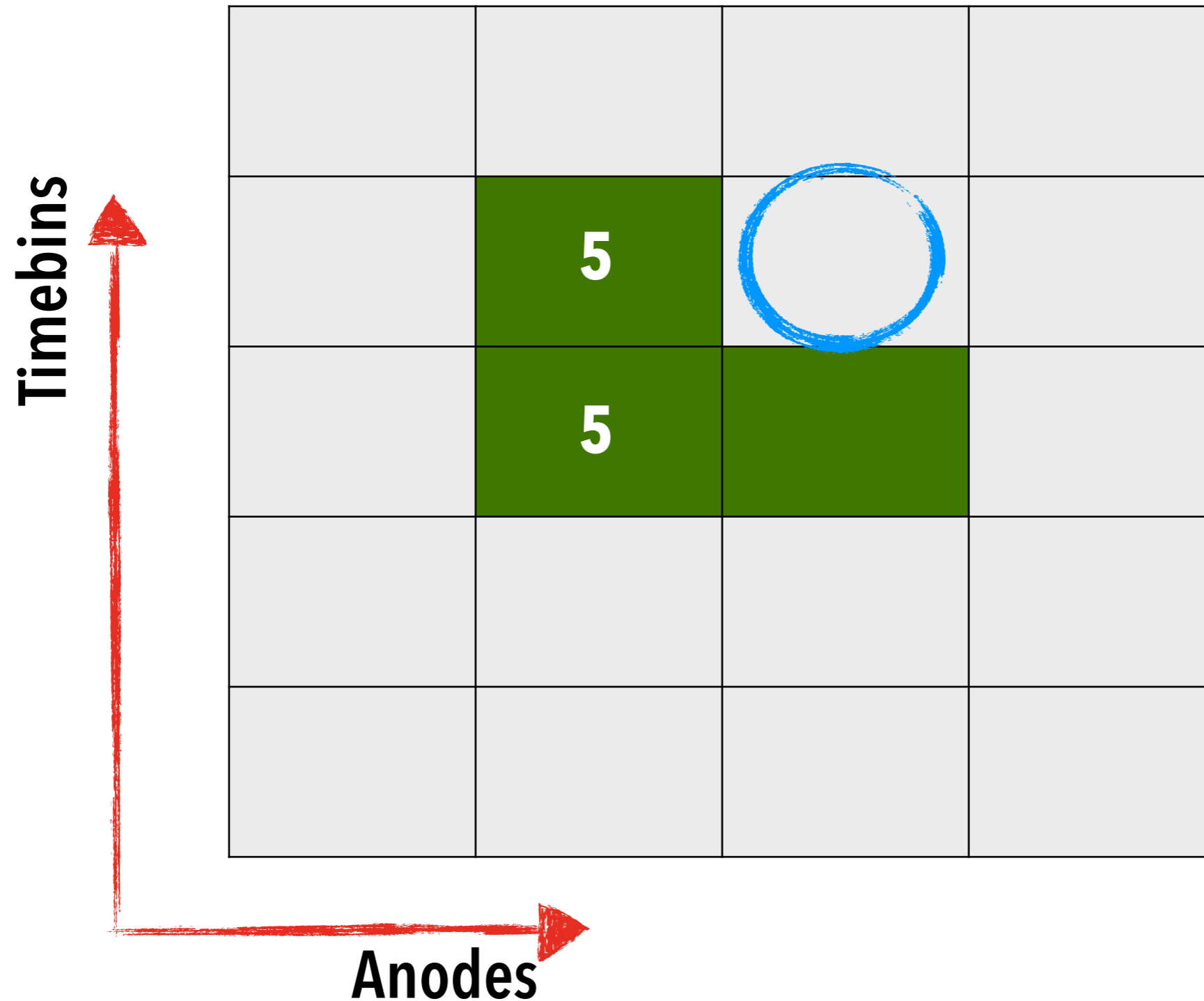
Fast algorithm



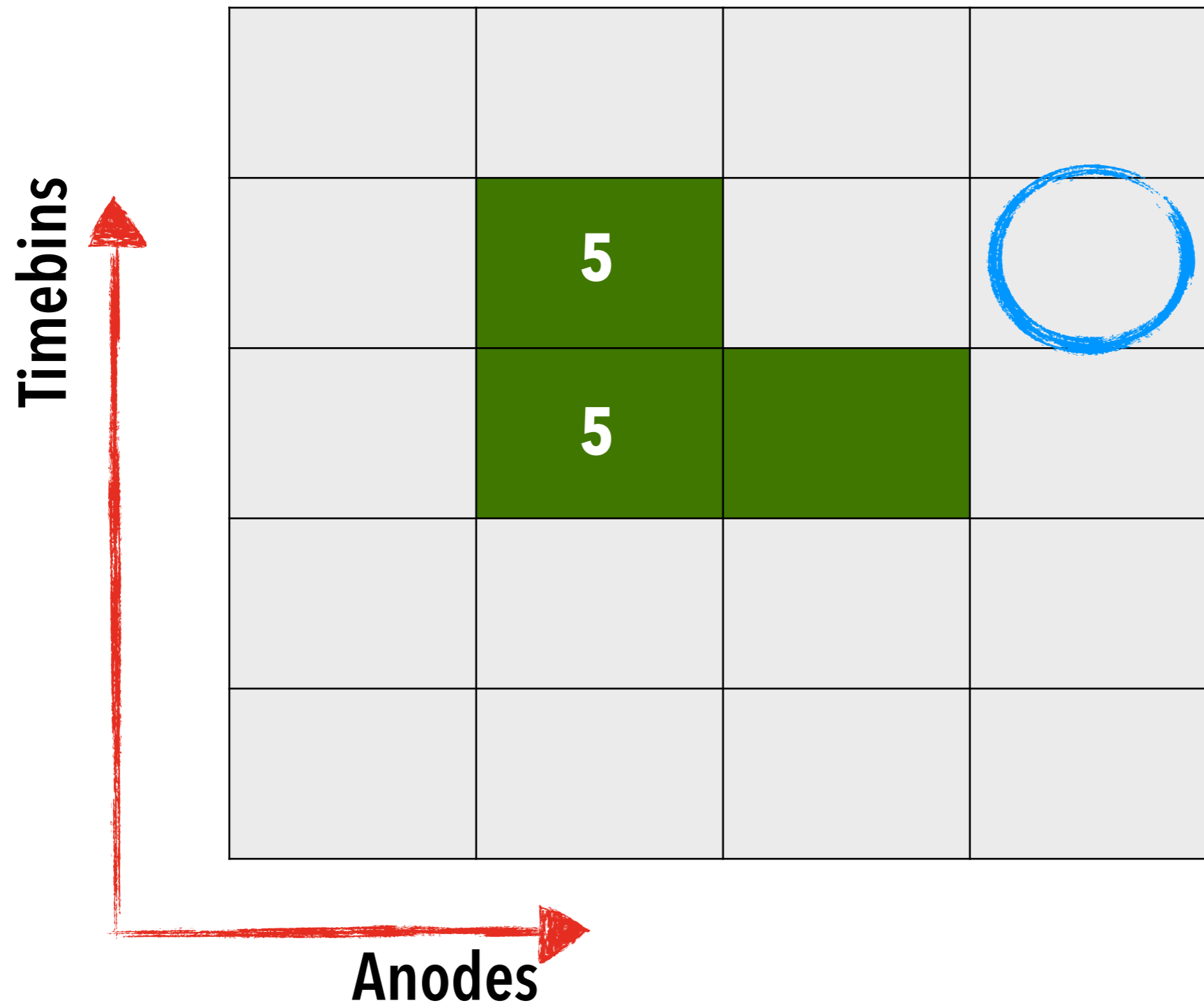
Fast algorithm



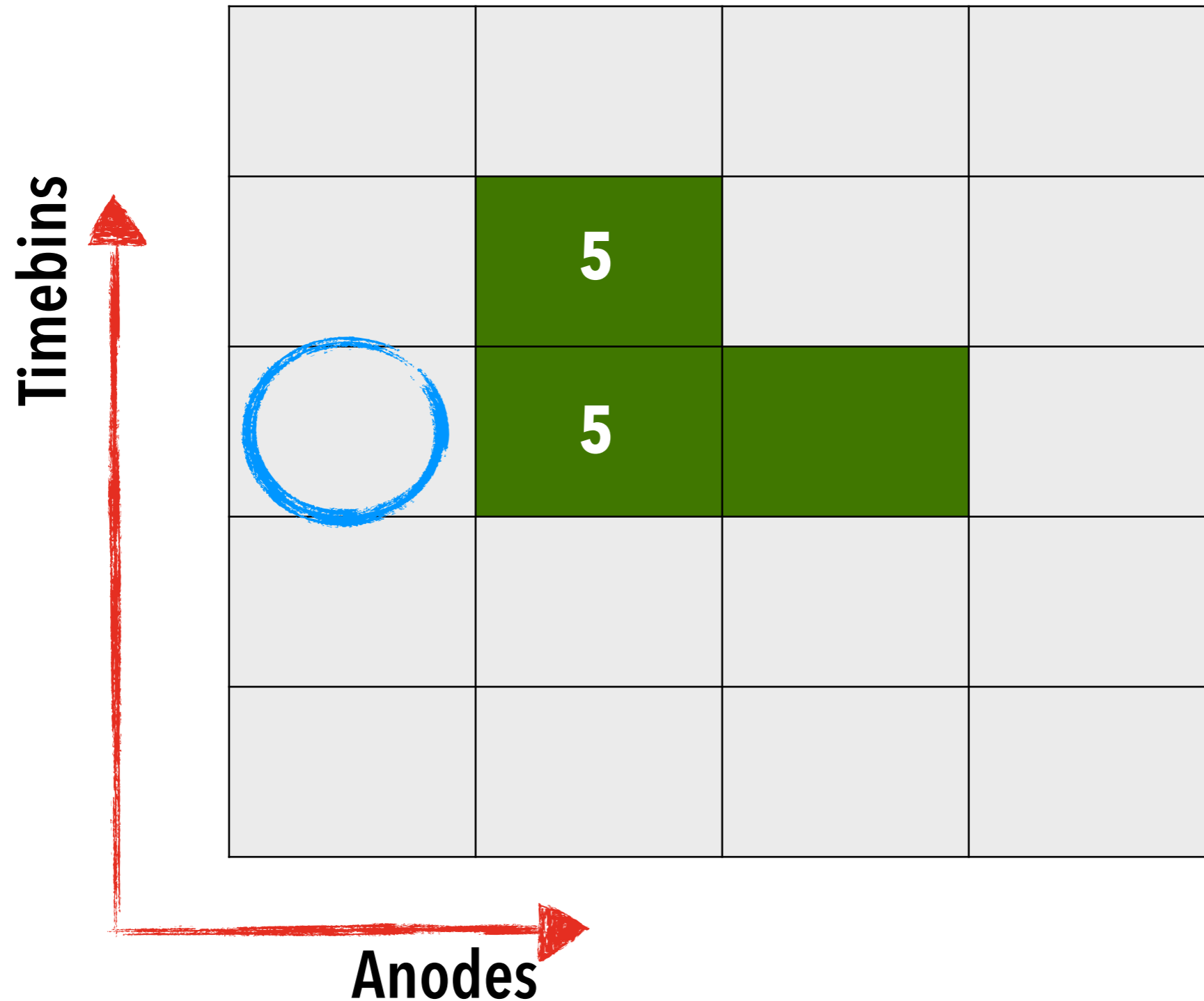
Fast algorithm



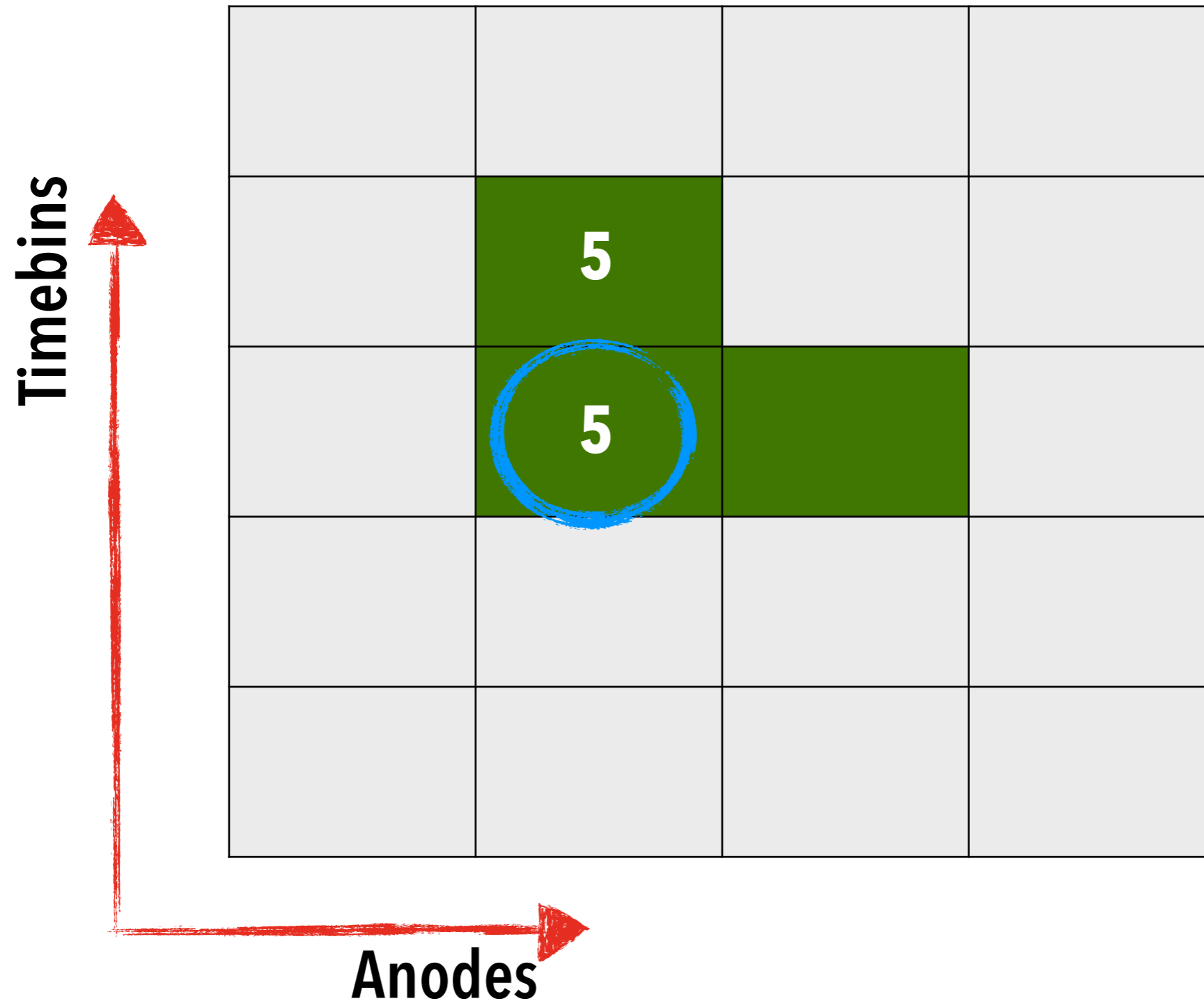
Fast algorithm



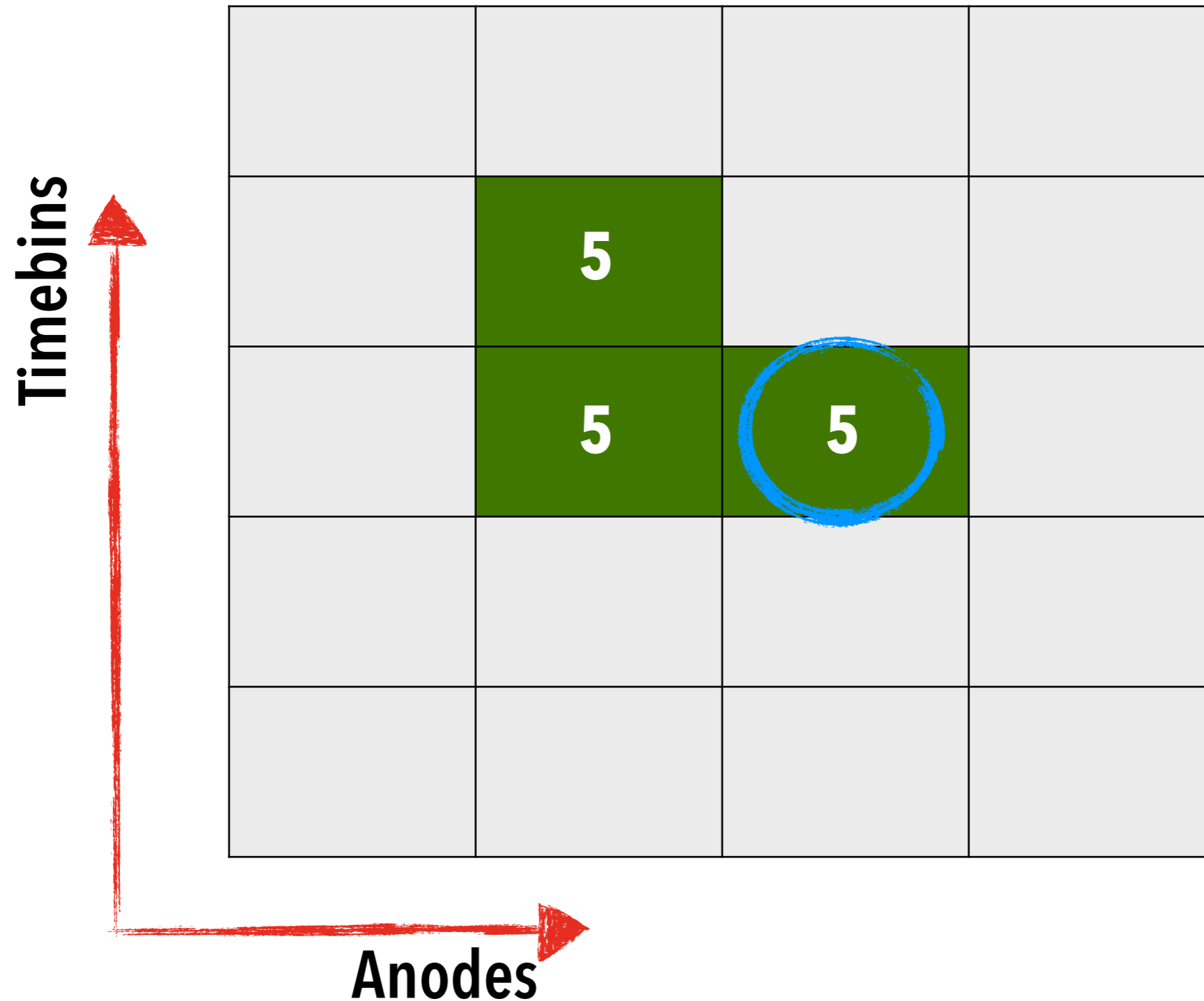
Fast algorithm



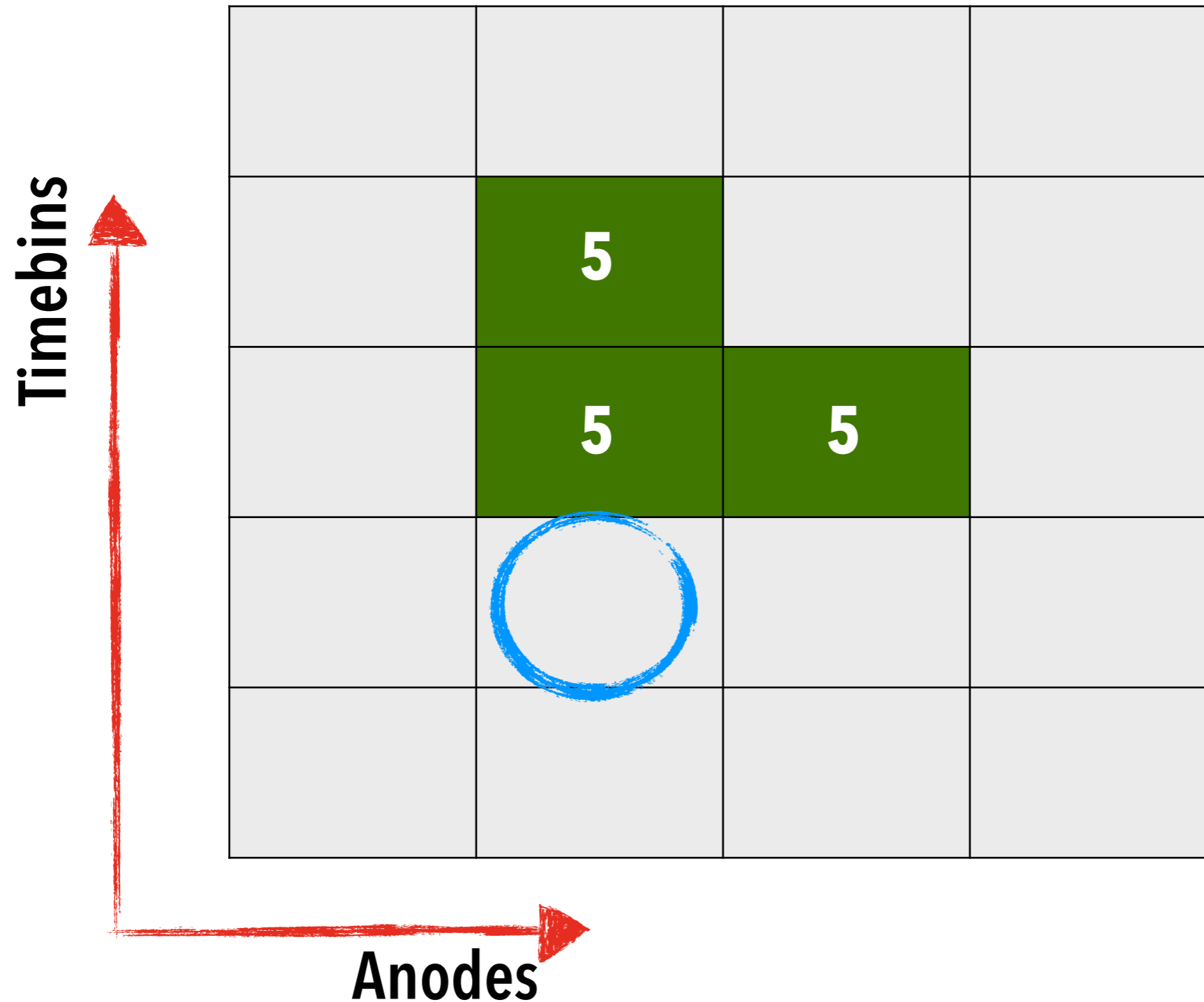
Fast algorithm



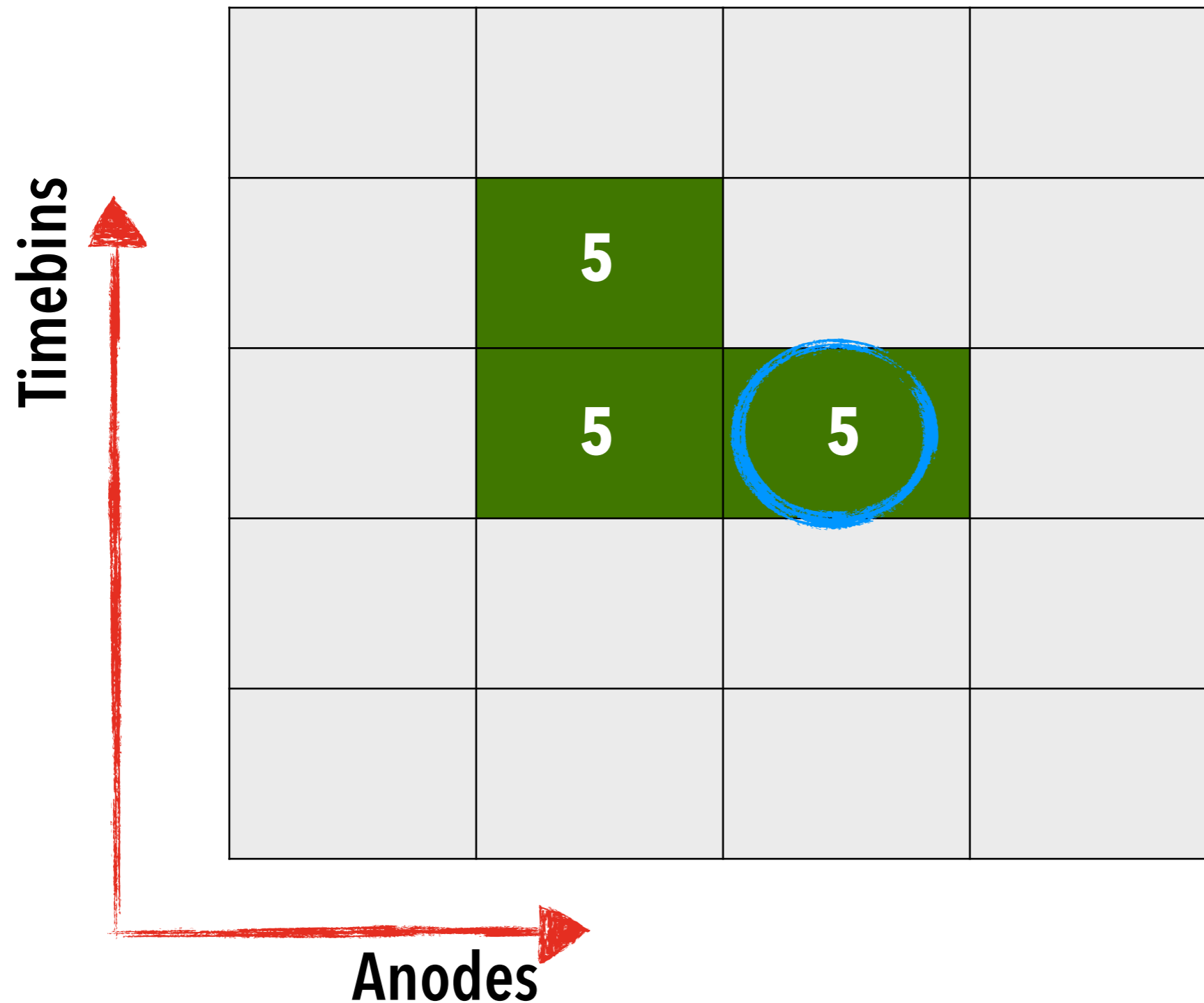
Fast algorithm



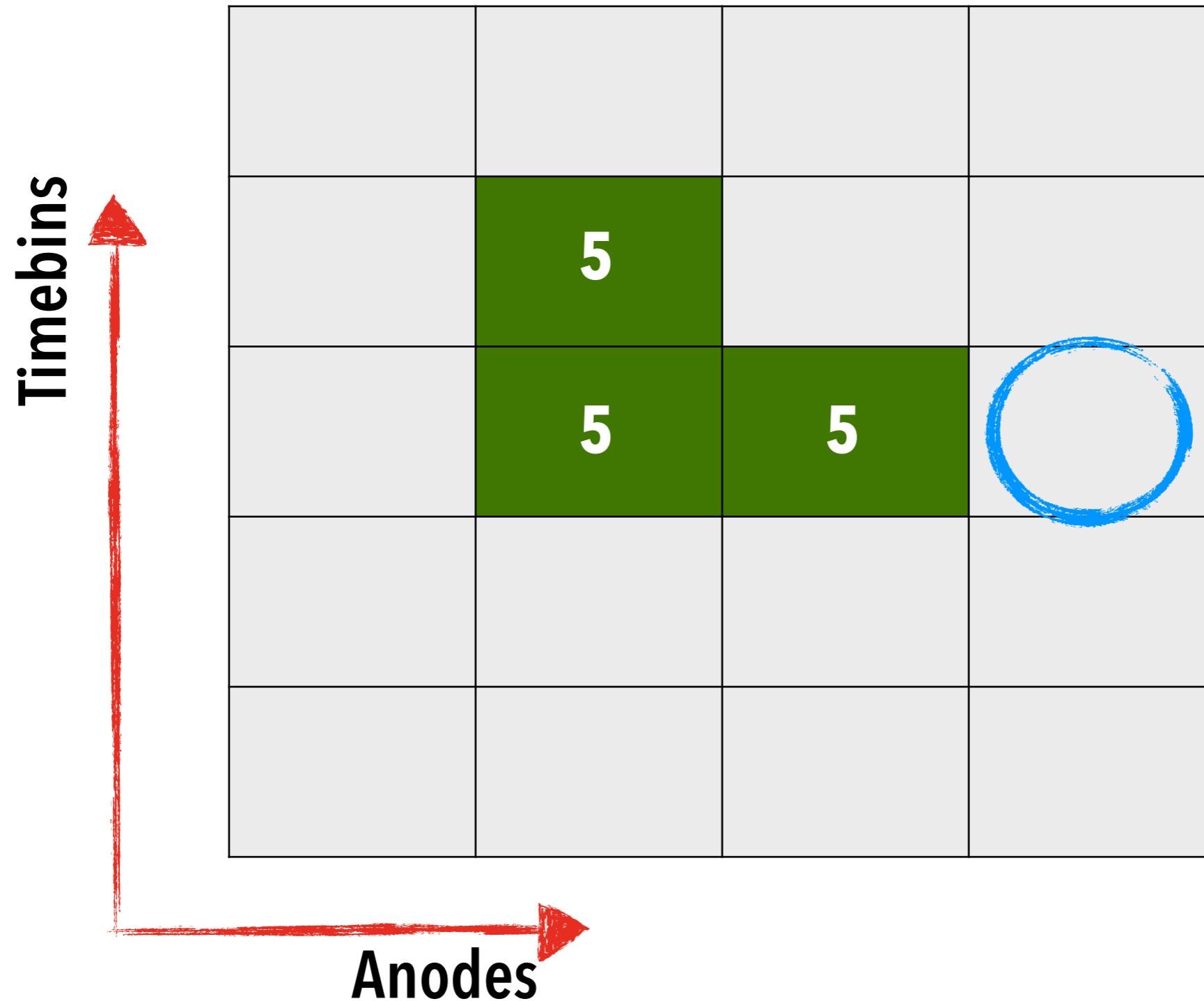
Fast algorithm



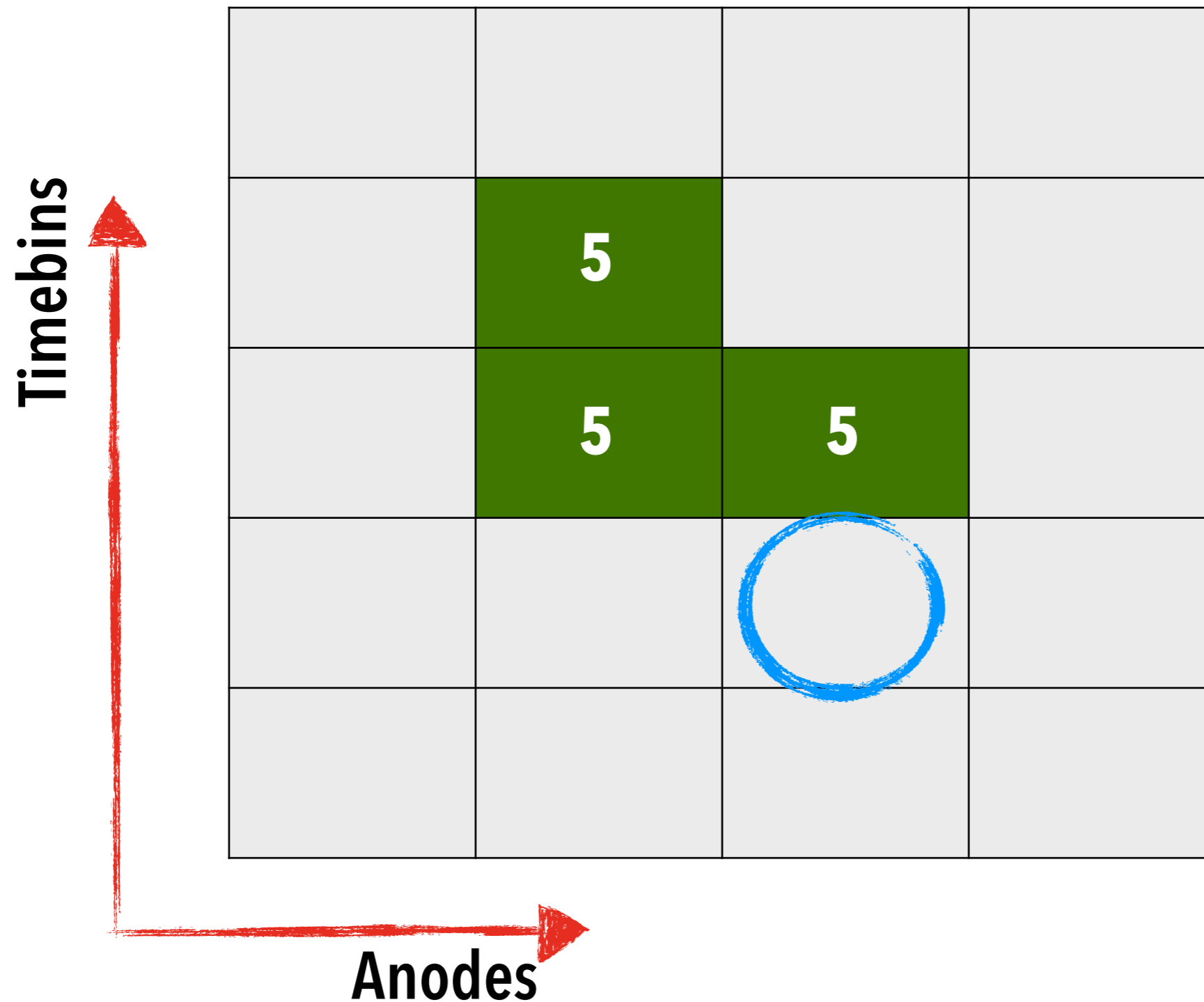
Fast algorithm



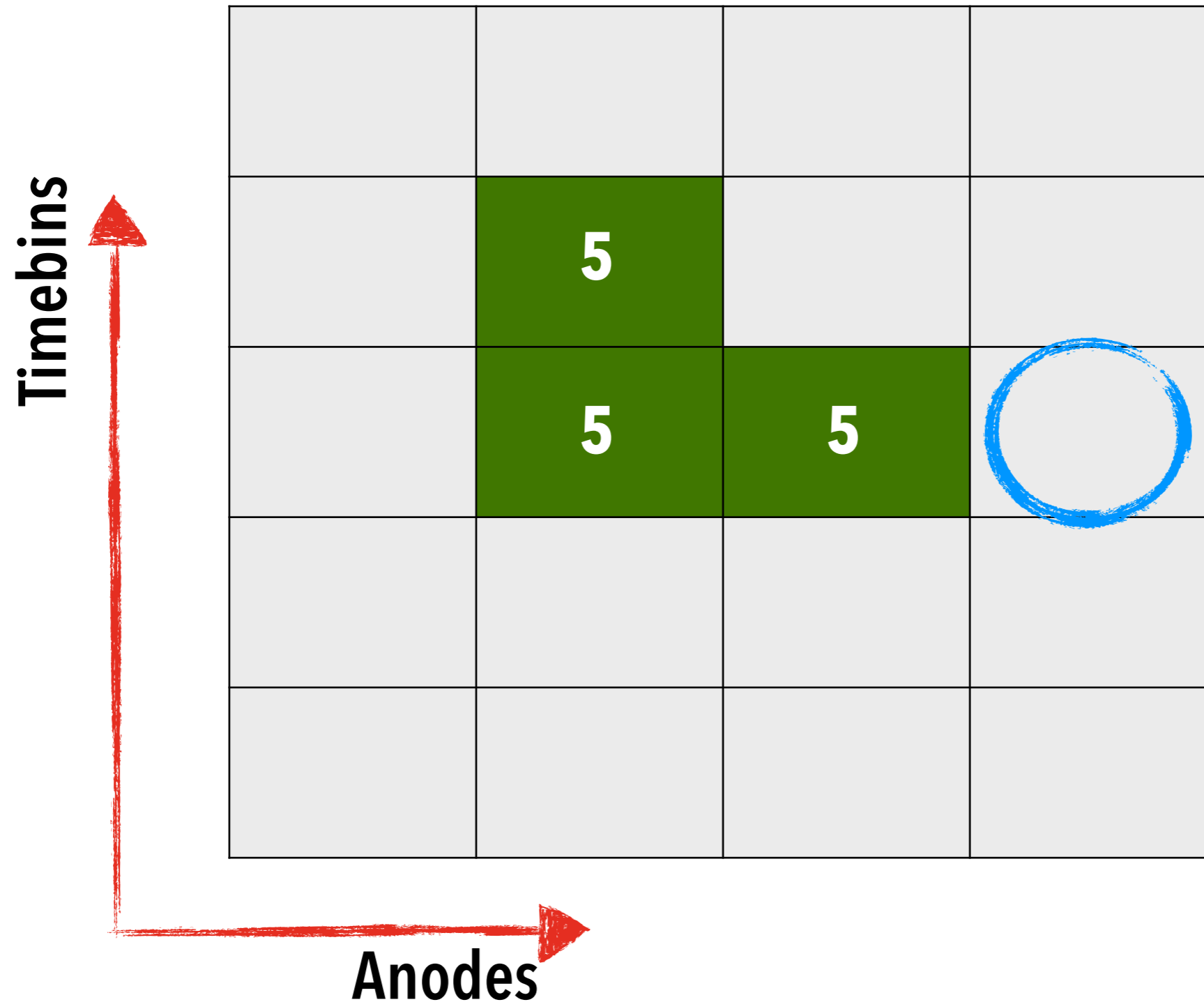
Fast algorithm



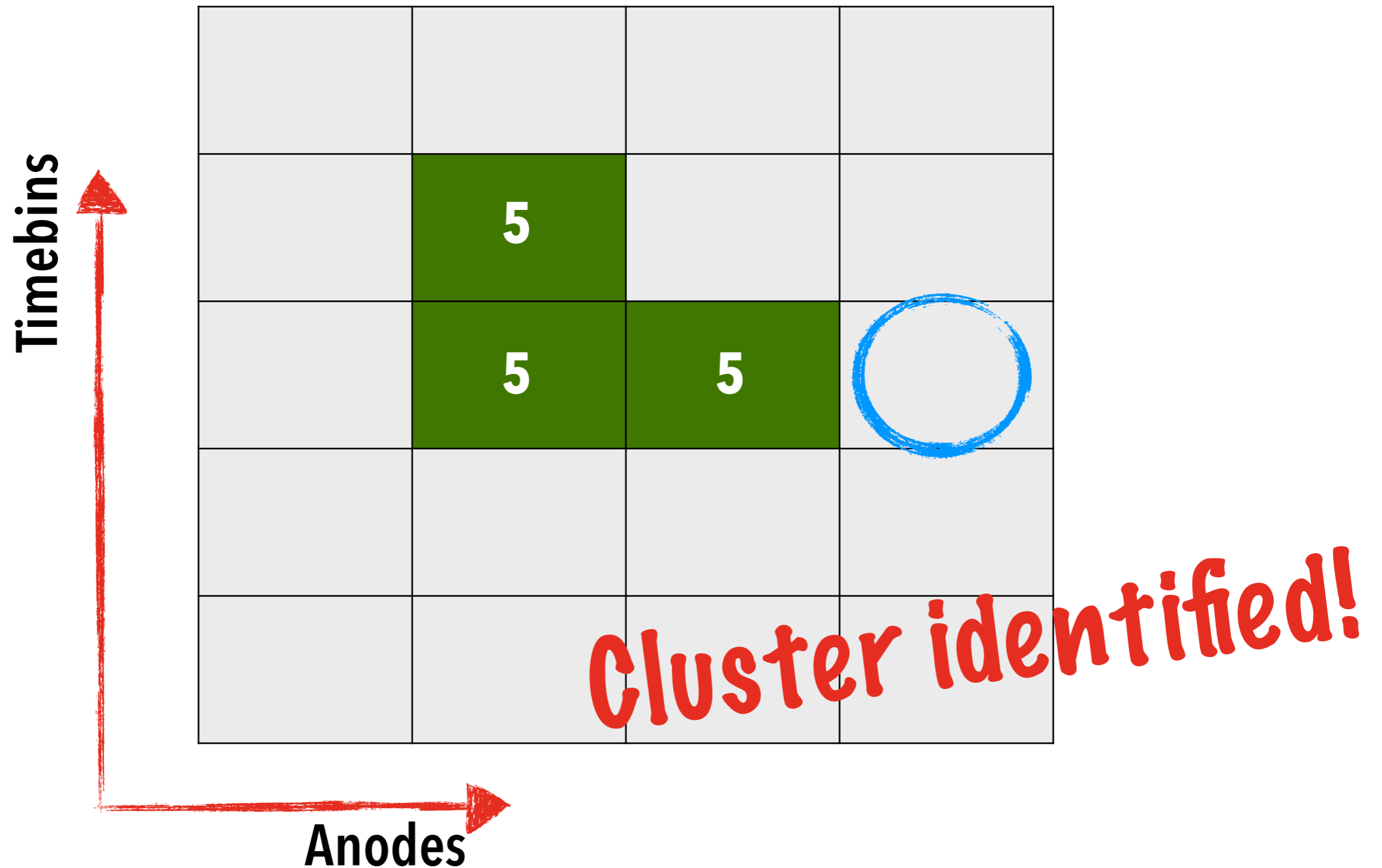
Fast algorithm



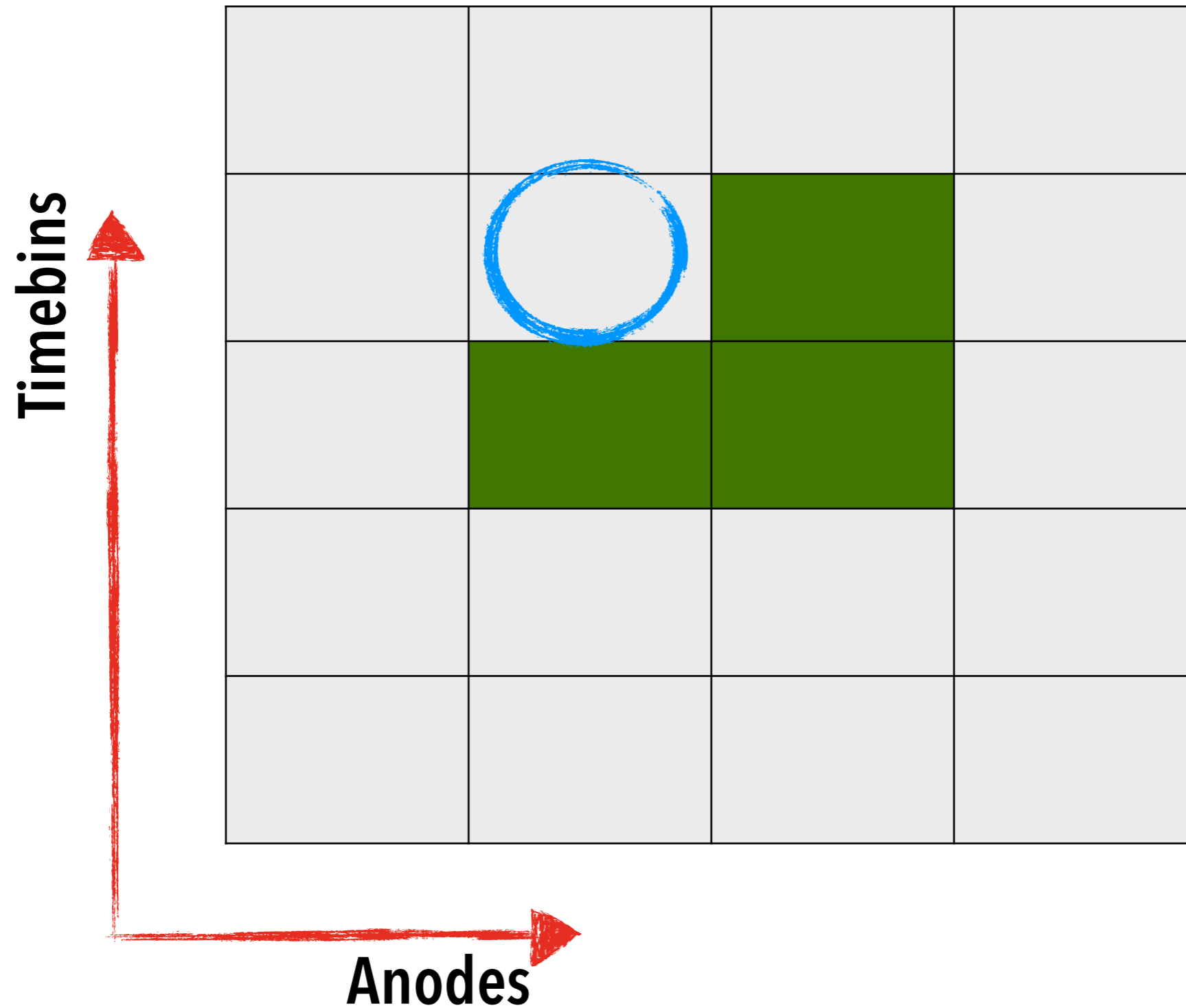
Fast algorithm



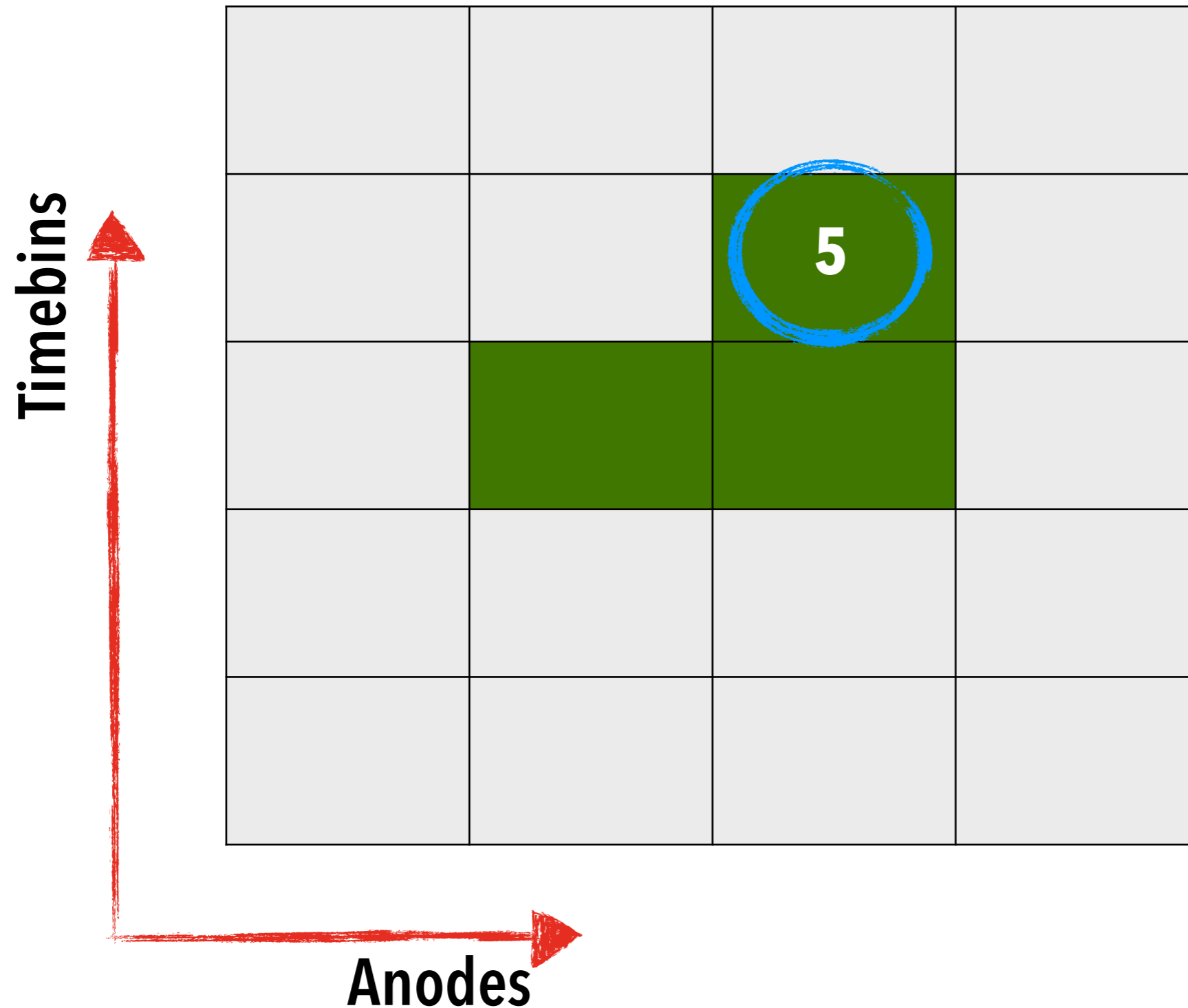
Fast algorithm



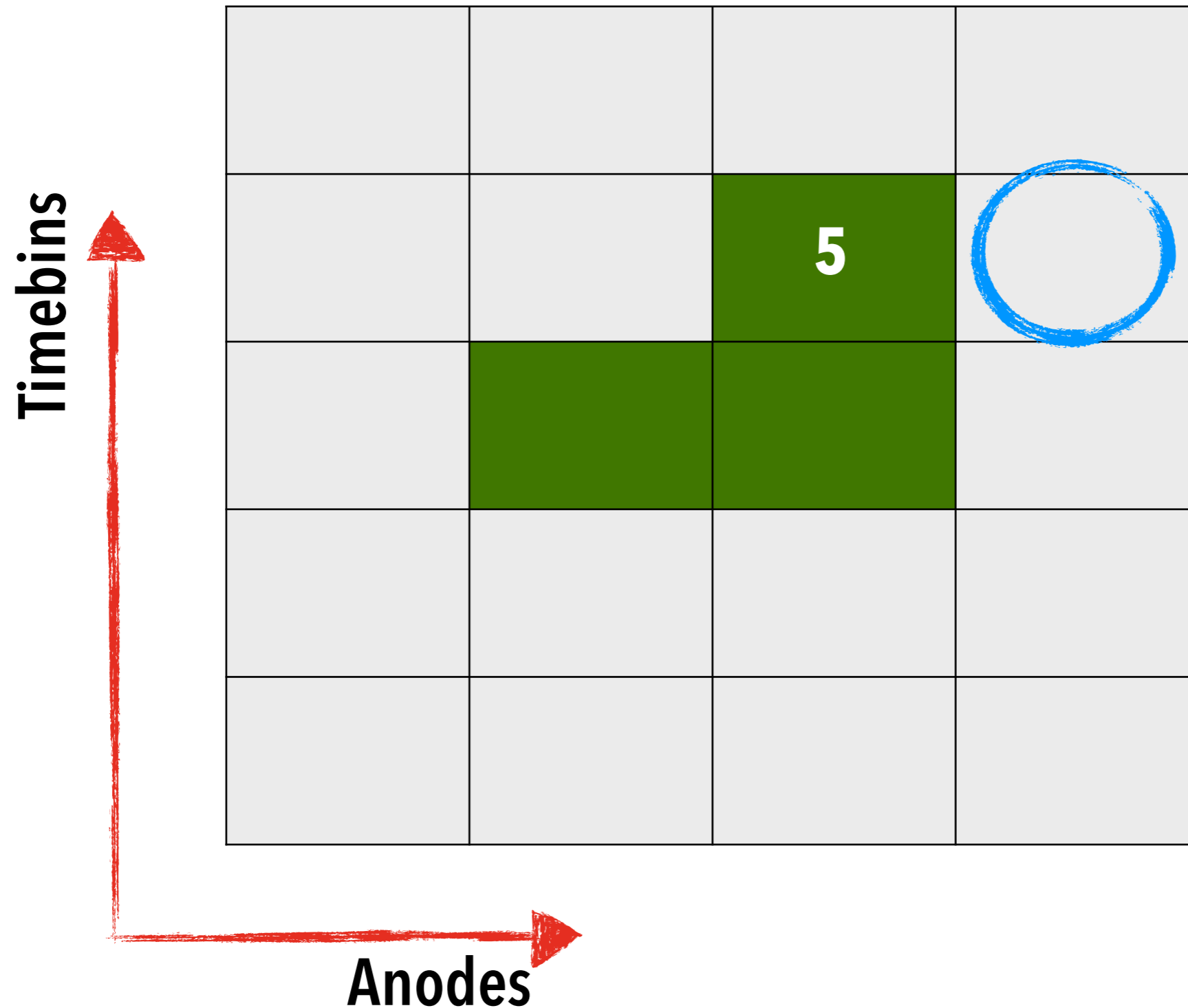
Fast algorithm



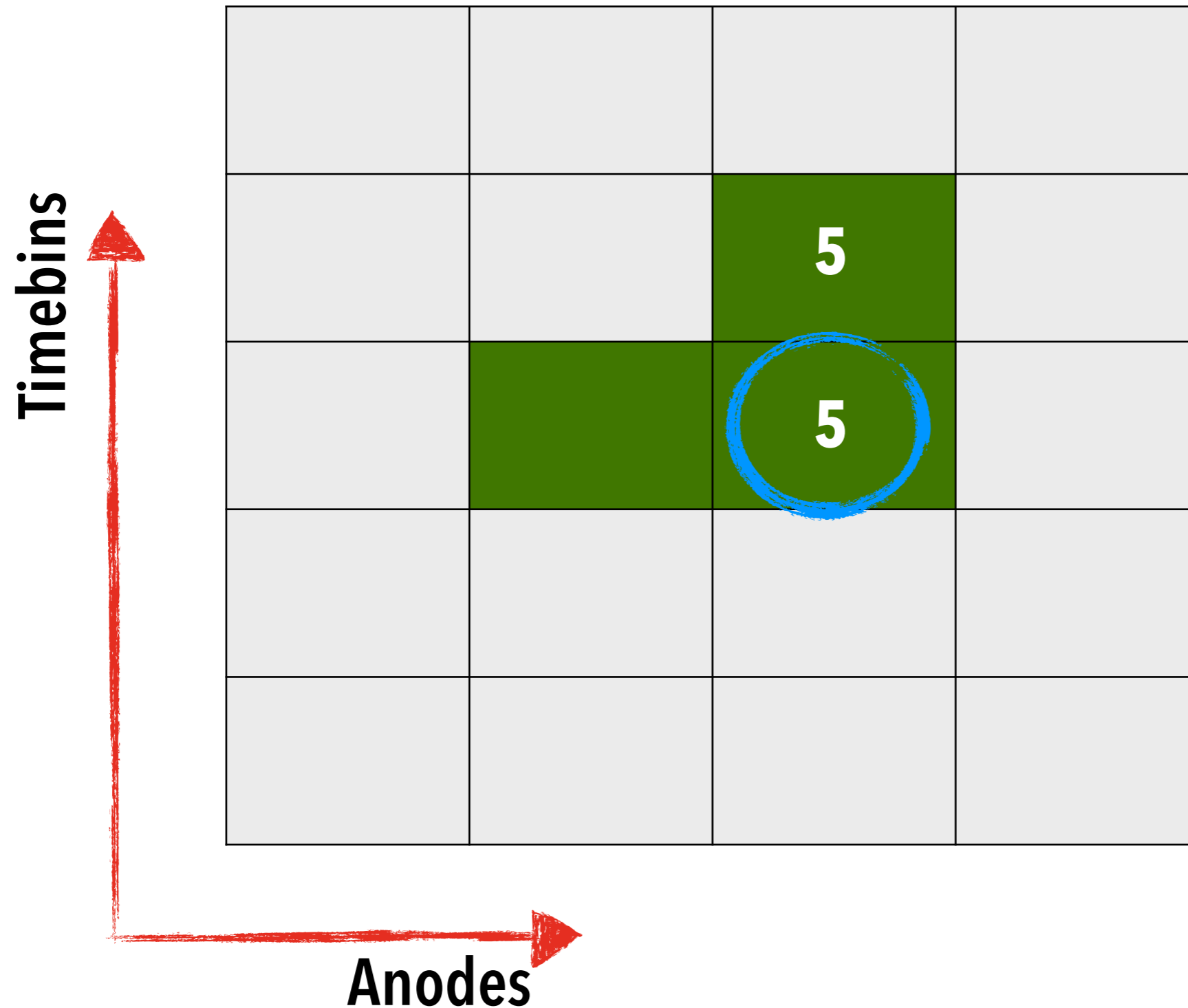
Fast algorithm



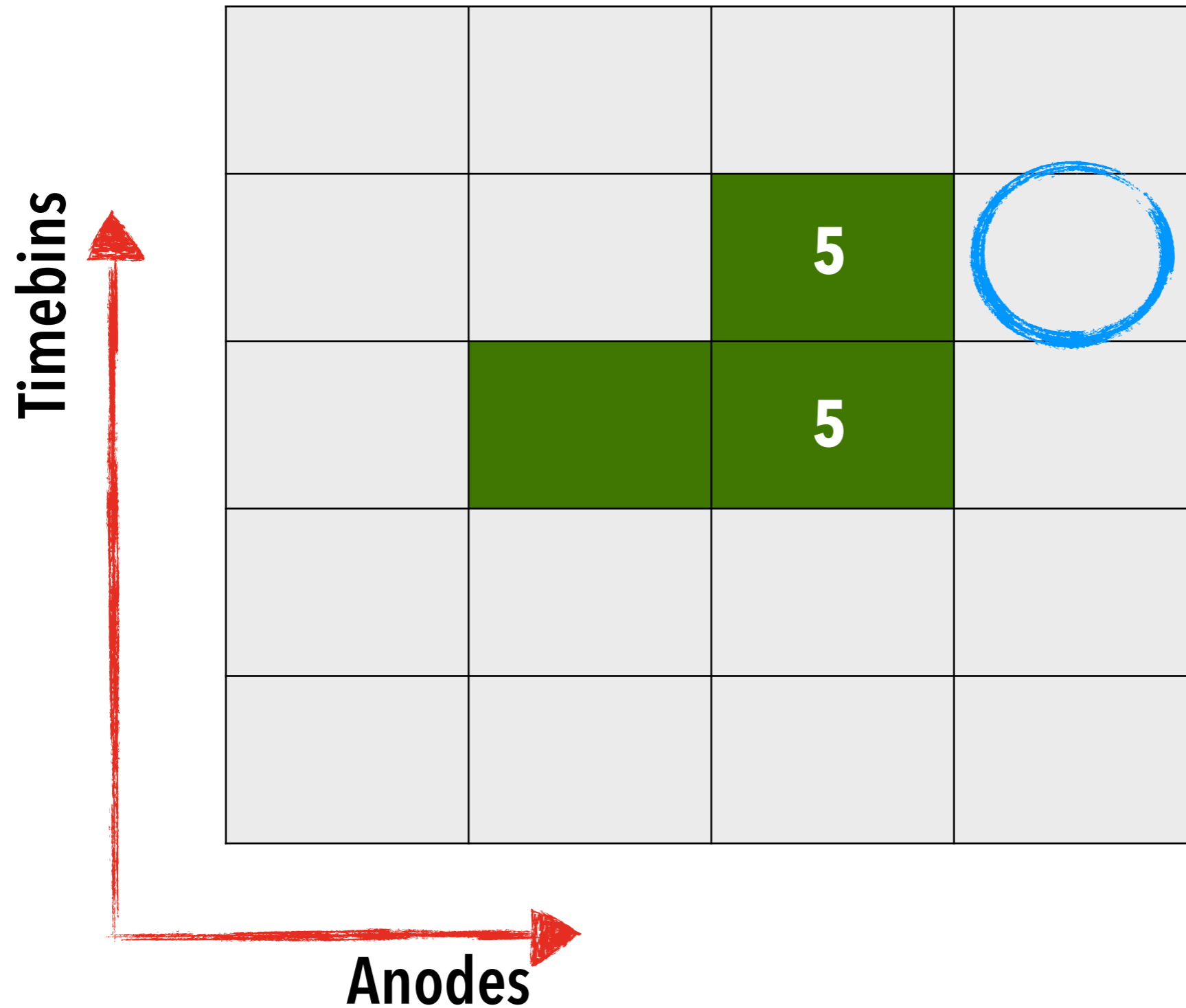
Fast algorithm



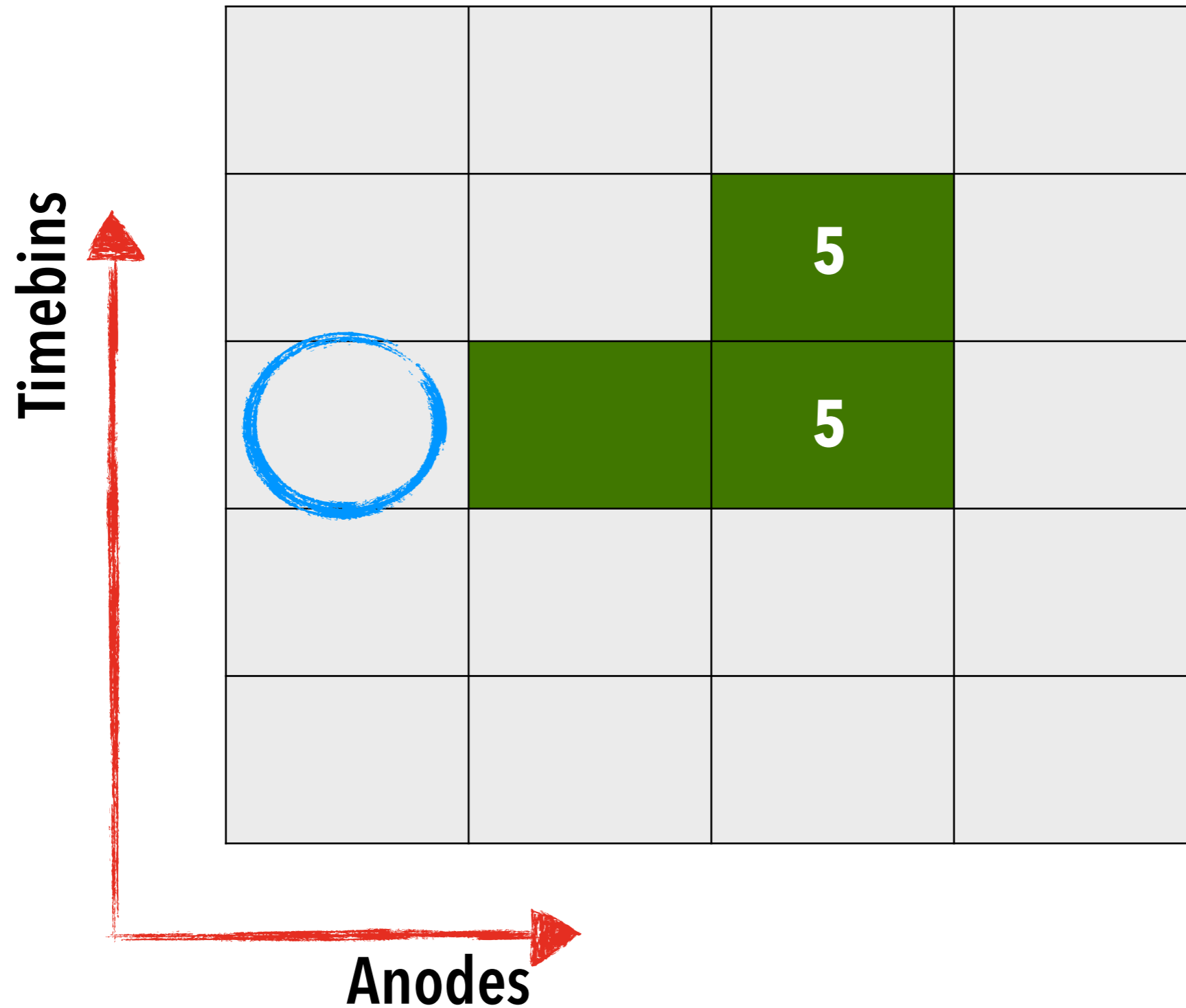
Fast algorithm



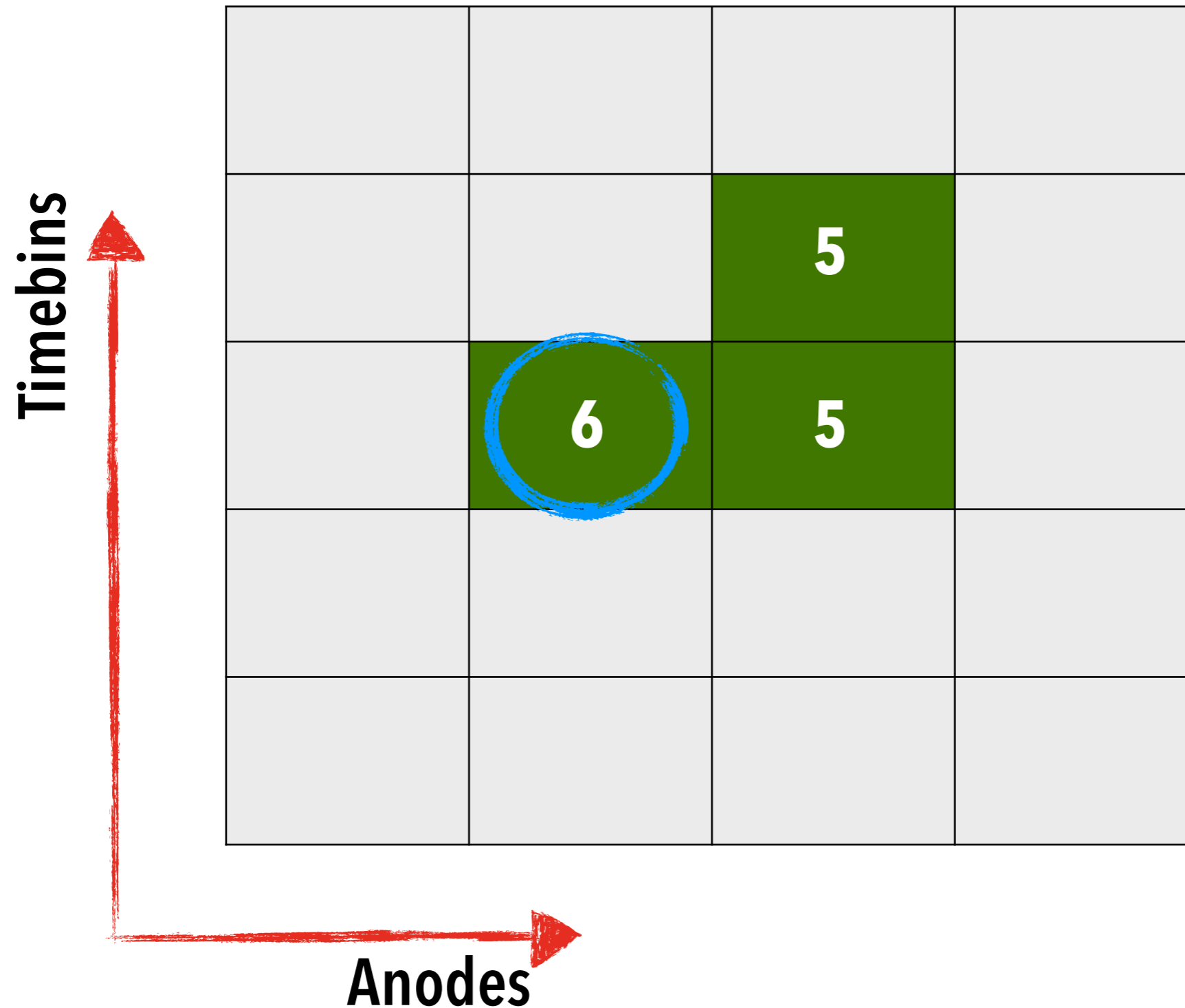
Fast algorithm



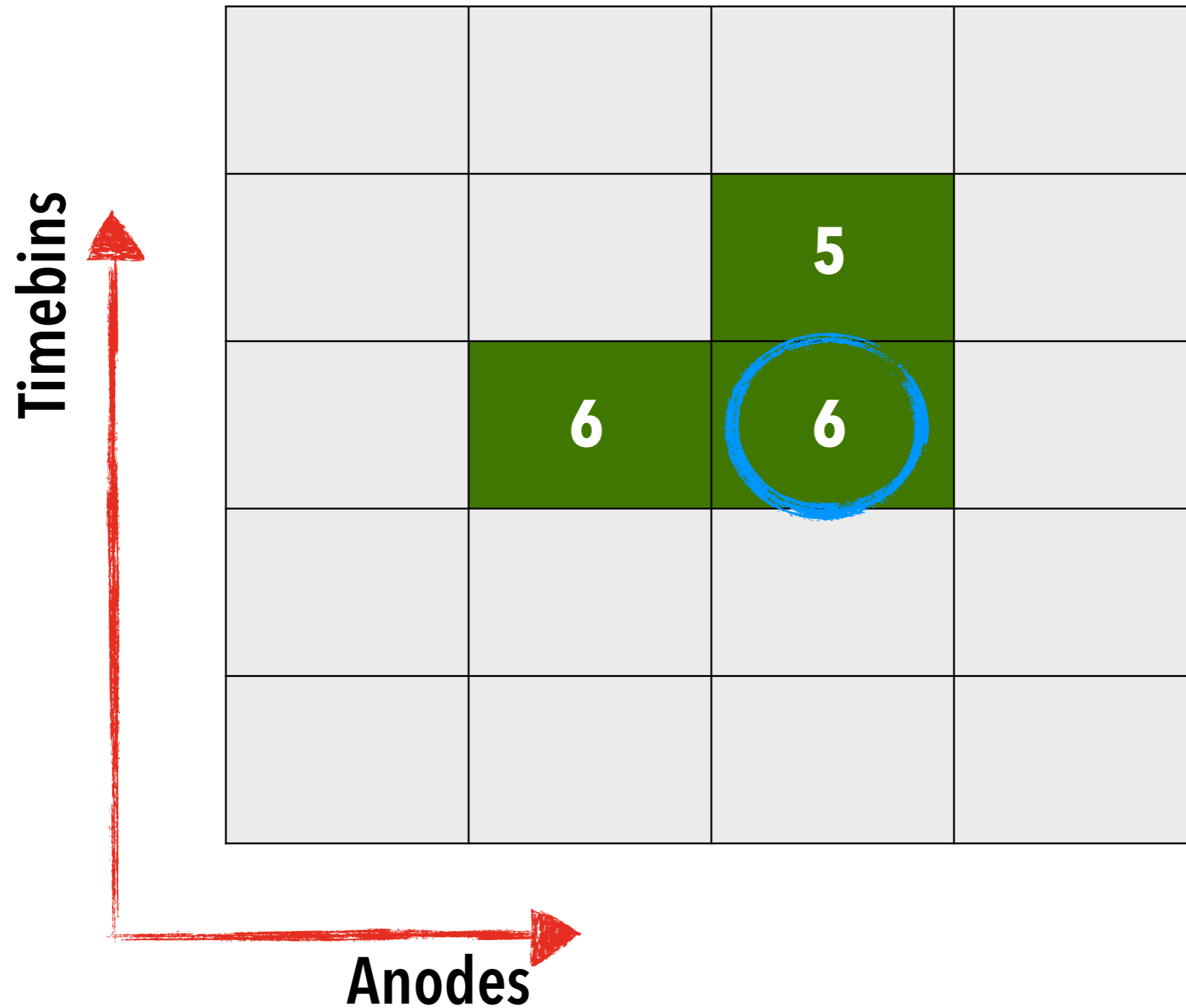
Fast algorithm



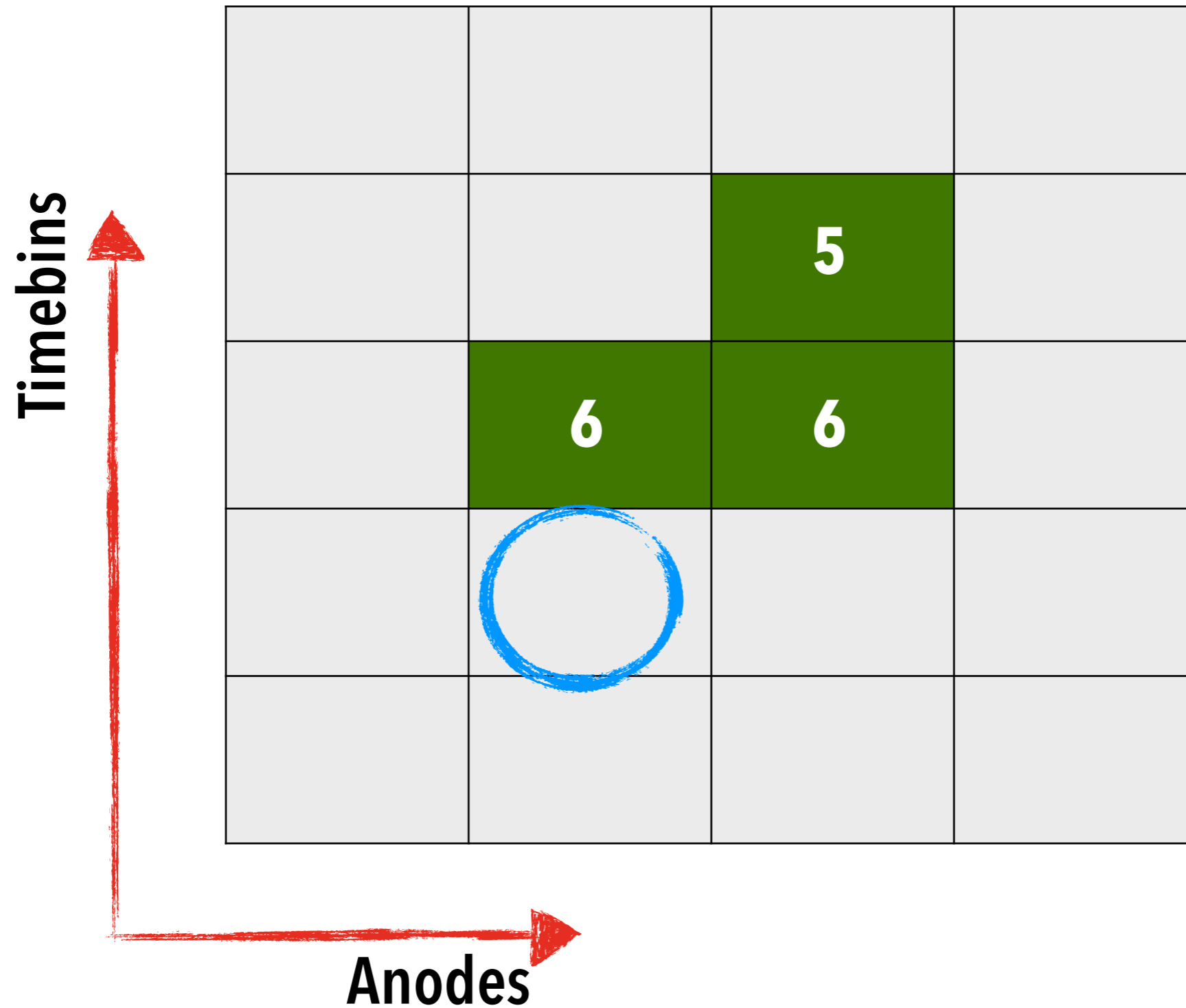
Fast algorithm



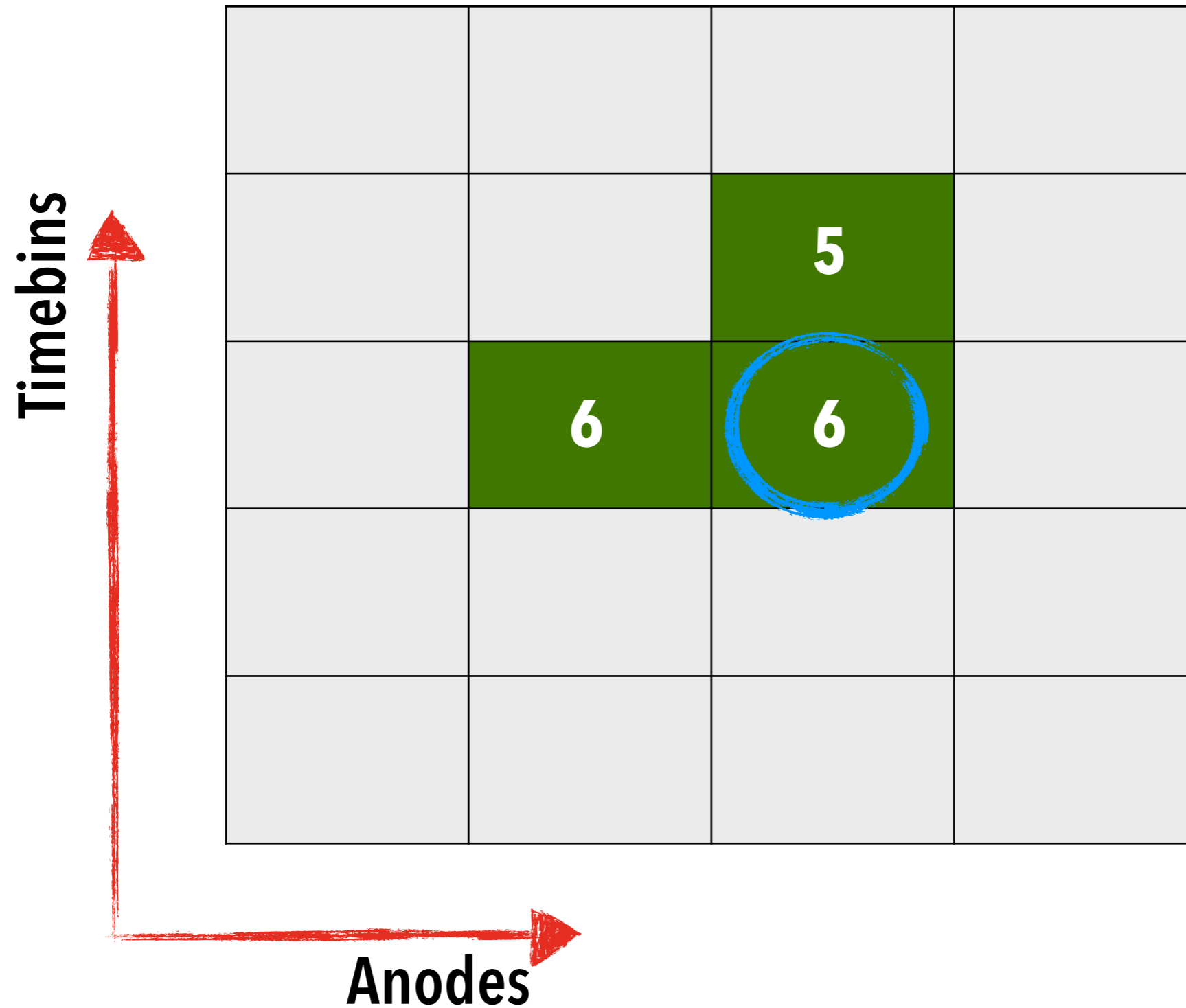
Fast algorithm



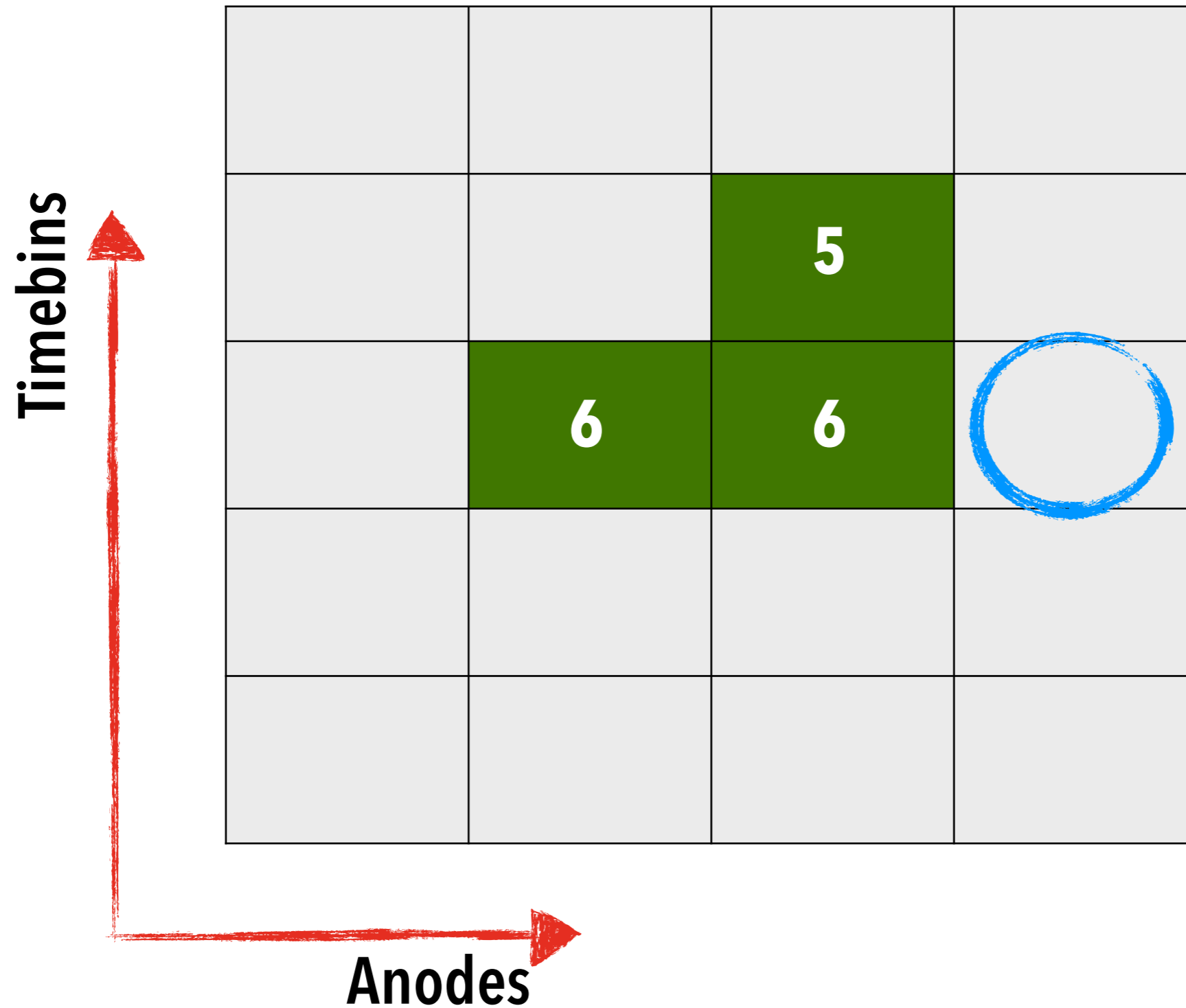
Fast algorithm



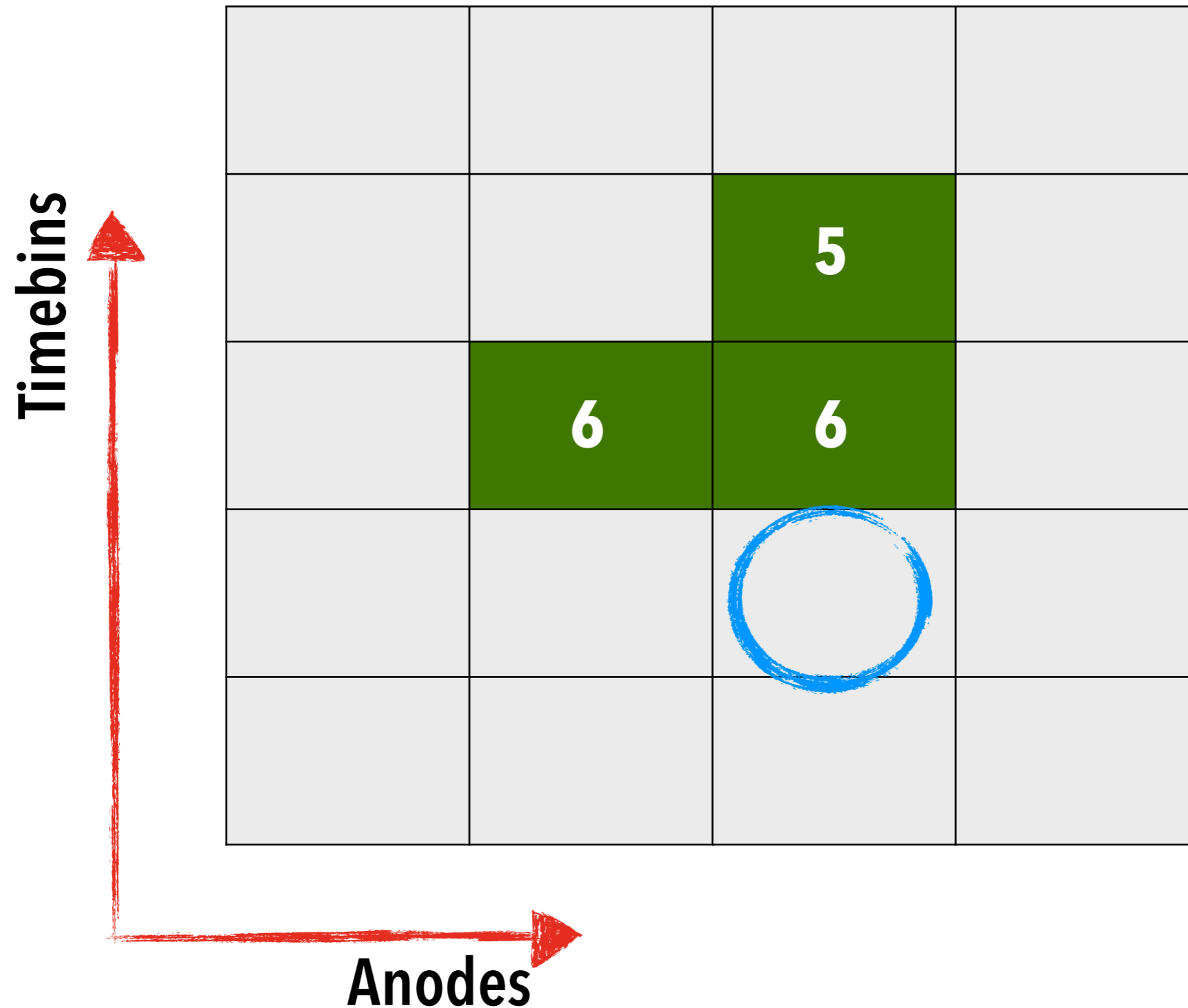
Fast algorithm



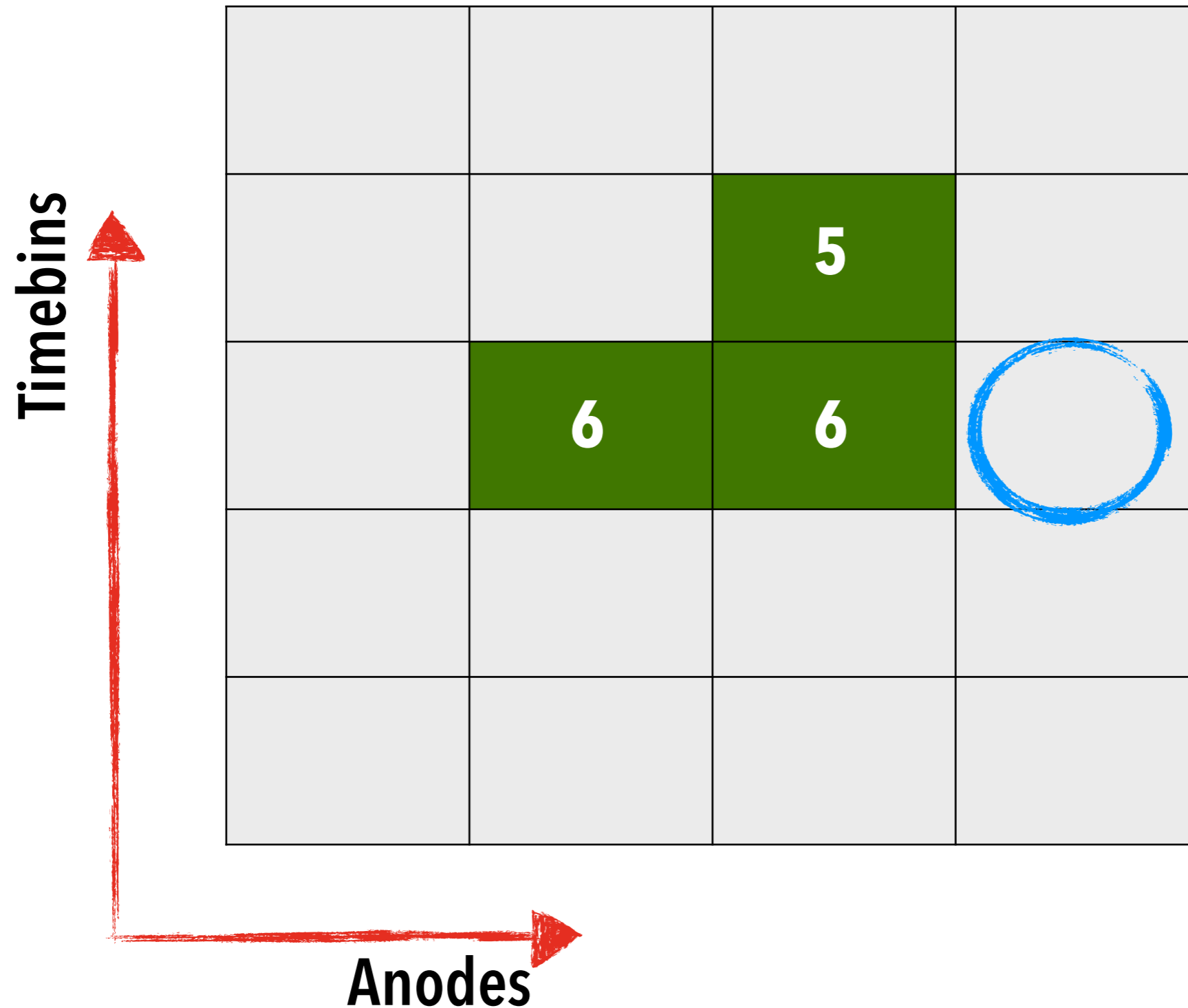
Fast algorithm



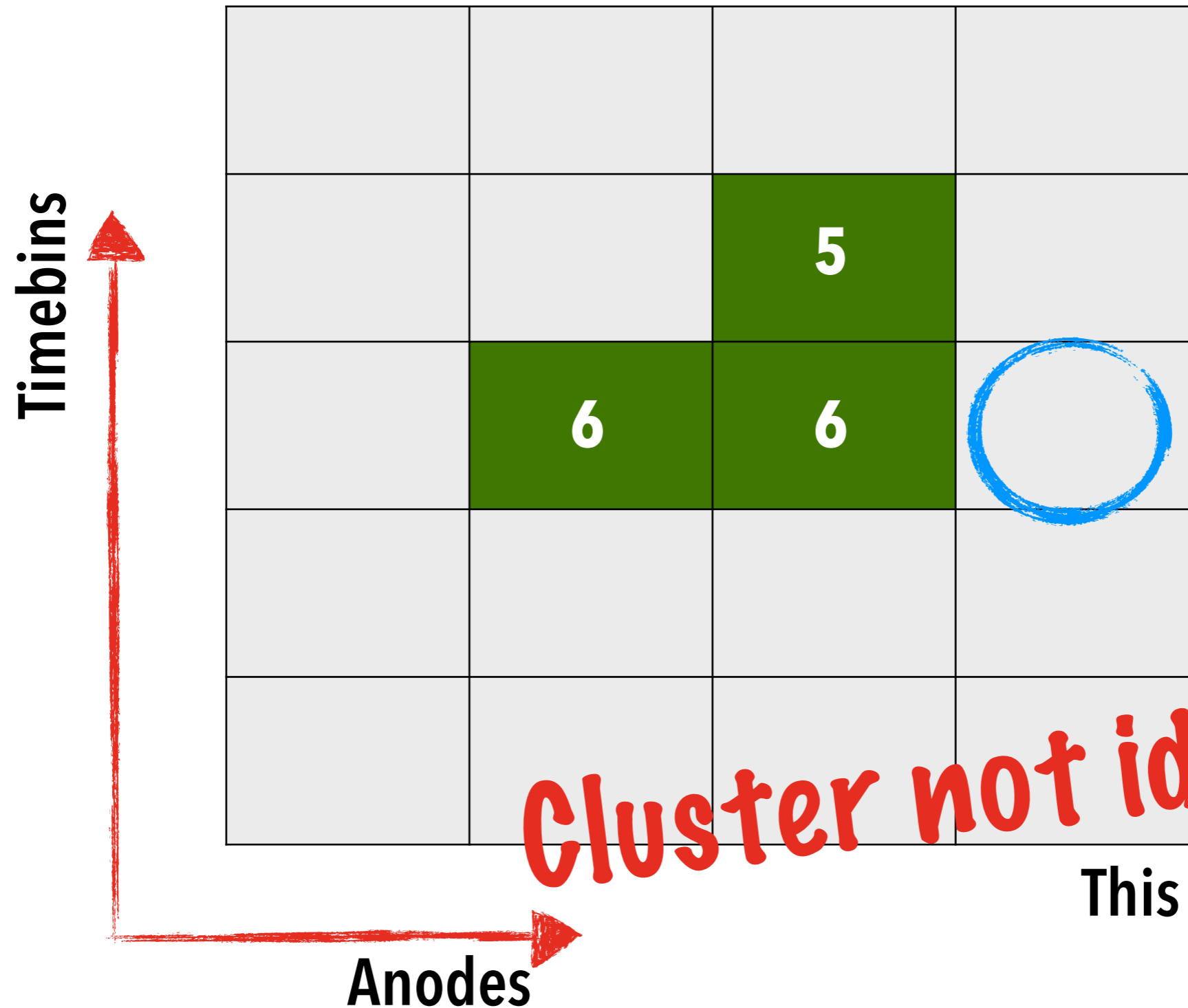
Fast algorithm



Fast algorithm

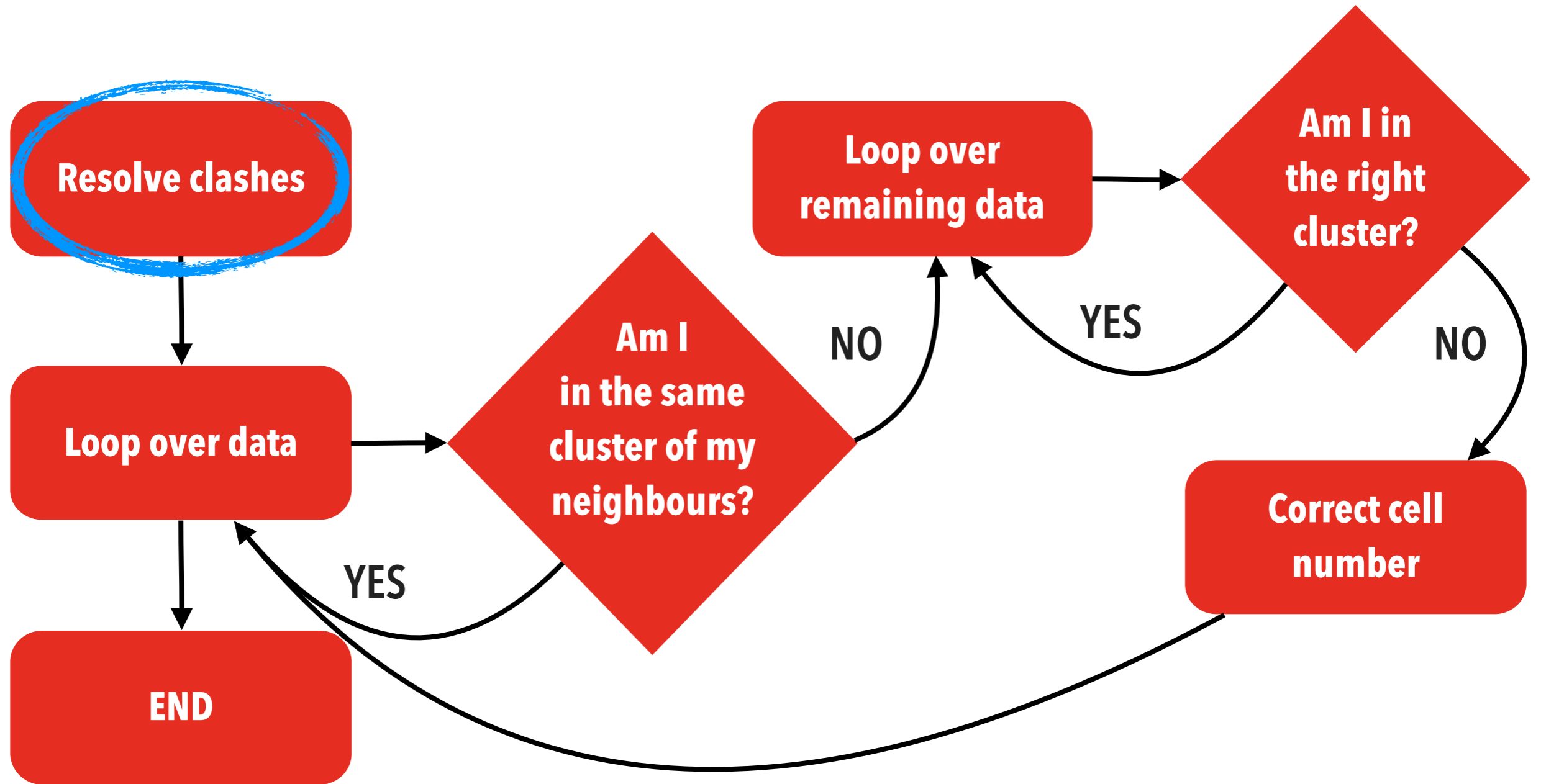


Fast algorithm



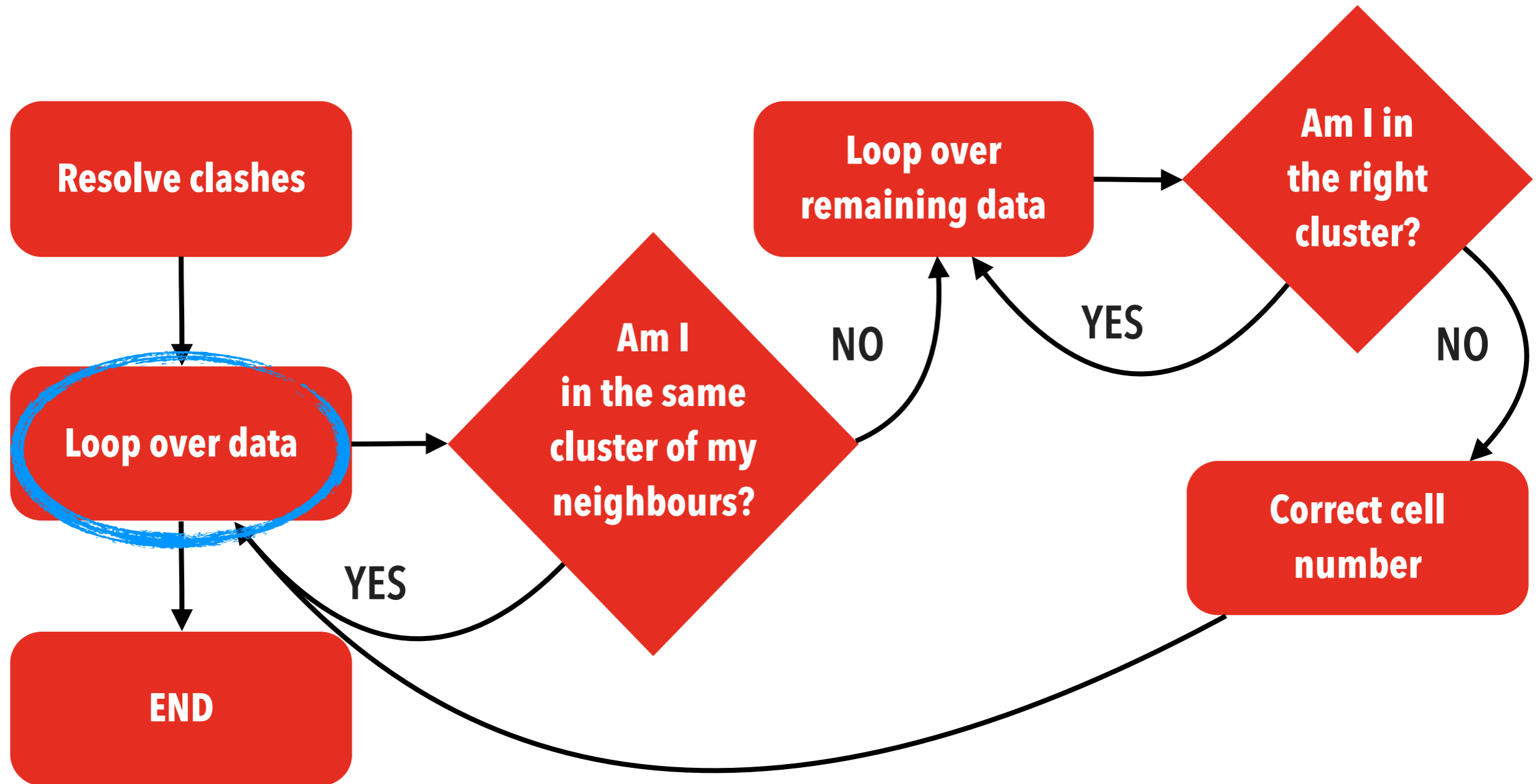
This is a typical clash
to be fixed

Fast algorithm



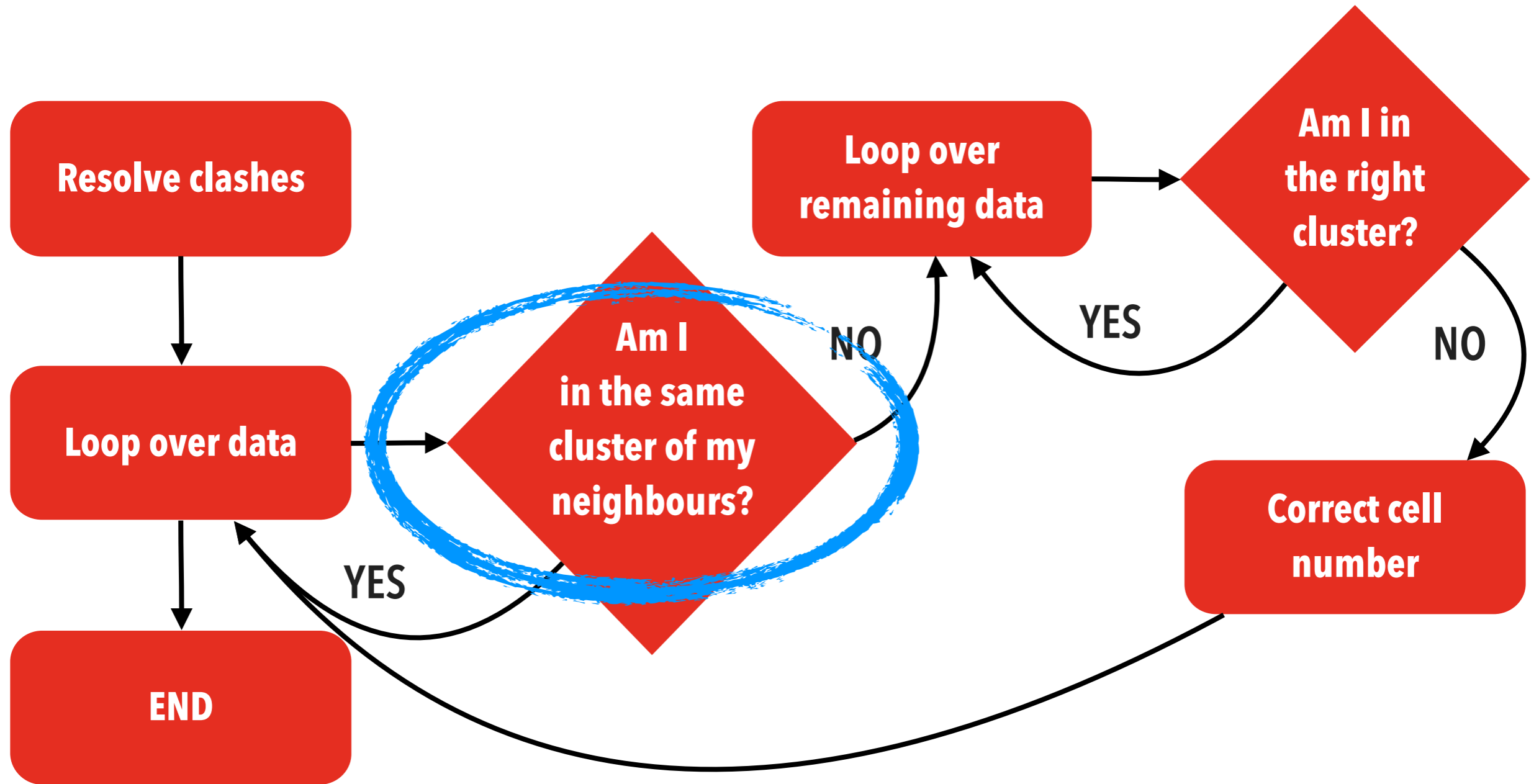
Executed on both sides of
each SDD module

Fast algorithm



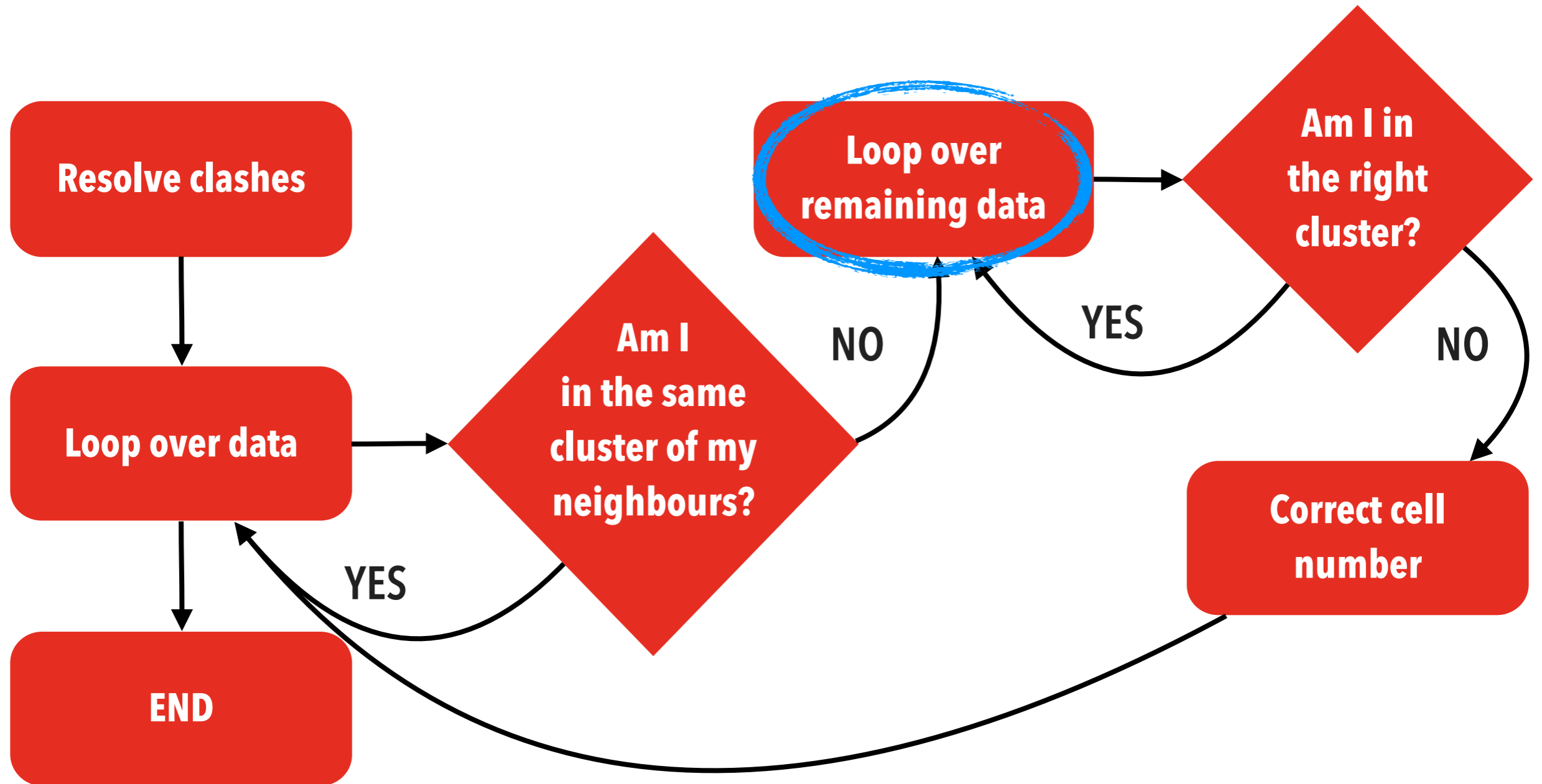
Executed on both sides of
each SDD module

Fast algorithm



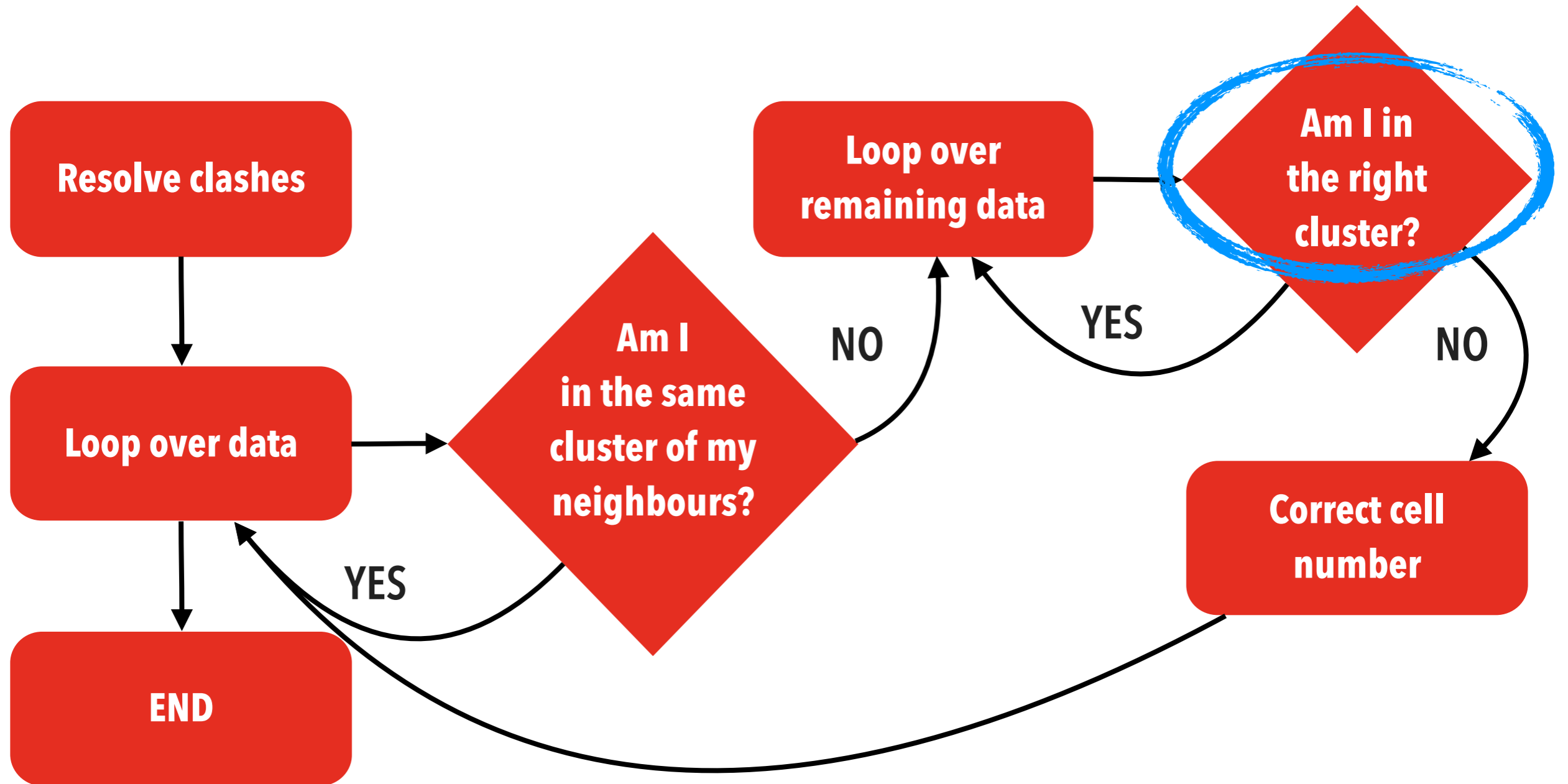
Executed on both sides of
each SDD module

Fast algorithm



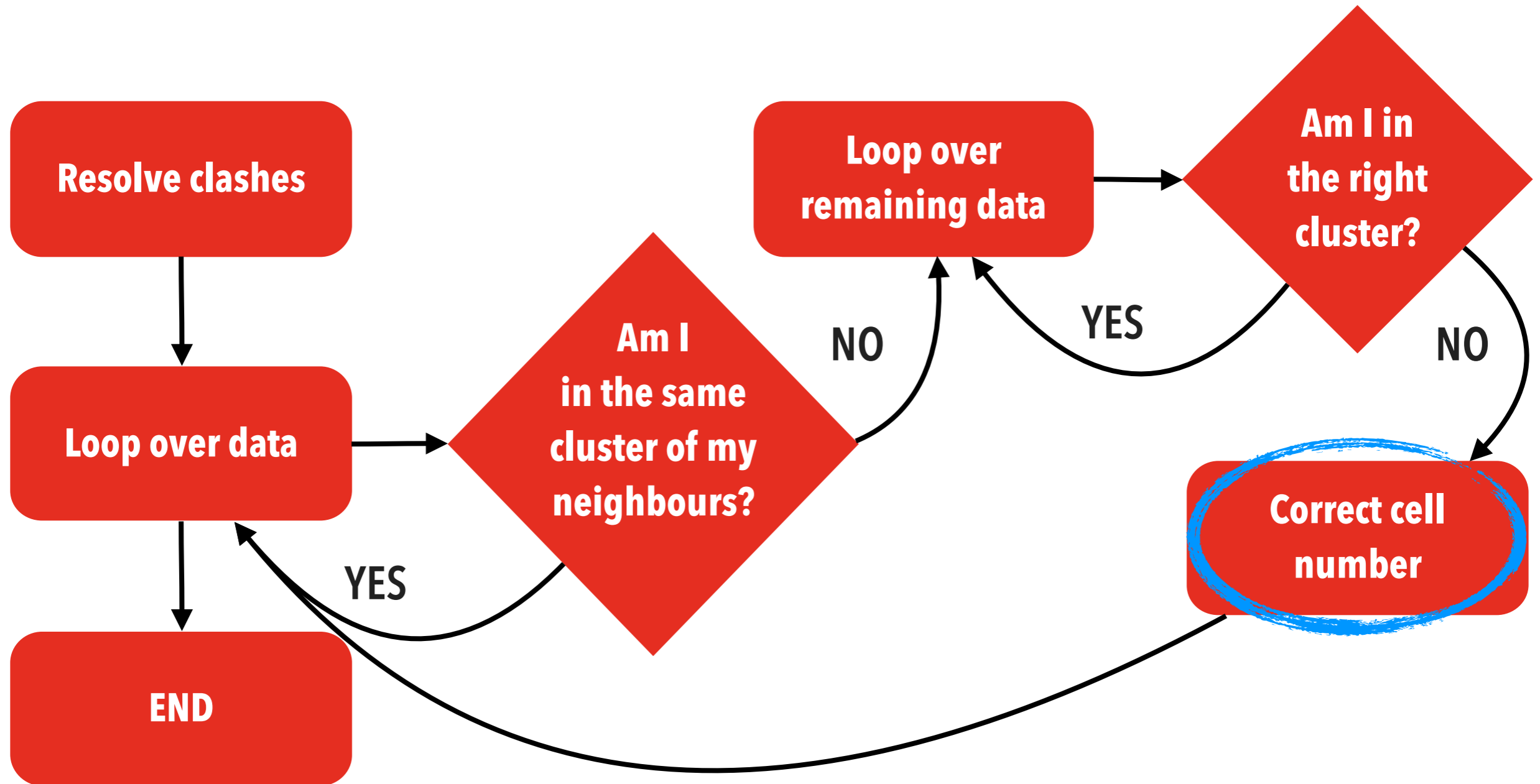
Executed on both sides of
each SDD module

Fast algorithm



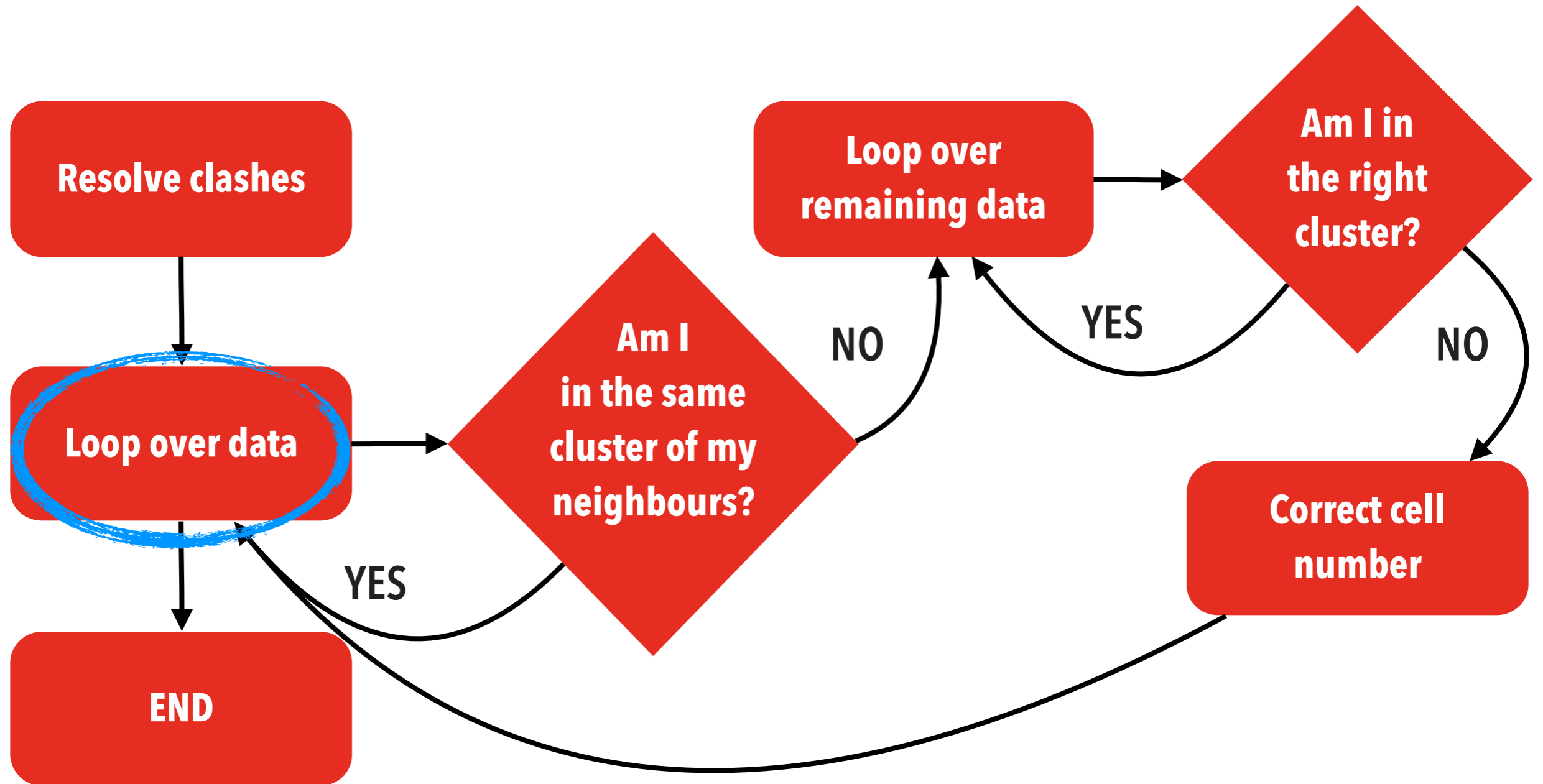
Executed on both sides of
each SDD module

Fast algorithm



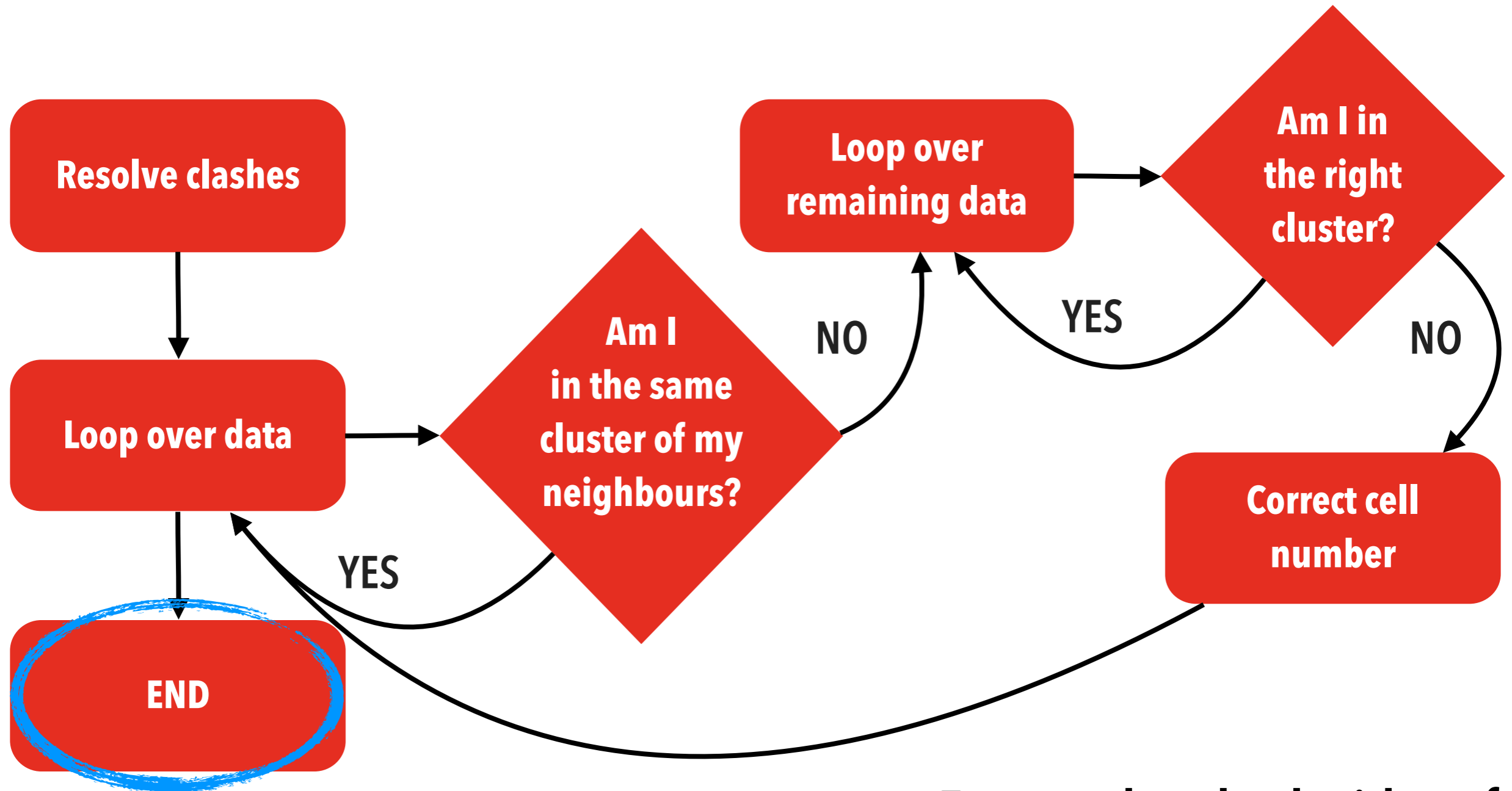
Executed on both sides of
each SDD module

Fast algorithm



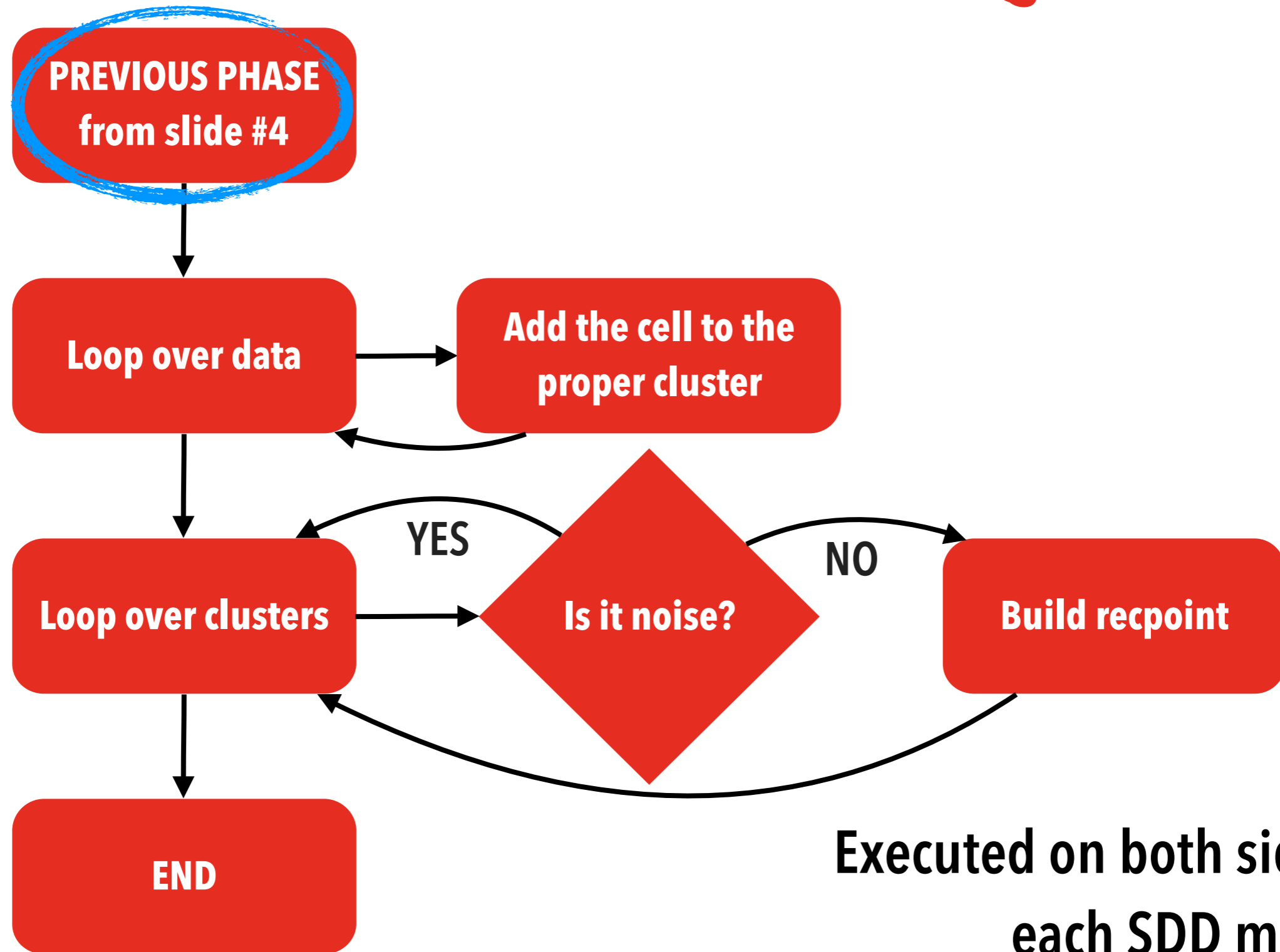
Executed on both sides of
each SDD module

Fast algorithm



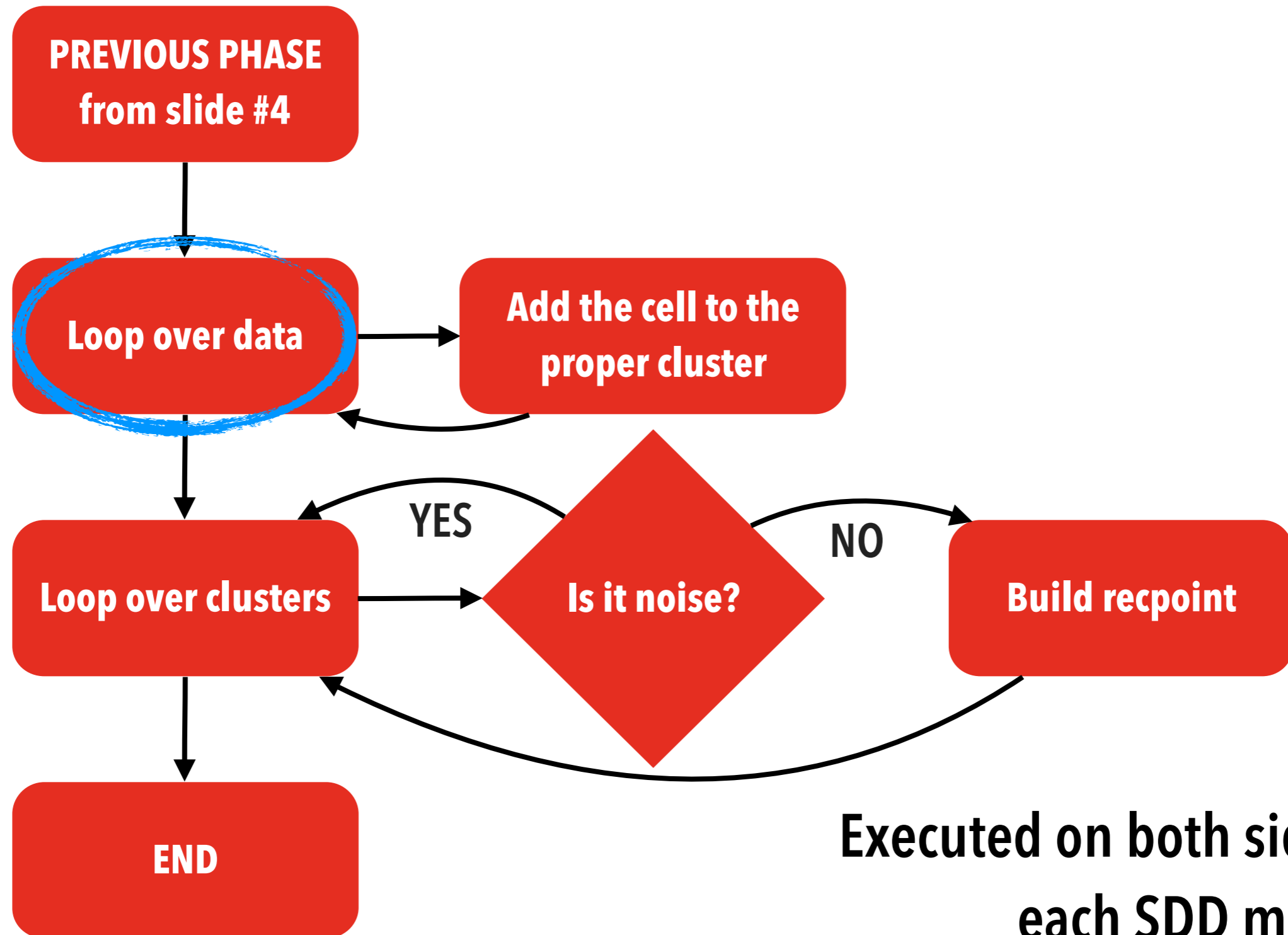
Executed on both sides of
each SDD module

Fast algorithm



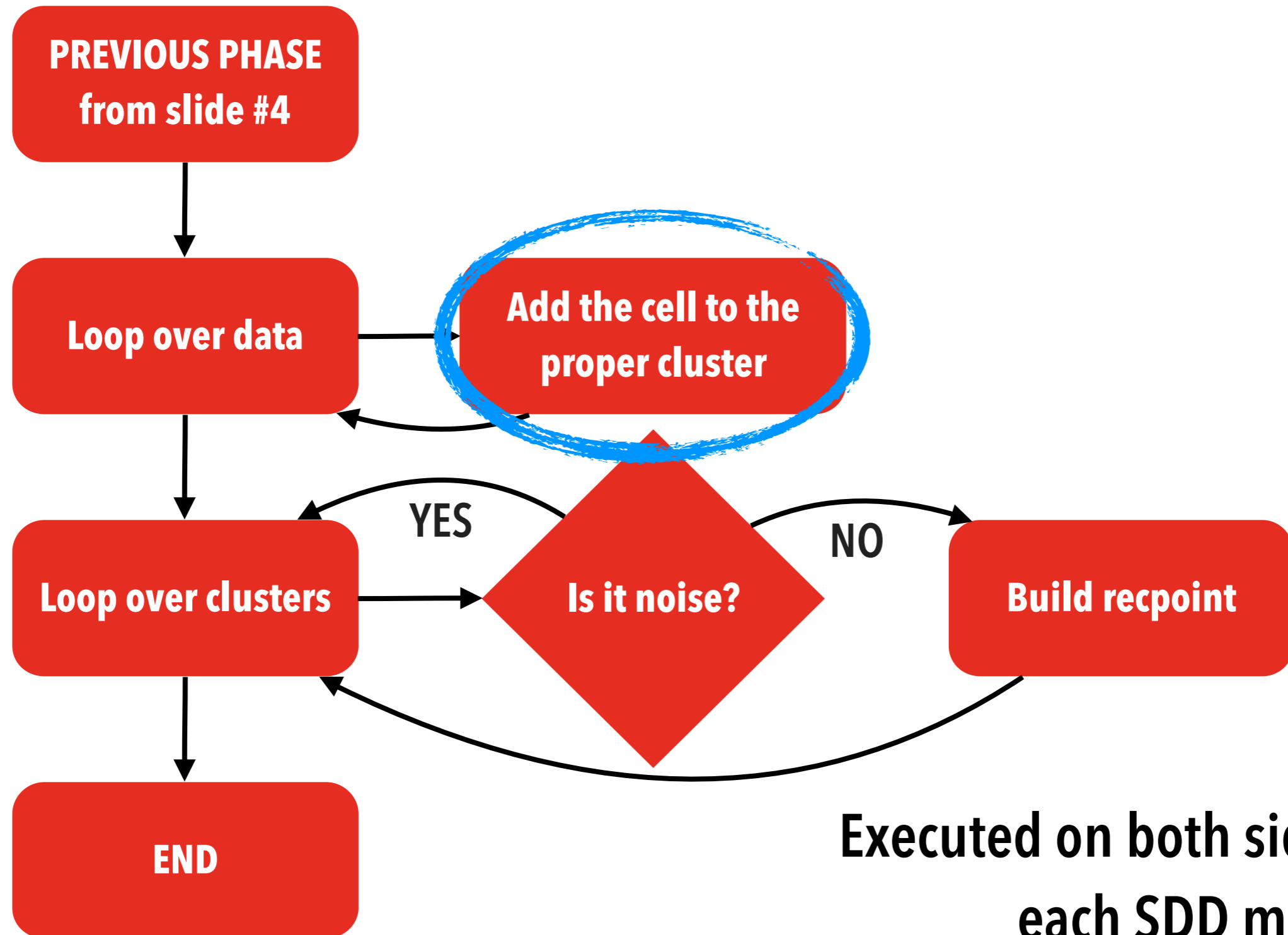
Executed on both sides of
each SDD module

Fast algorithm



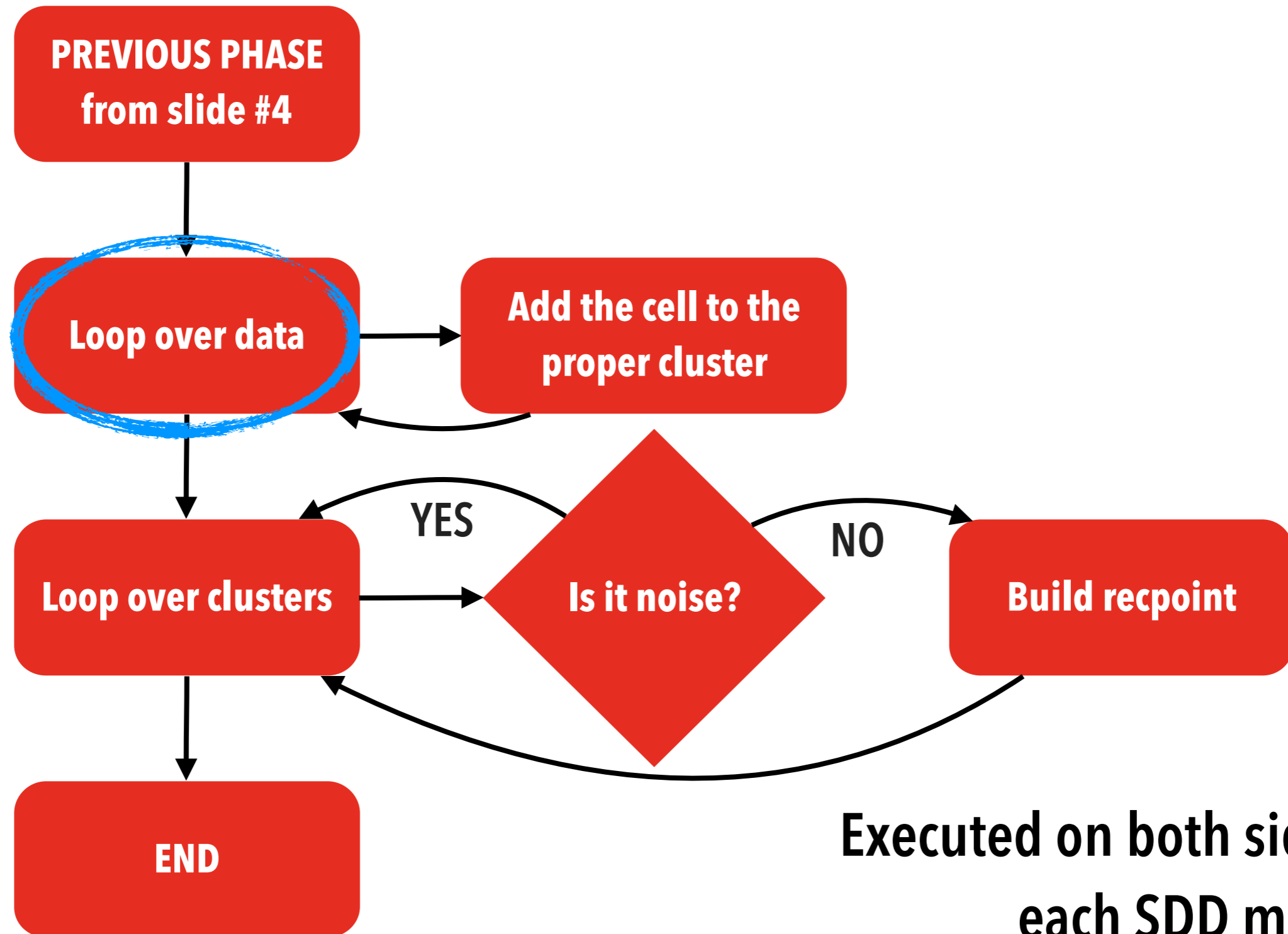
Executed on both sides of
each SDD module

Fast algorithm



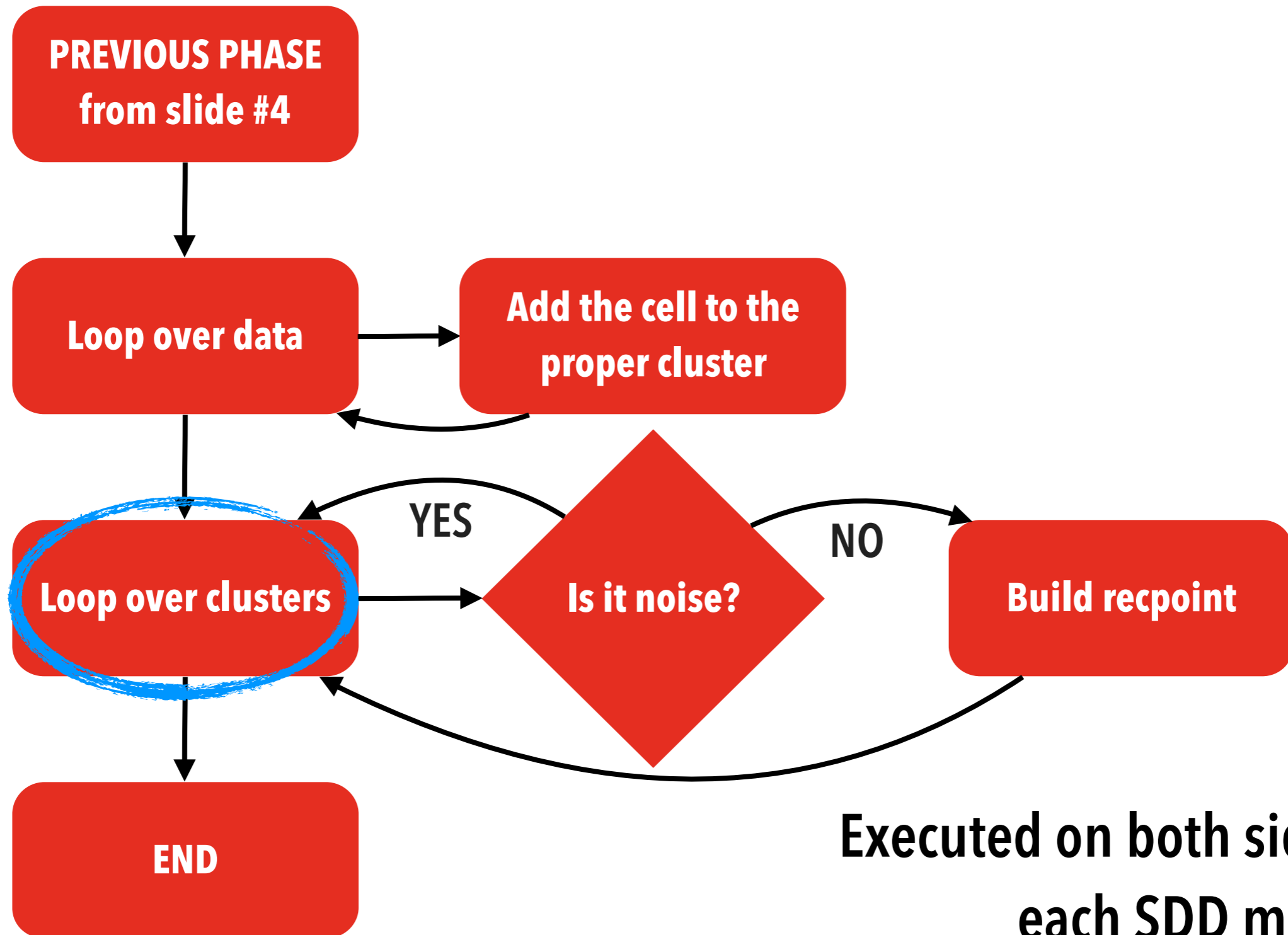
Executed on both sides of
each SDD module

Fast algorithm



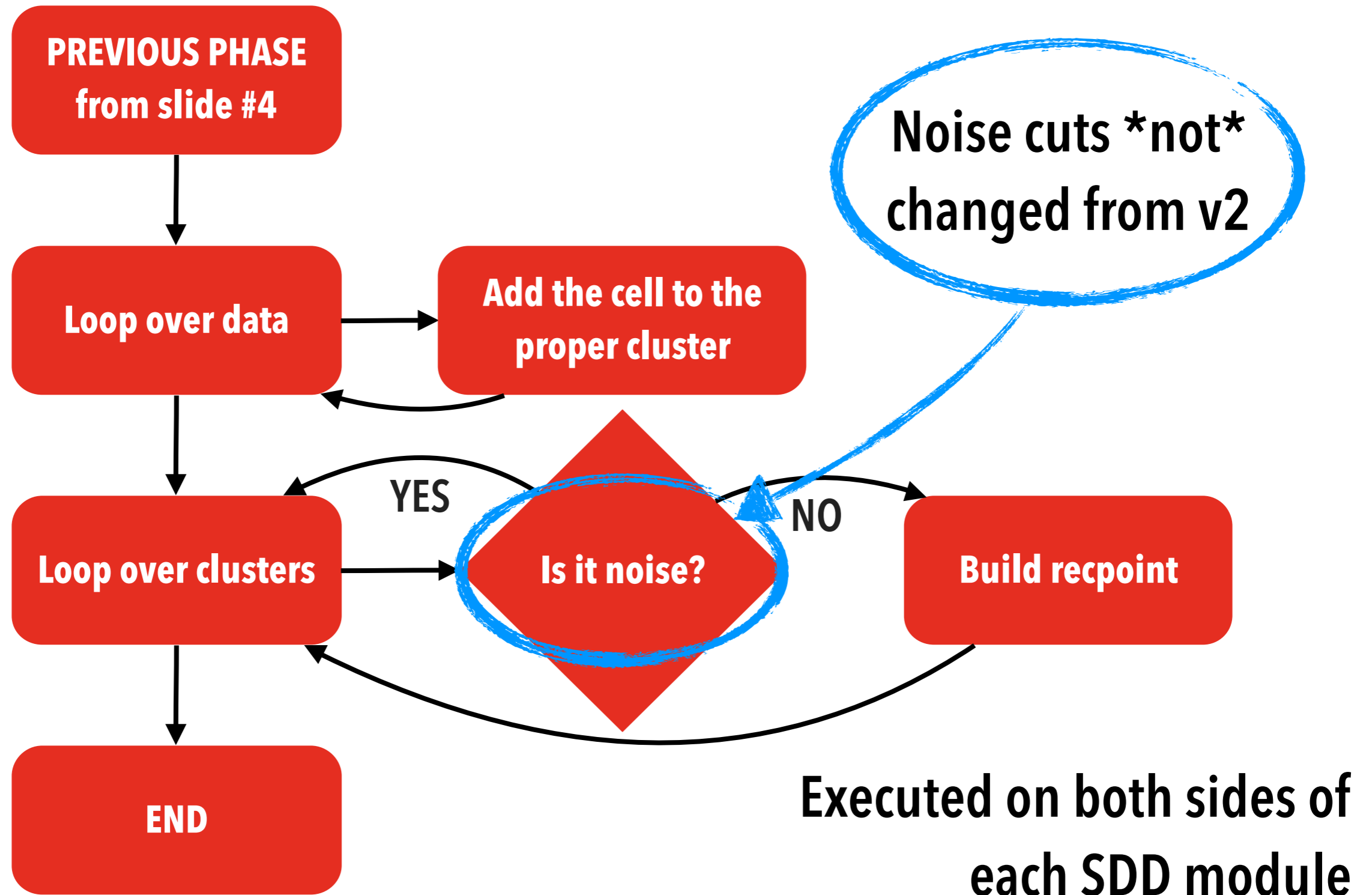
Executed on both sides of
each SDD module

Fast algorithm

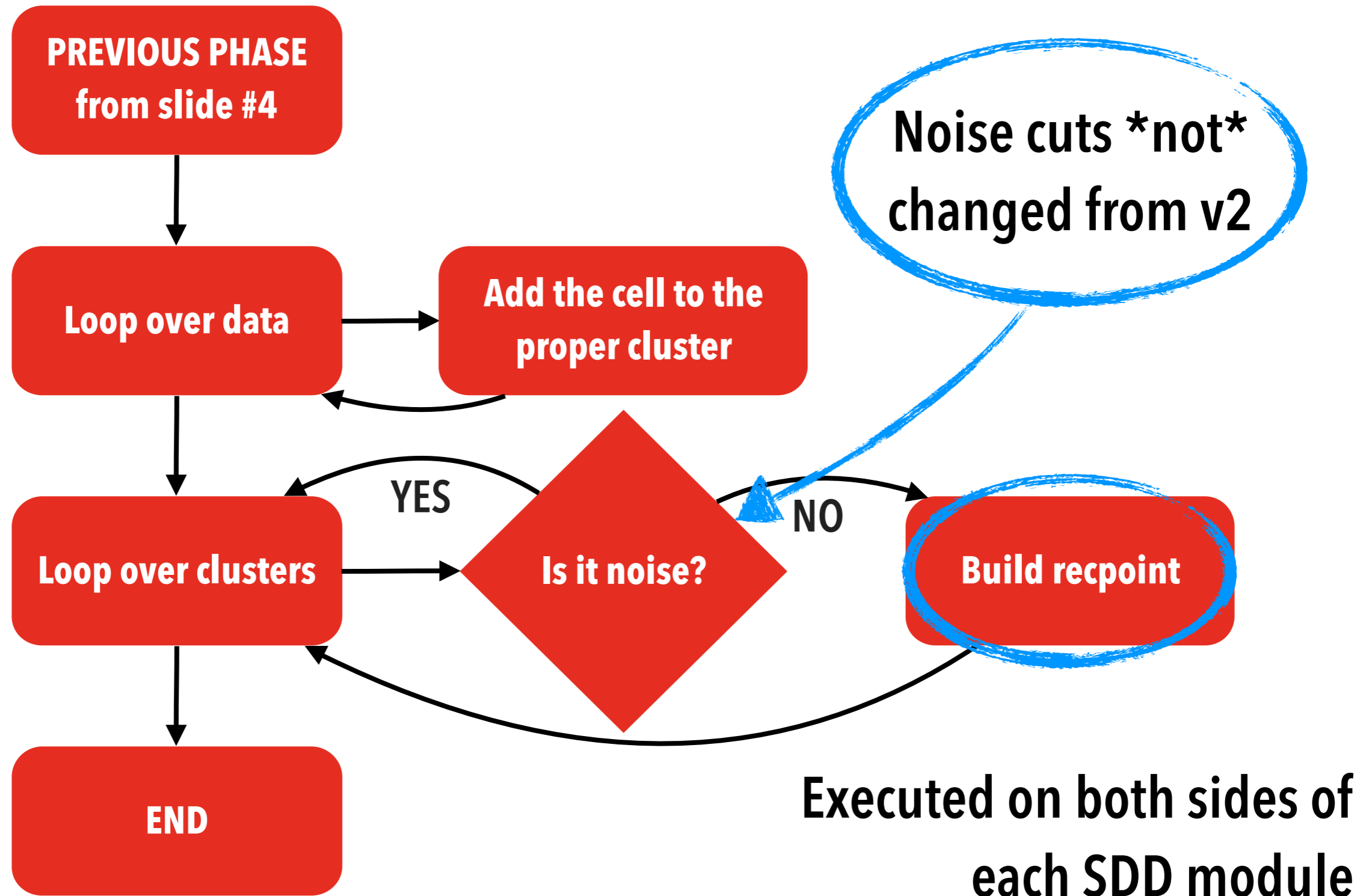


Executed on both sides of
each SDD module

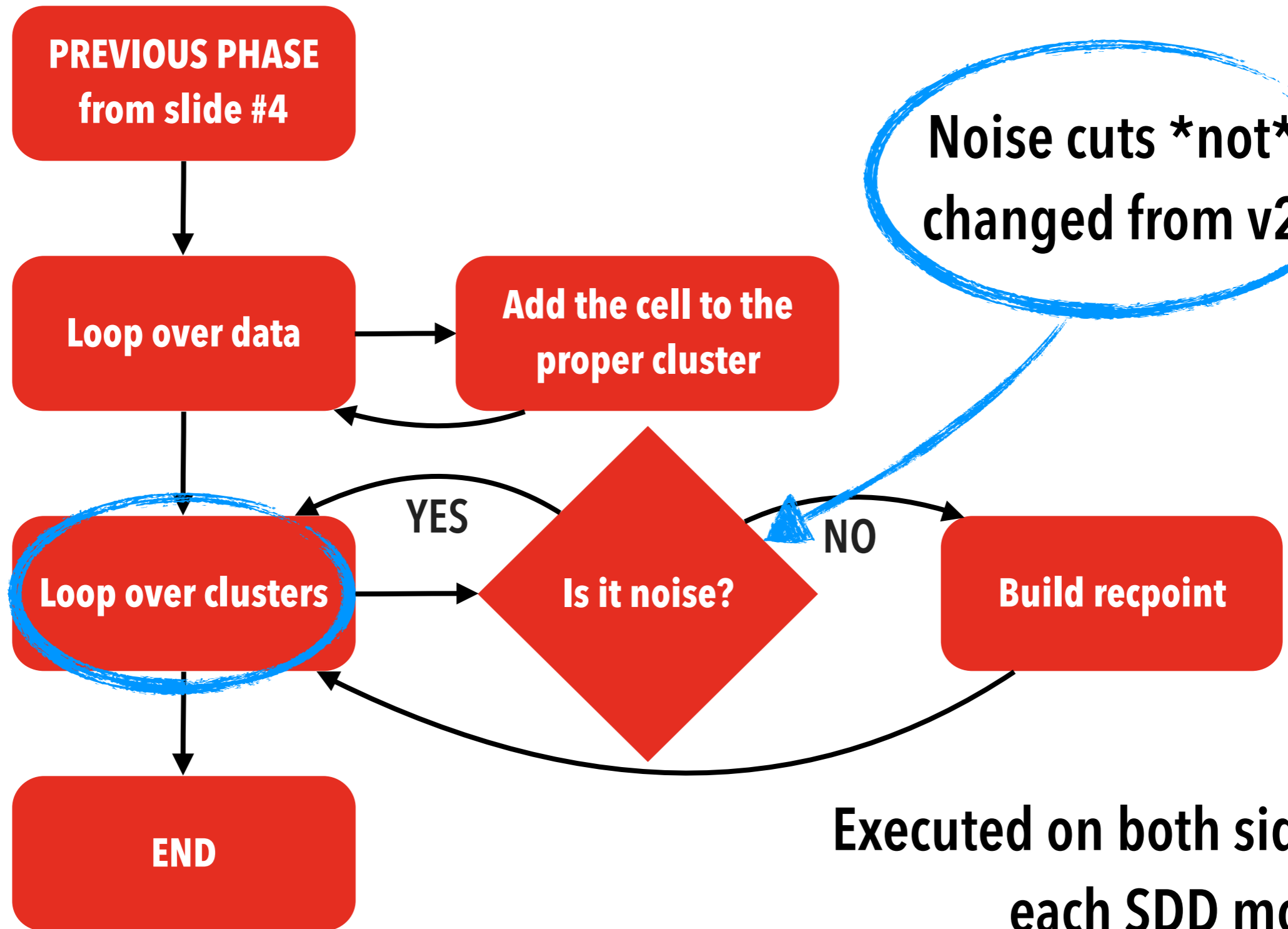
Fast algorithm



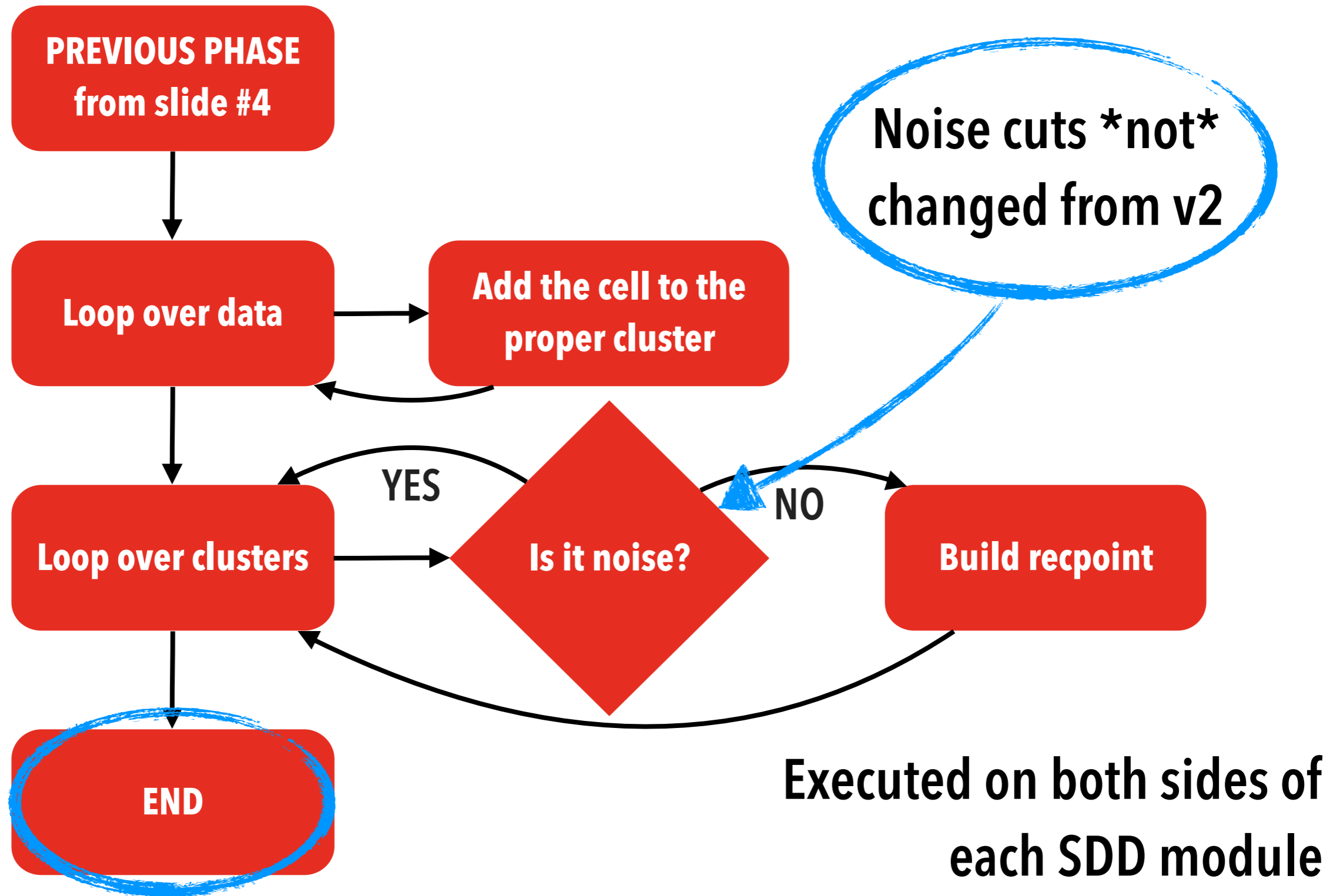
Fast algorithm



Fast algorithm



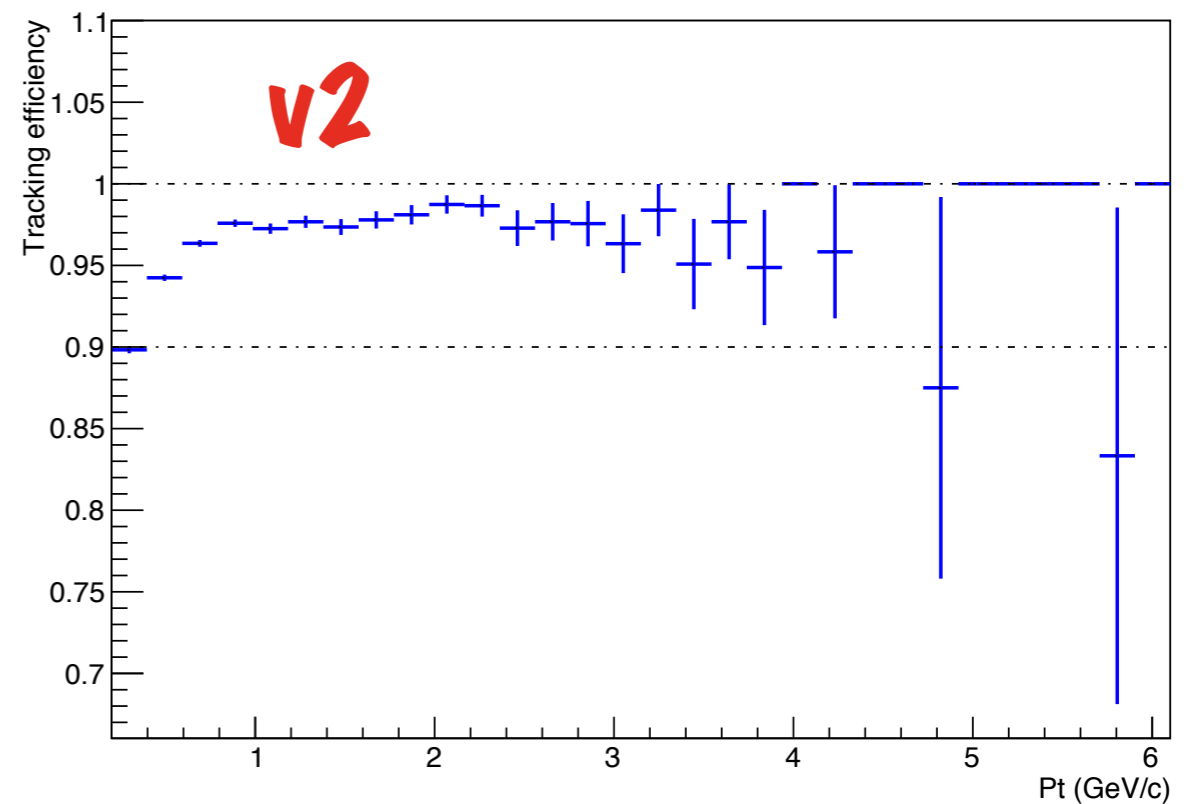
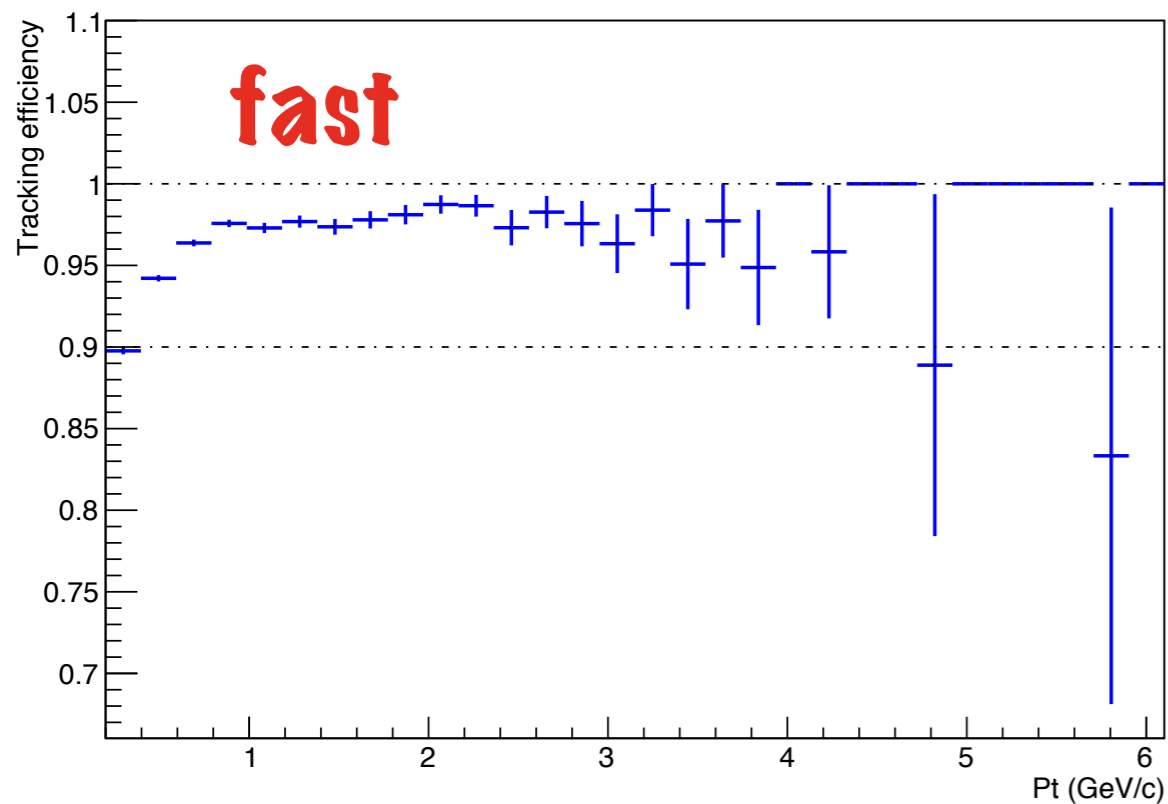
Fast algorithm



Reconstruction performance

Pb-Pb MonteCarlo

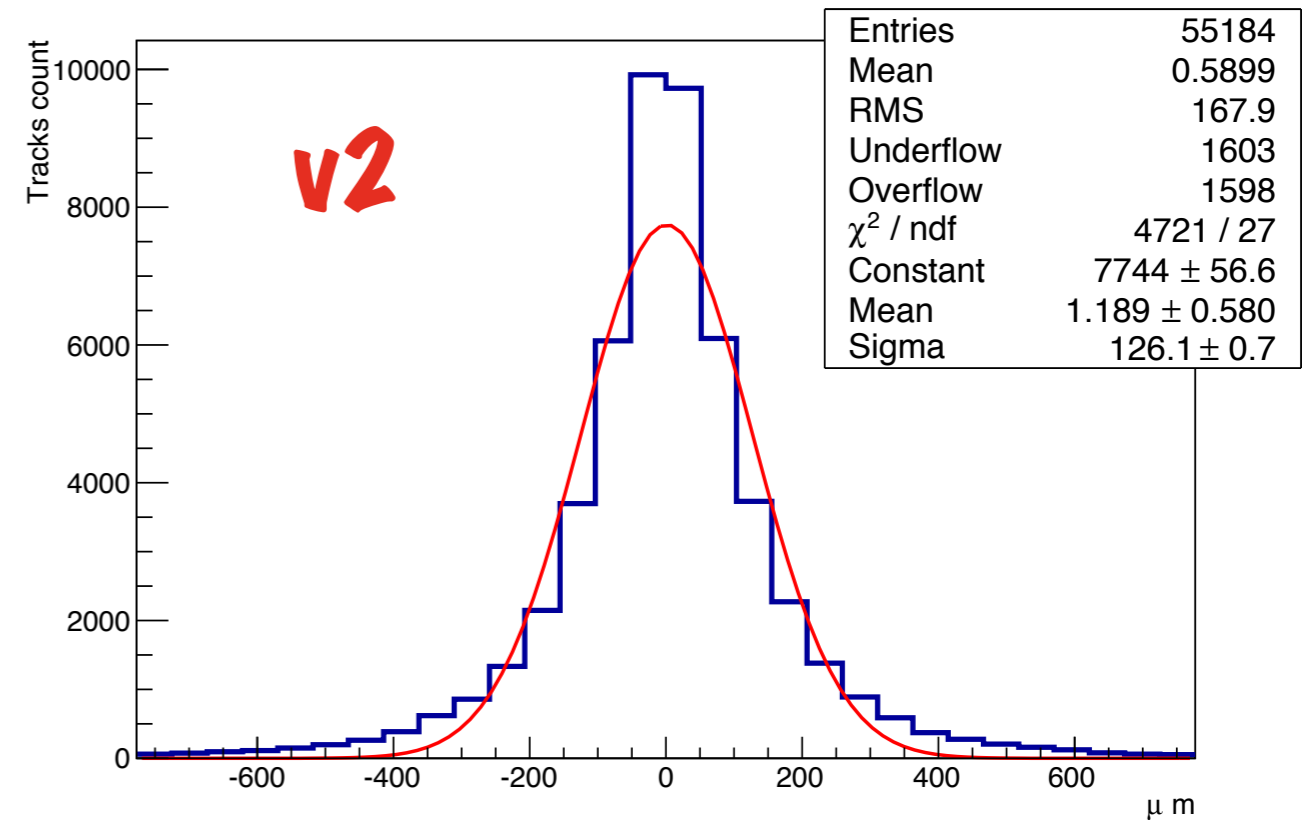
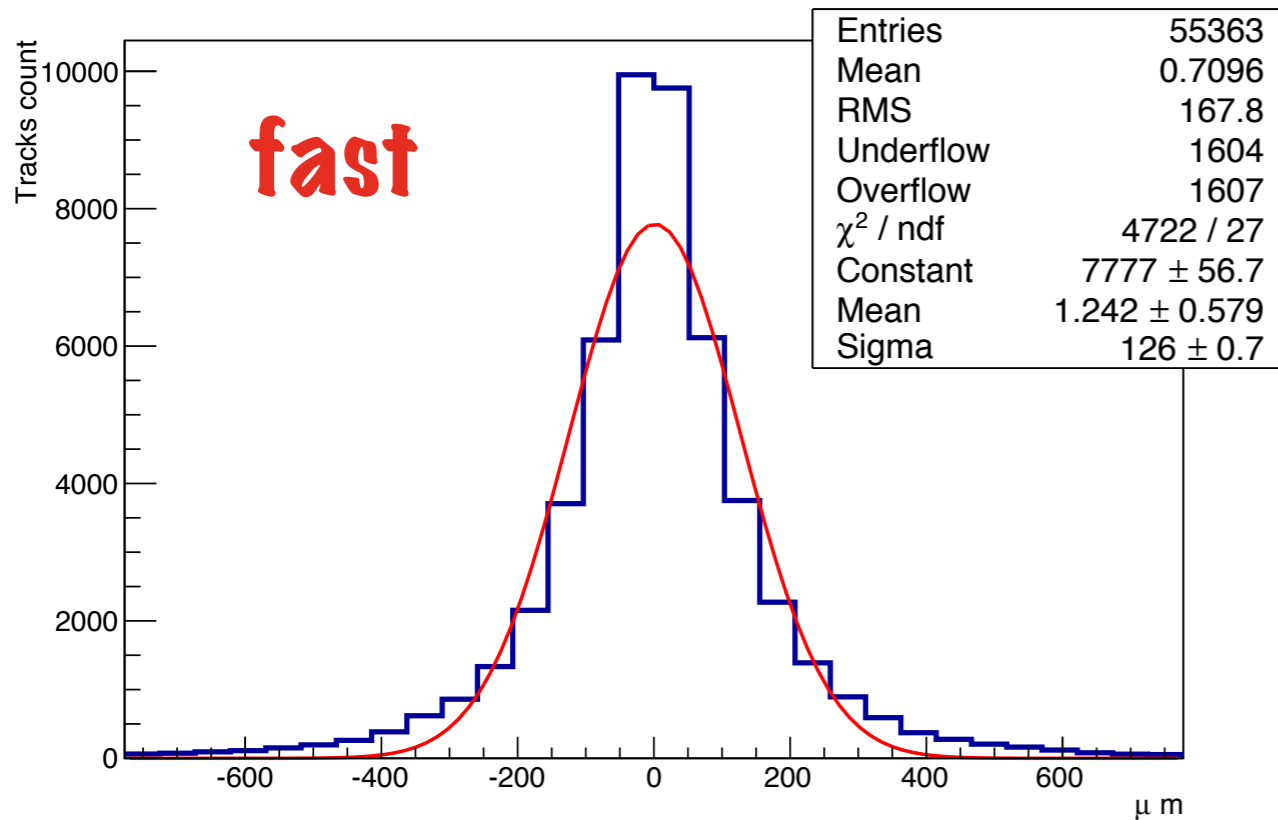
Data set: 92 central MC events



Almost the same tracking efficiency using both CF

Pb-Pb MonteCarlo

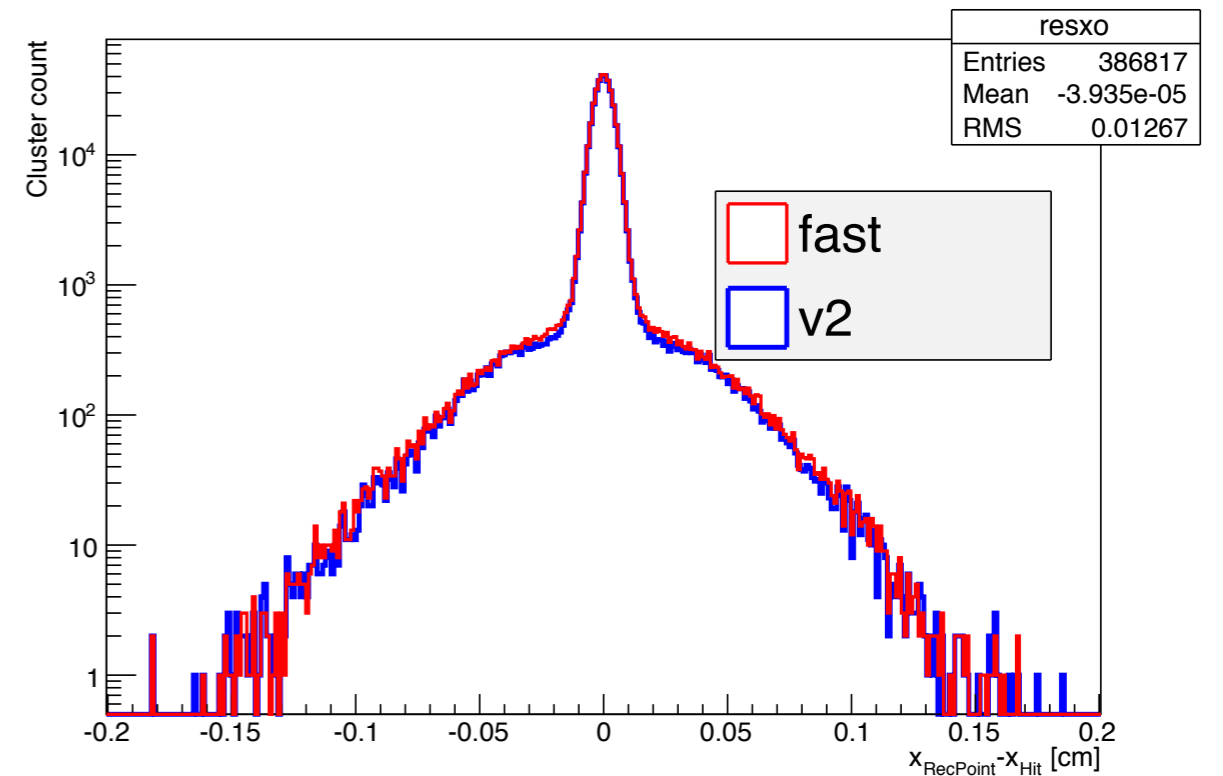
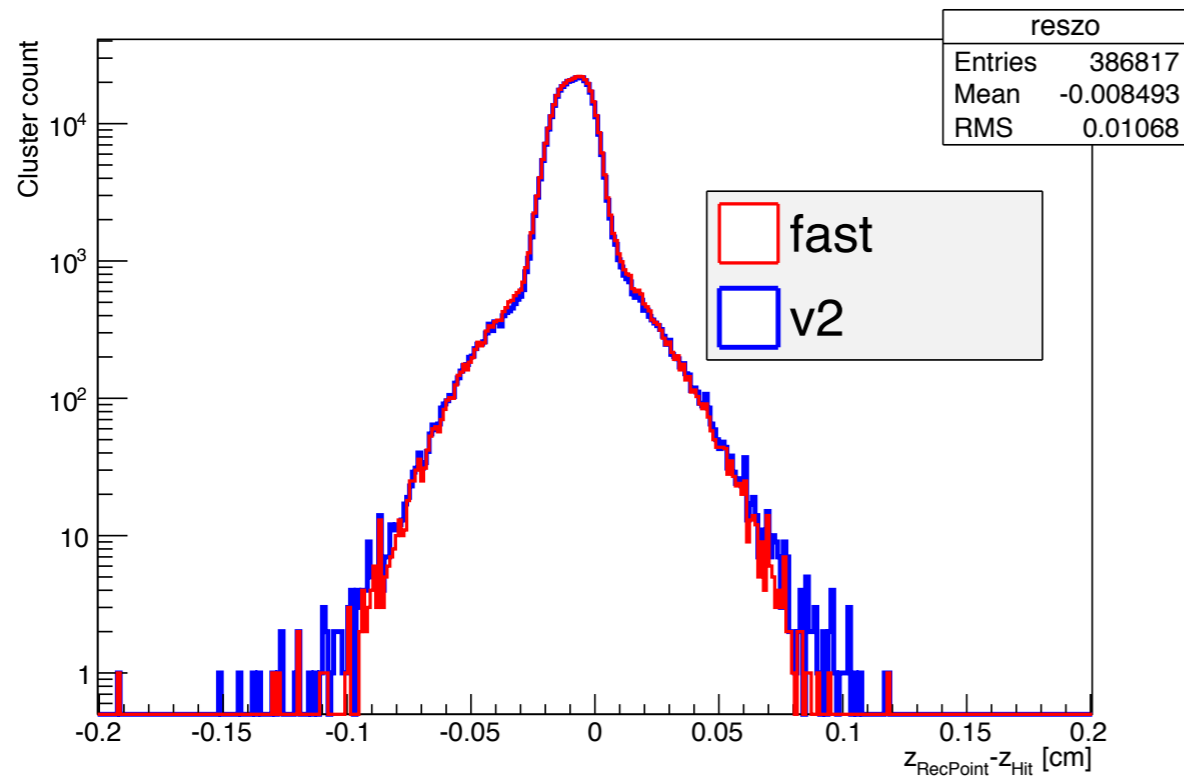
Data set: 92 central MC events



Also the impact parameter resolution is quite the same

Pb-Pb MonteCarlo

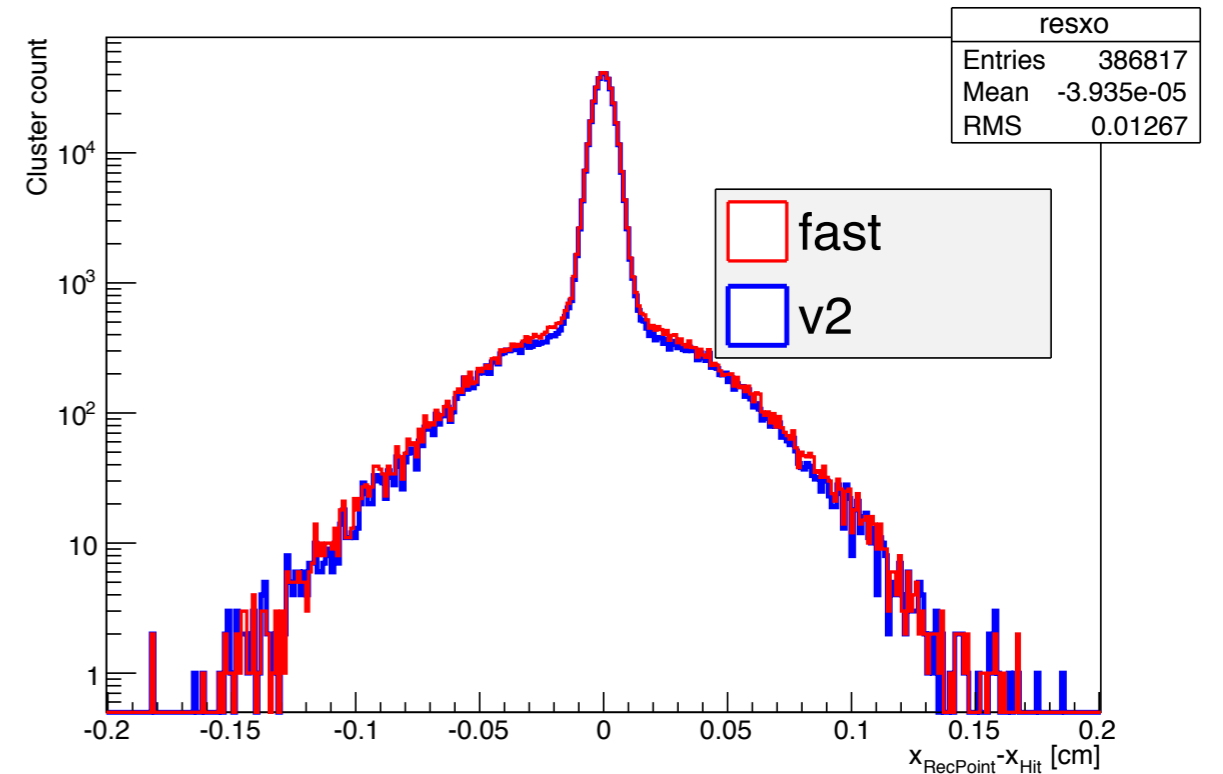
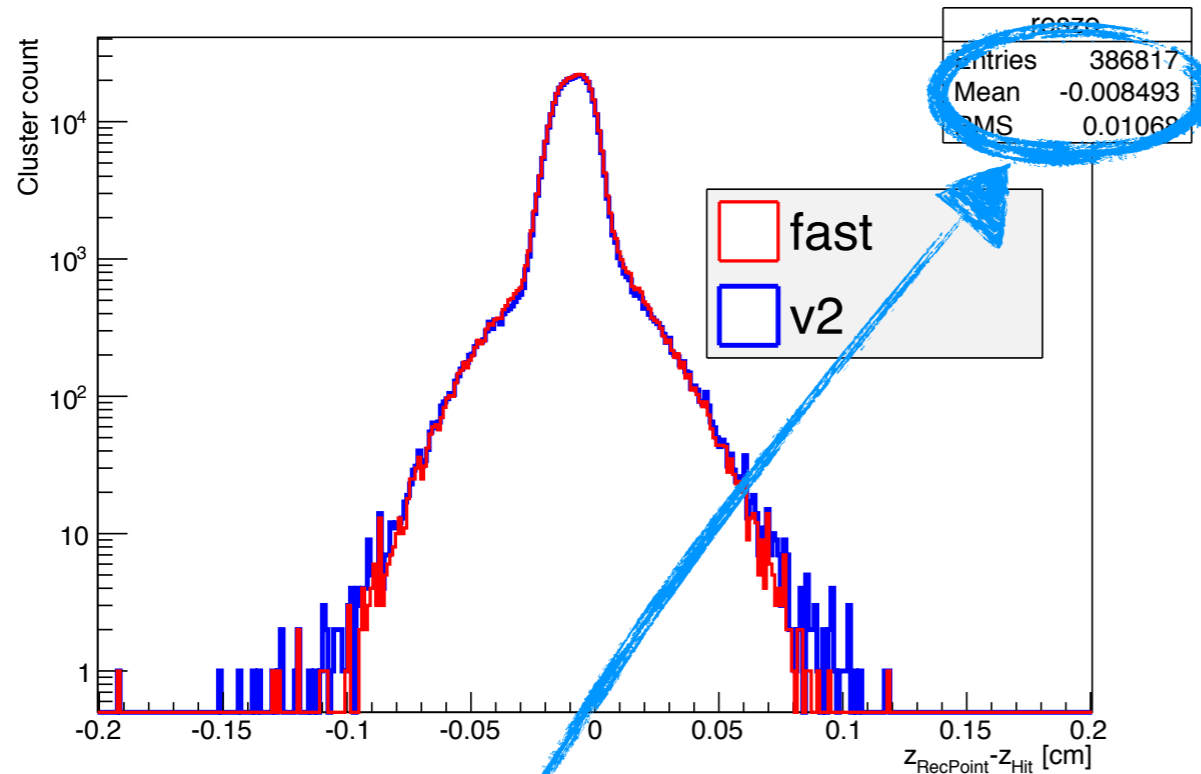
Data set: 18 central MC events



○ Same distribution with both CF

Pb-Pb MonteCarlo

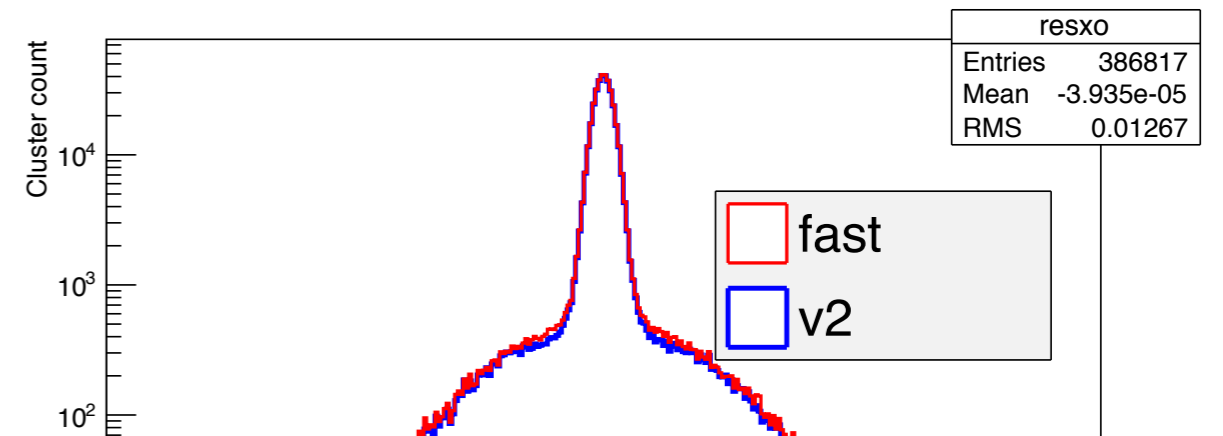
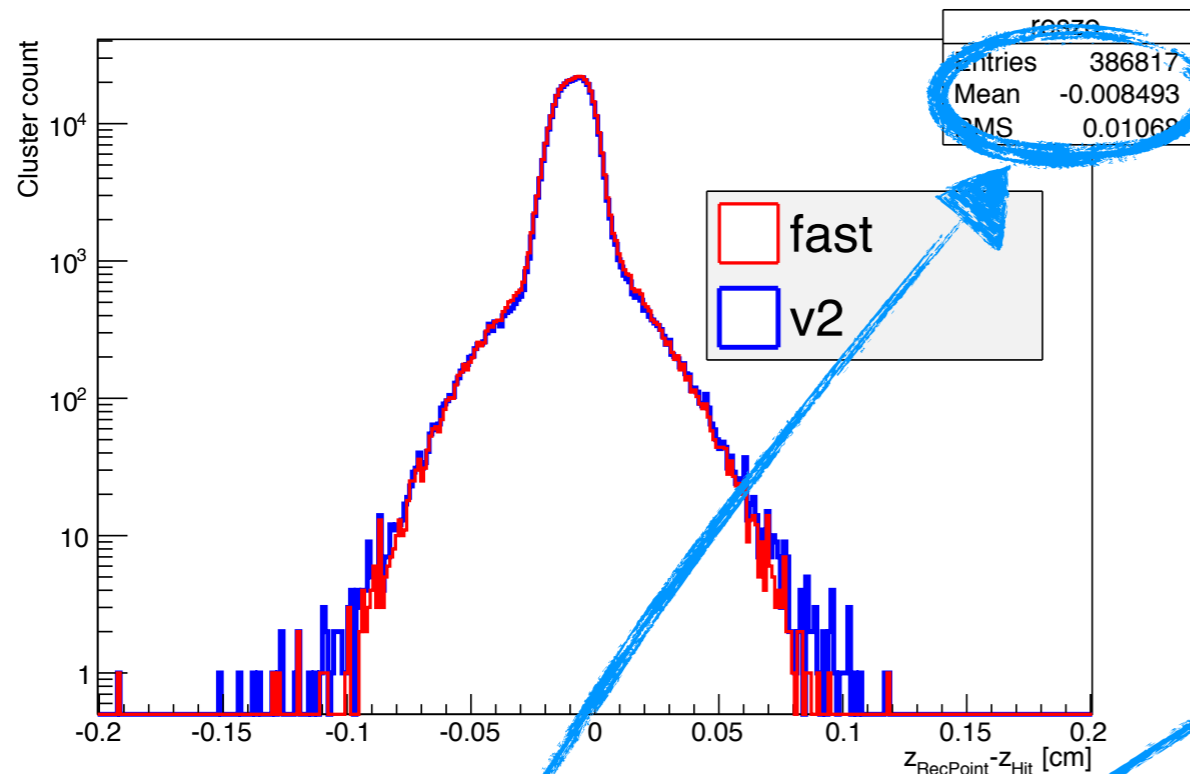
Data set: 18 central MC events



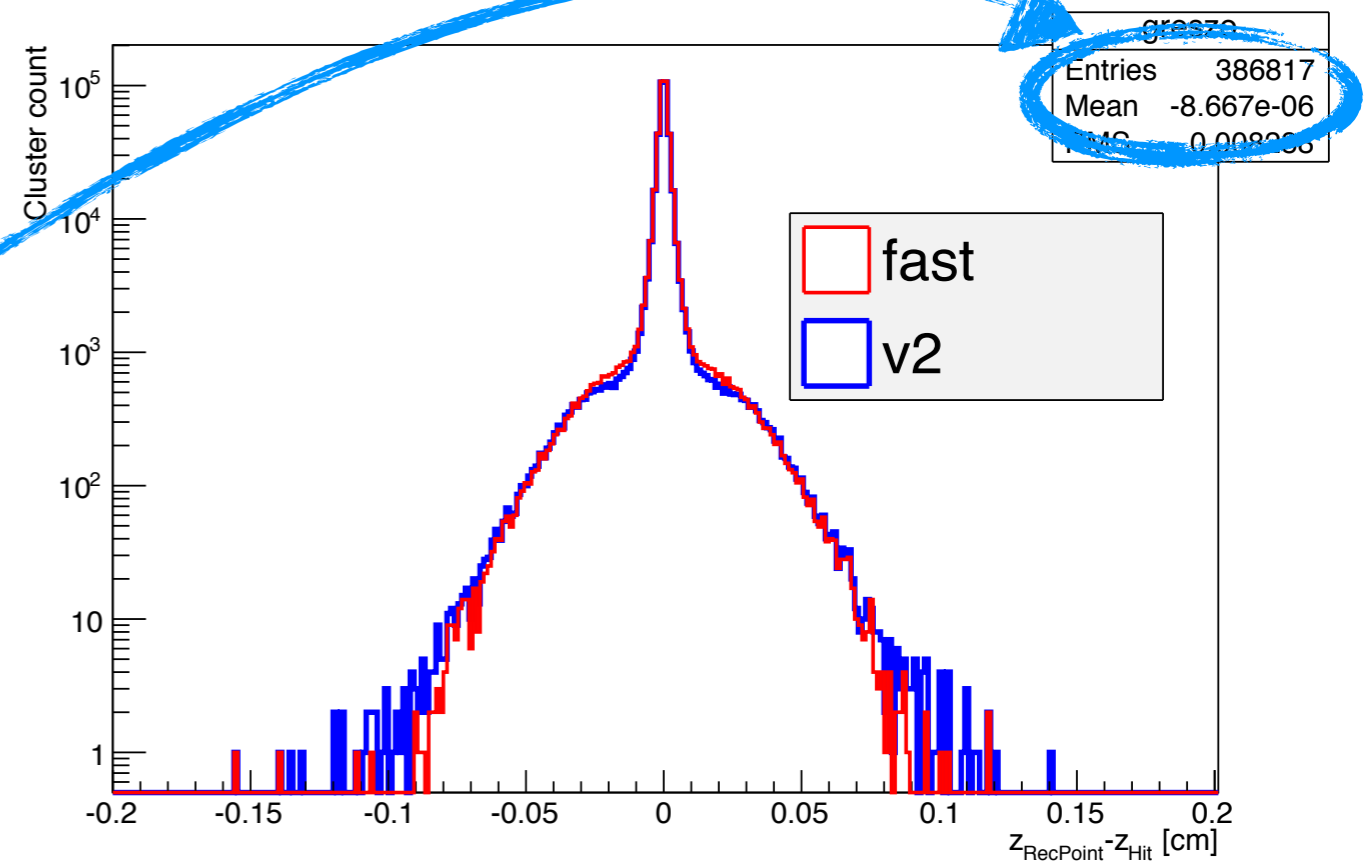
- Same distribution with both CF
- $\sim 85 \mu\text{m}$ bias on local Z

Pb-Pb MonteCarlo

Data set: 18 central MC events

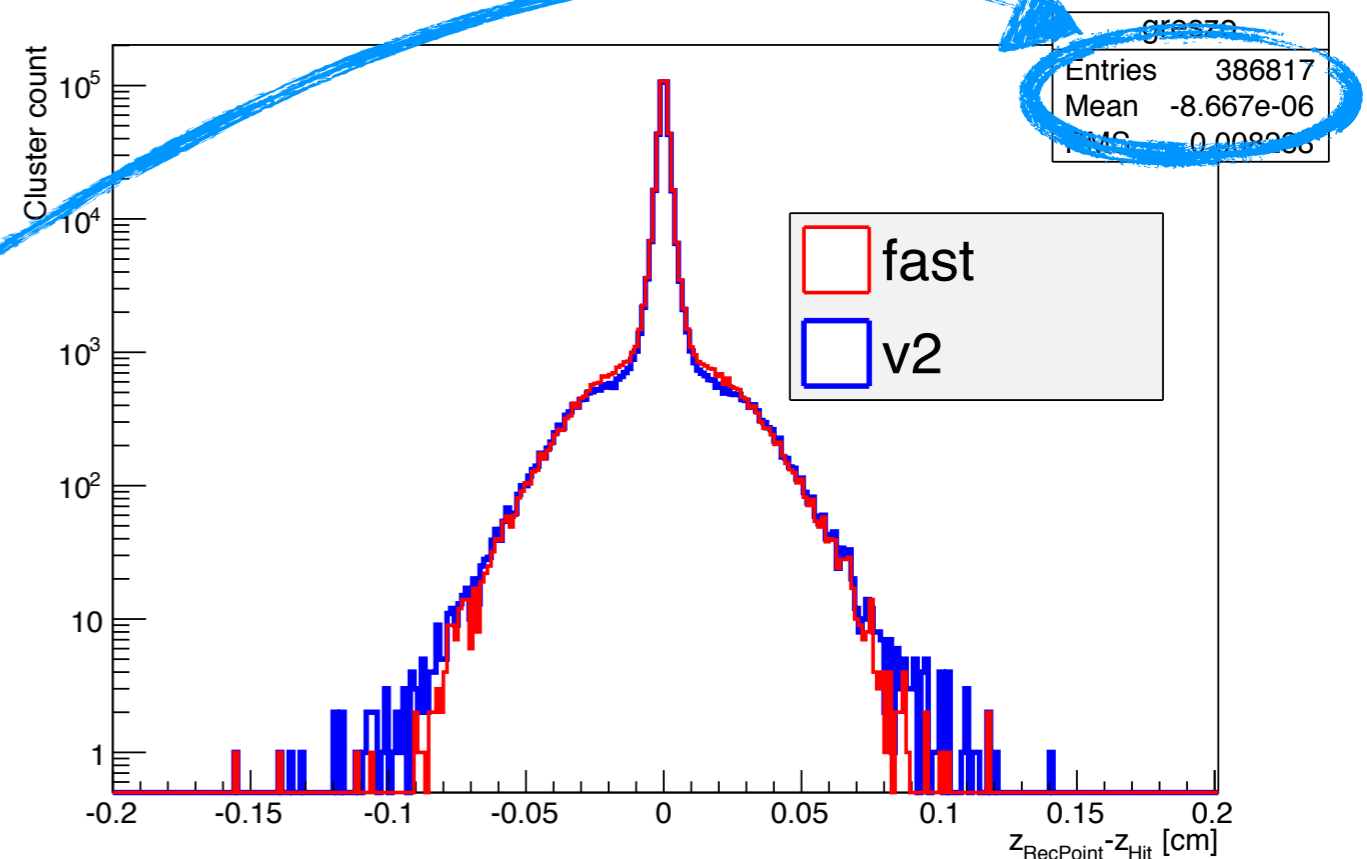
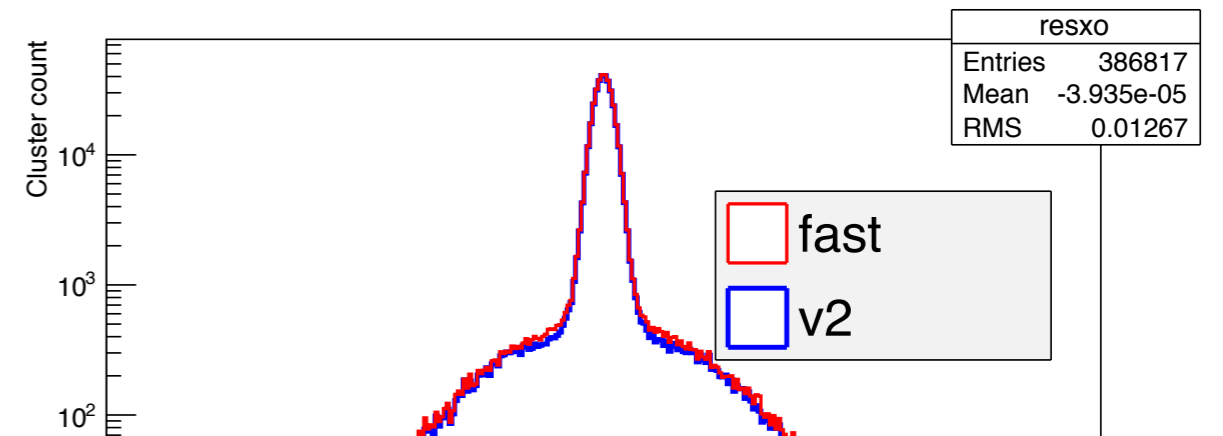
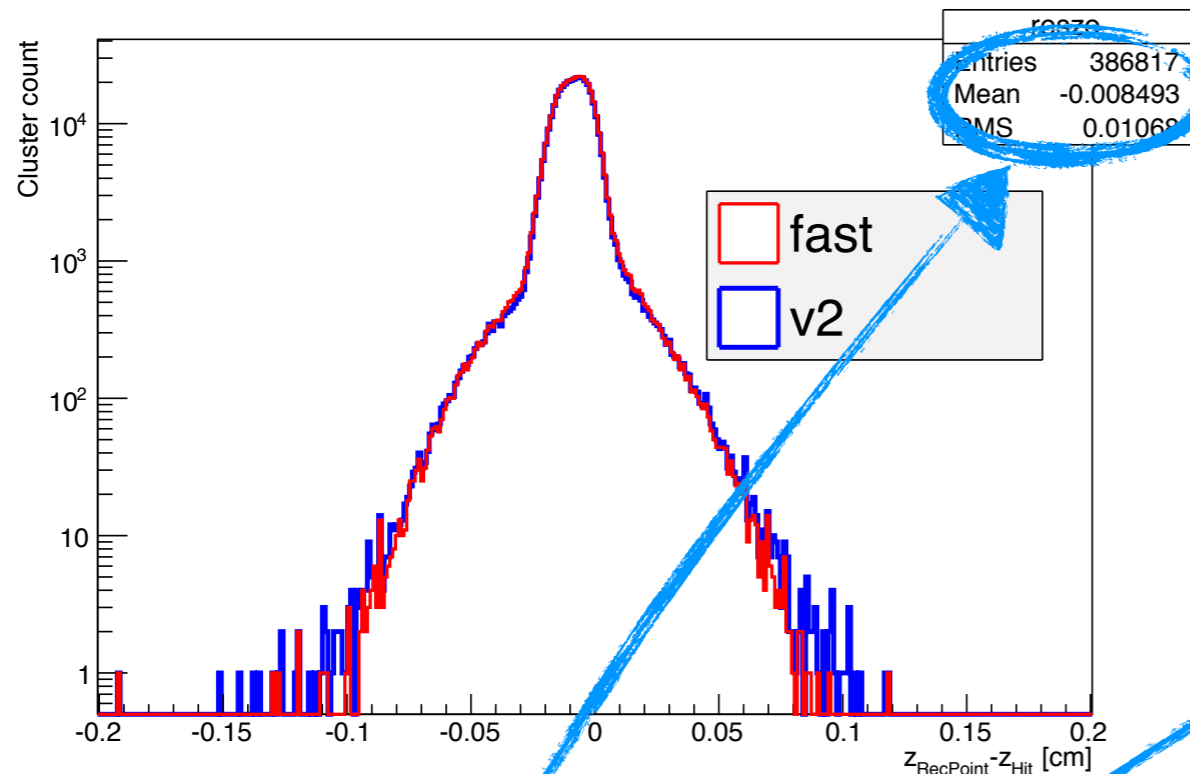


- Same distribution with both CF
- $\sim 85 \mu\text{m}$ bias on local Z
- No apparent bias on global Z



Pb-Pb MonteCarlo

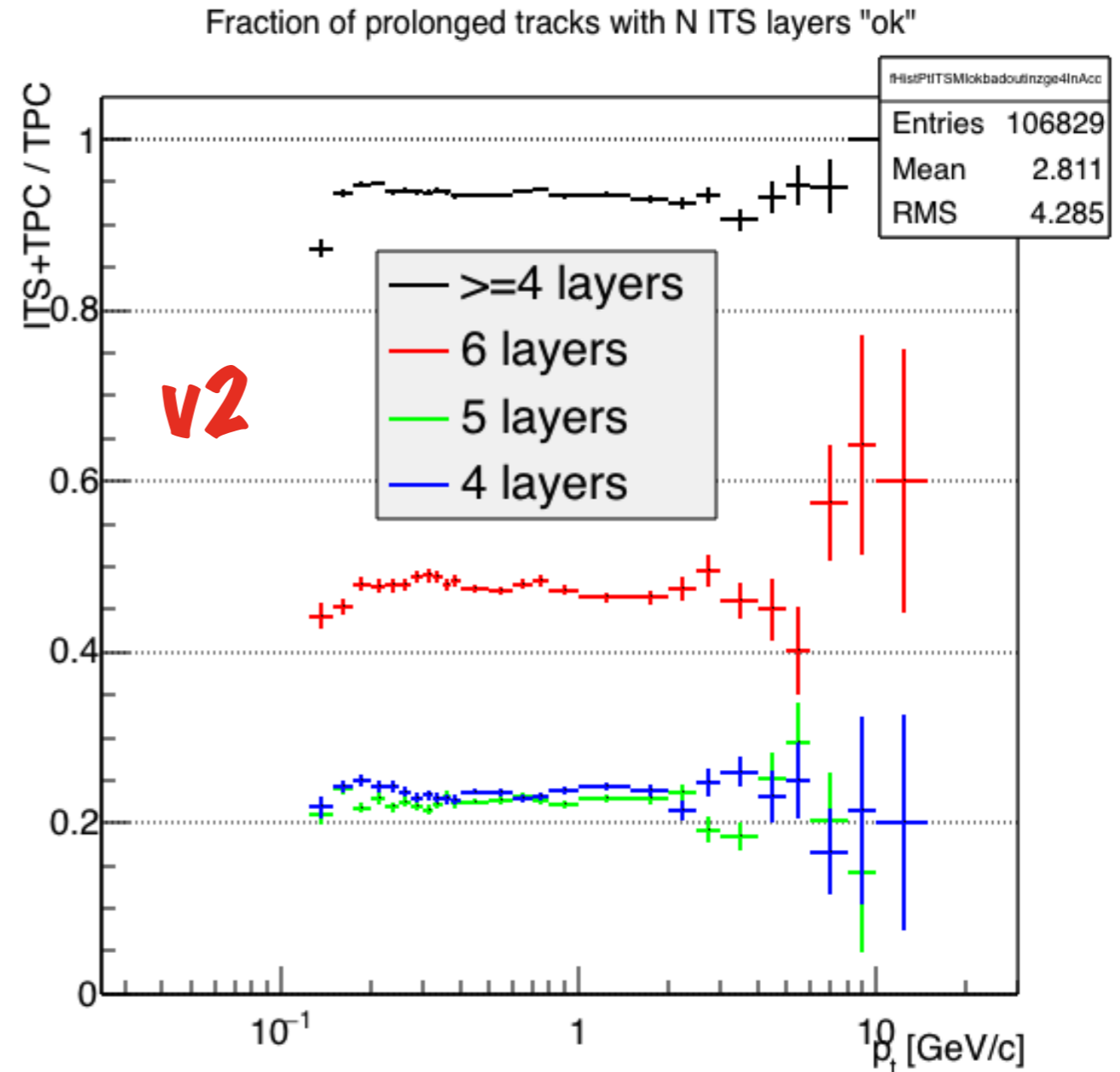
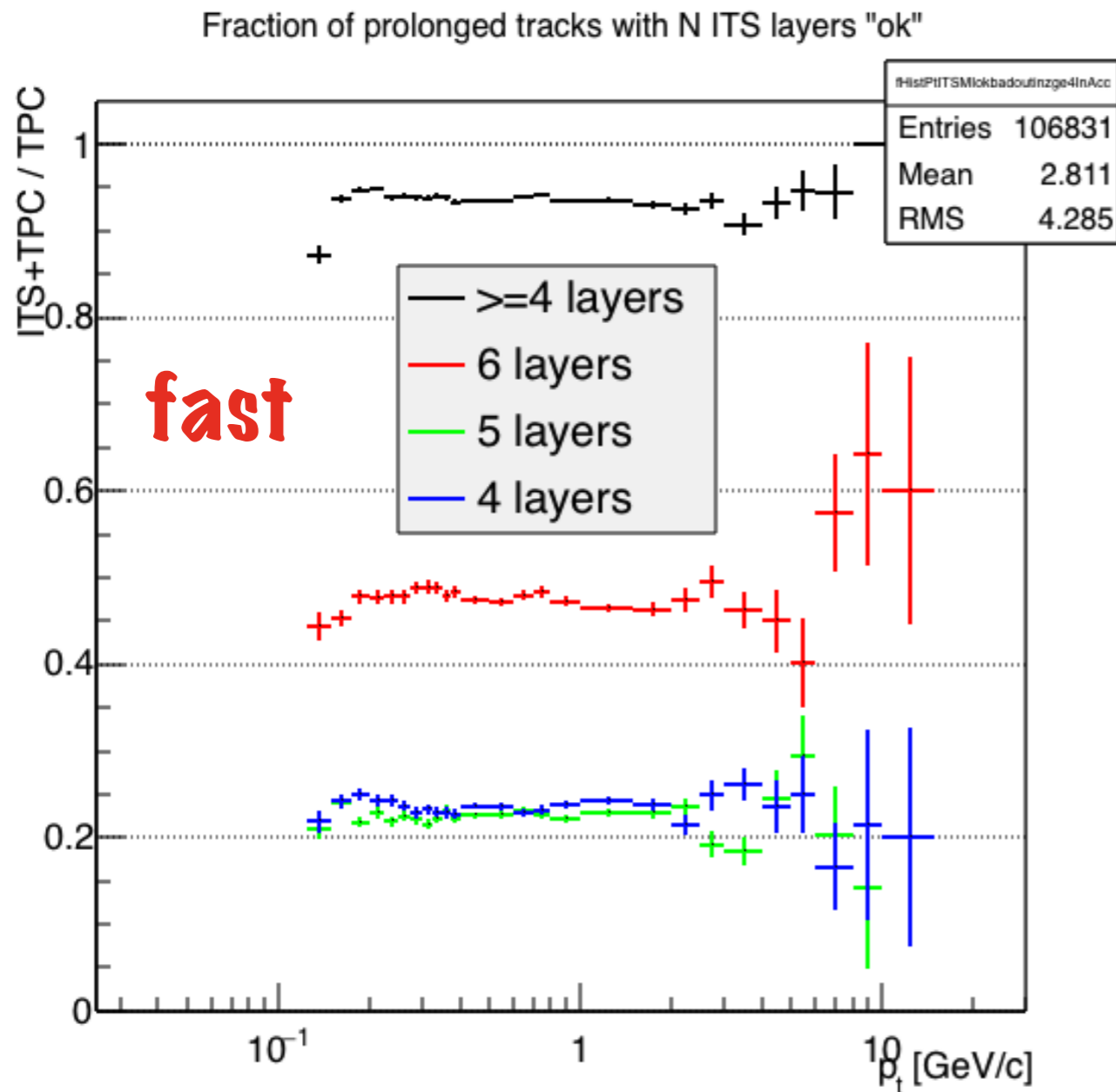
Data set: 18 central MC events



- Same distribution with both CF
- $\sim 85 \mu\text{m}$ bias on local Z
- No apparent bias on global Z
- To be understood...

Pb-Pb MonteCarlo

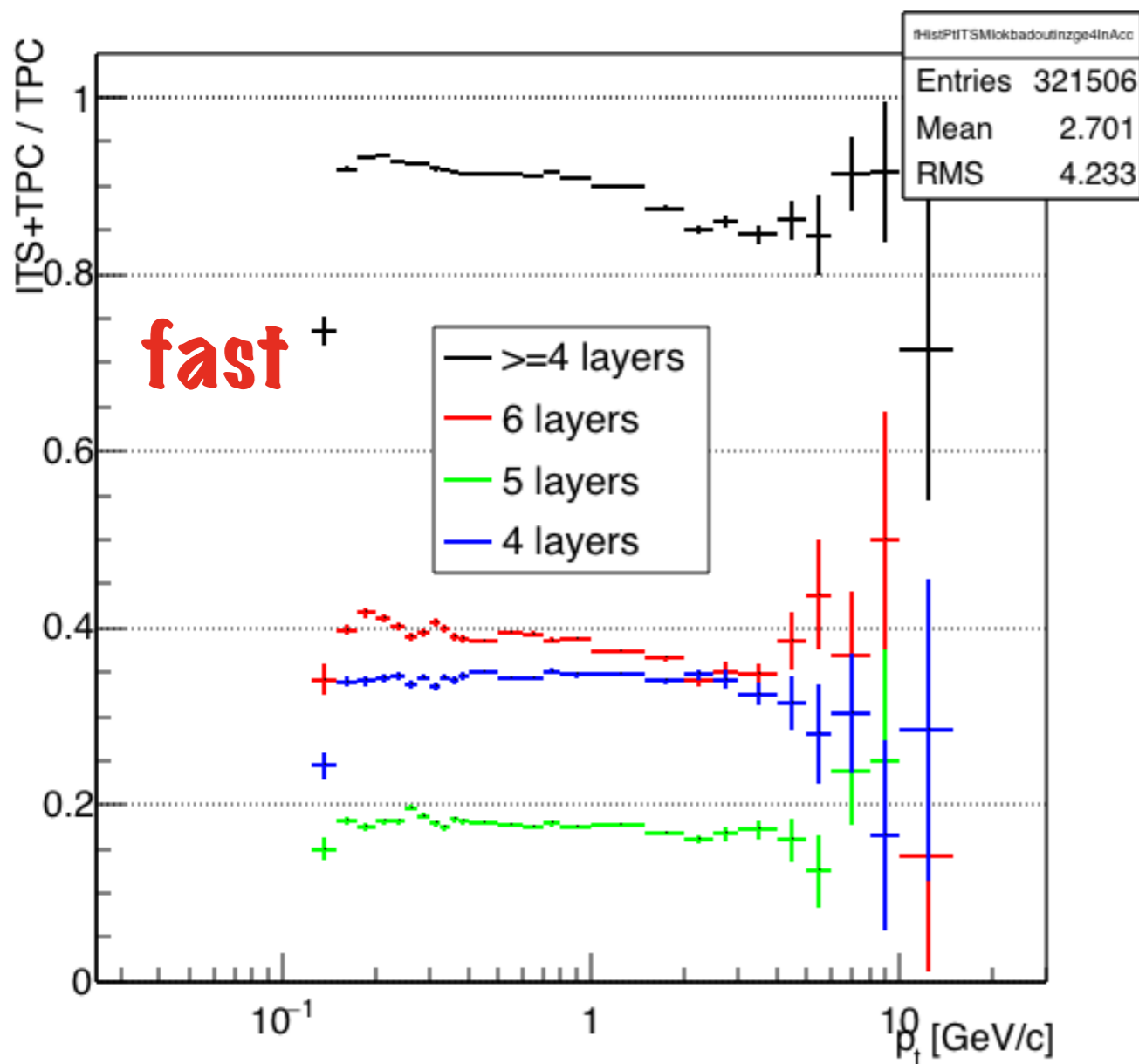
Data set: 92 central MC events



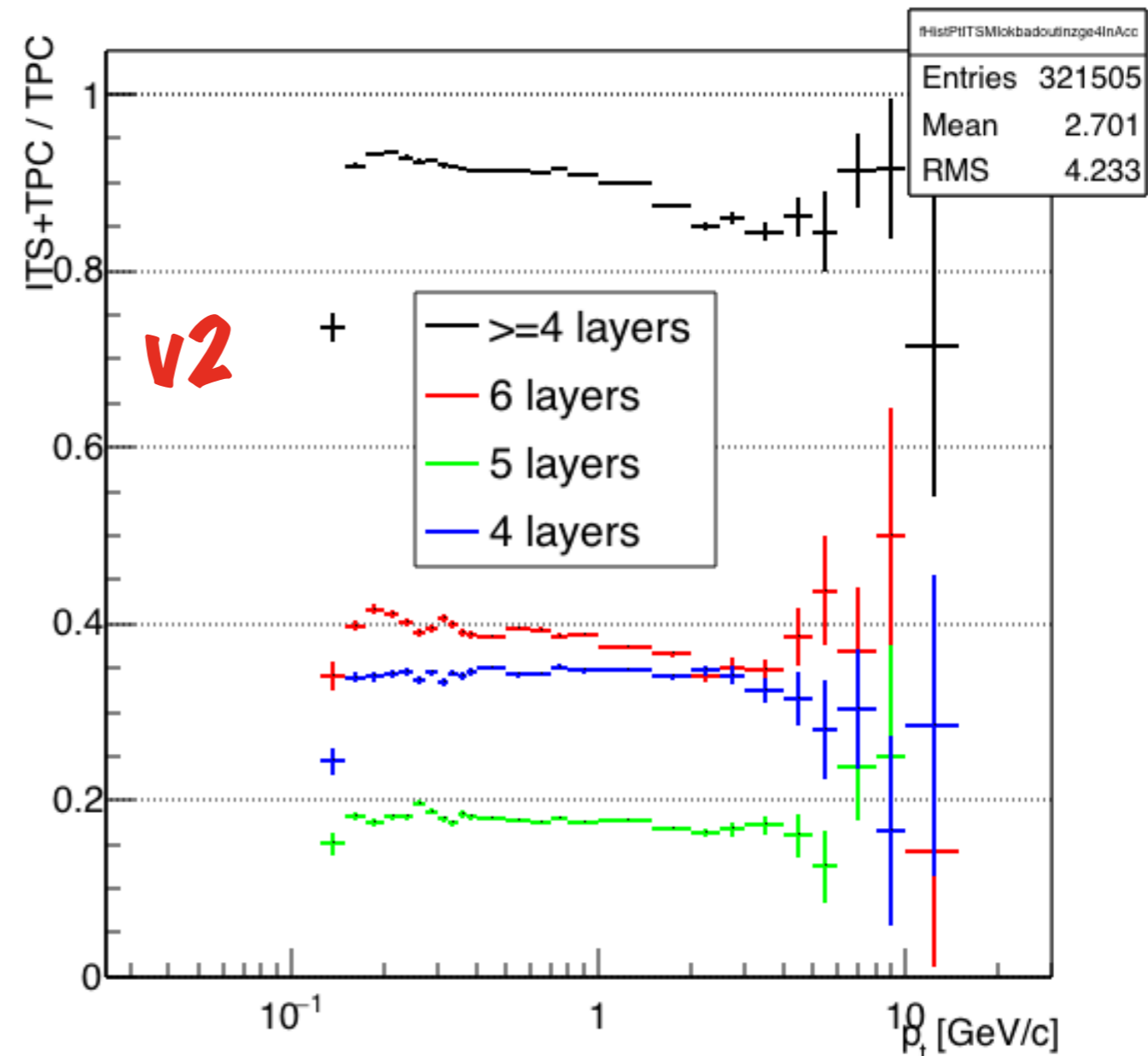
Pb-Pb data

Data set: 4 chunk from LHC11h
(~800 events)

Fraction of prolonged tracks with N ITS layers "ok"

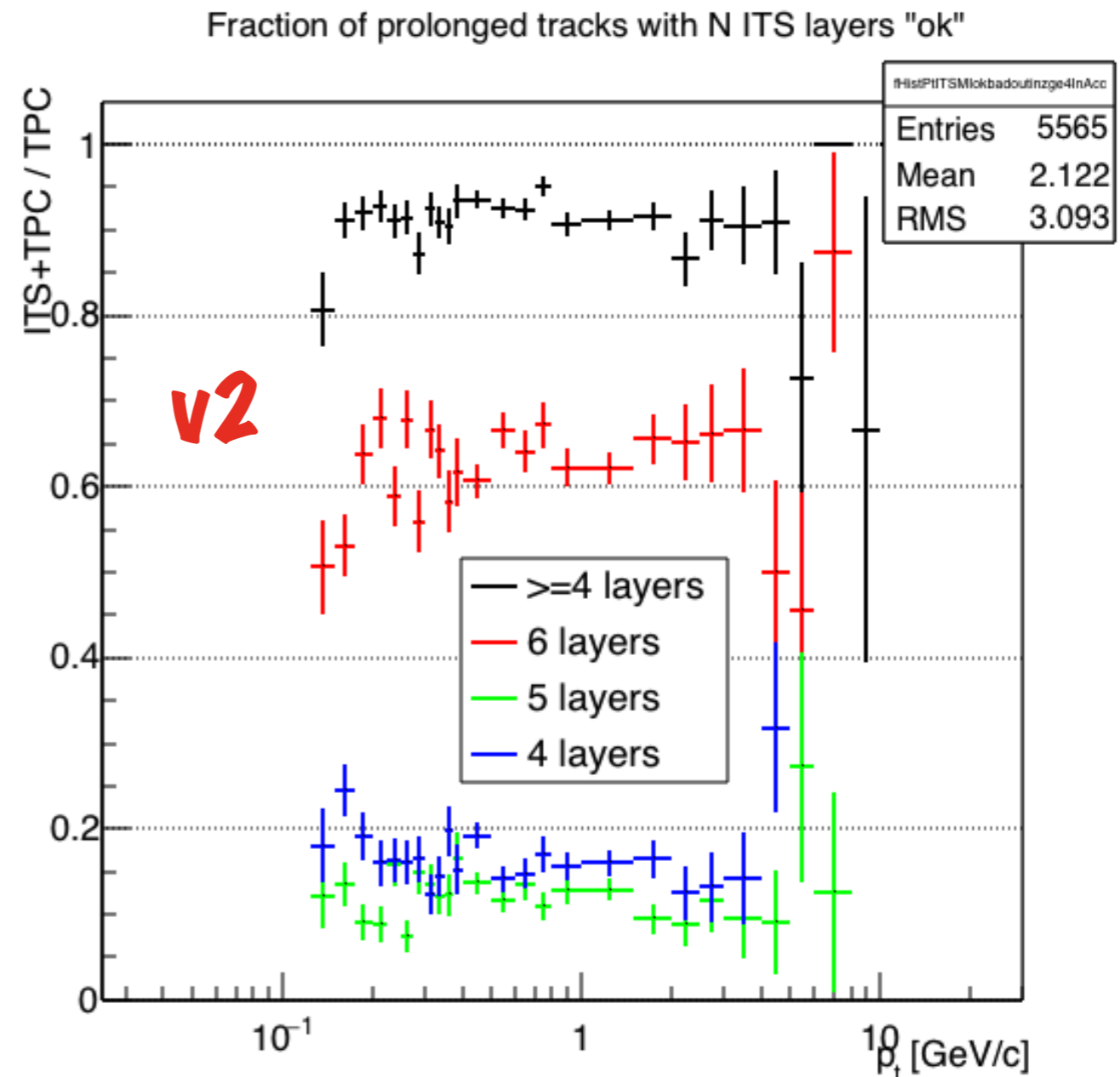
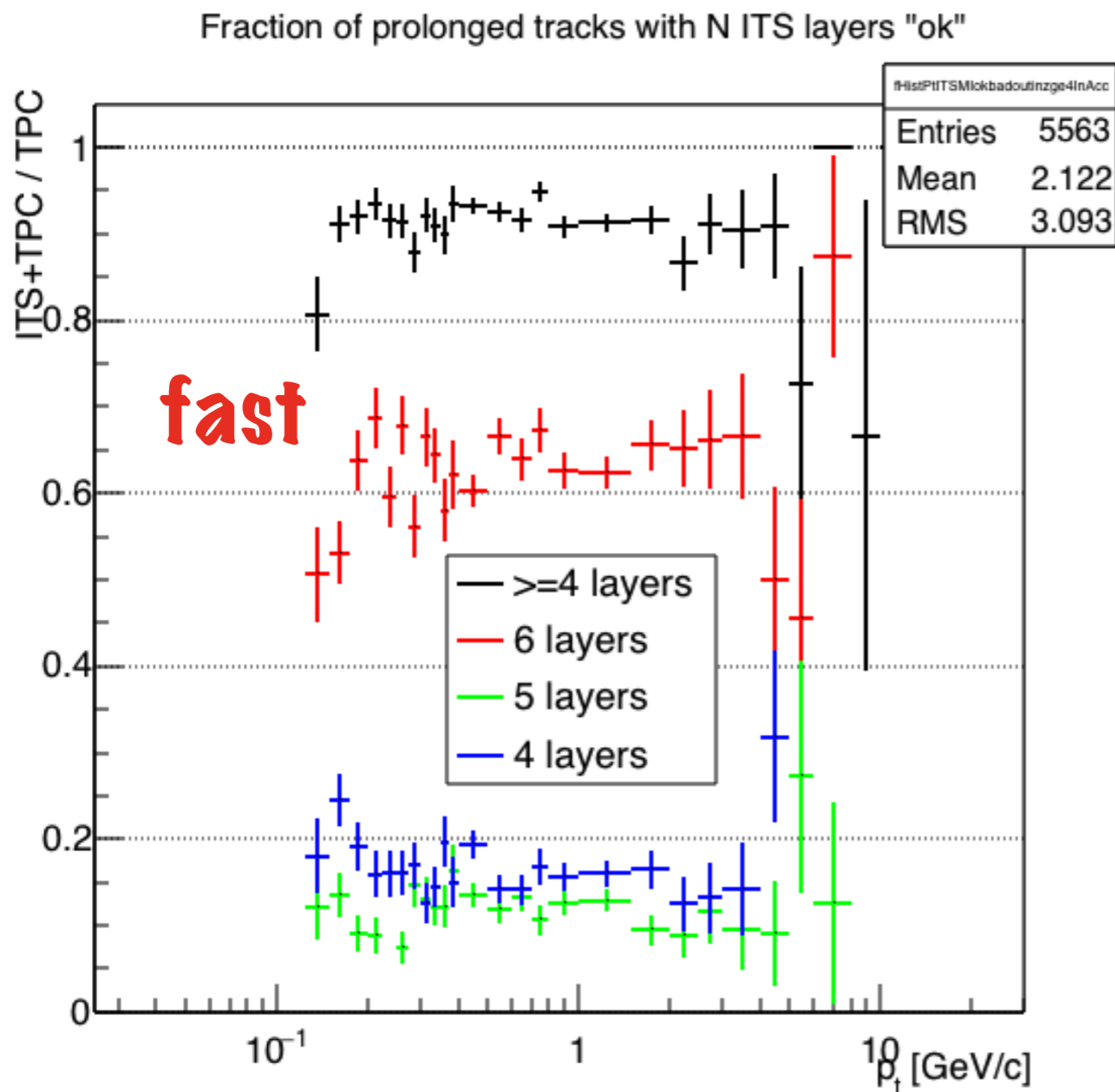


Fraction of prolonged tracks with N ITS layers "ok"



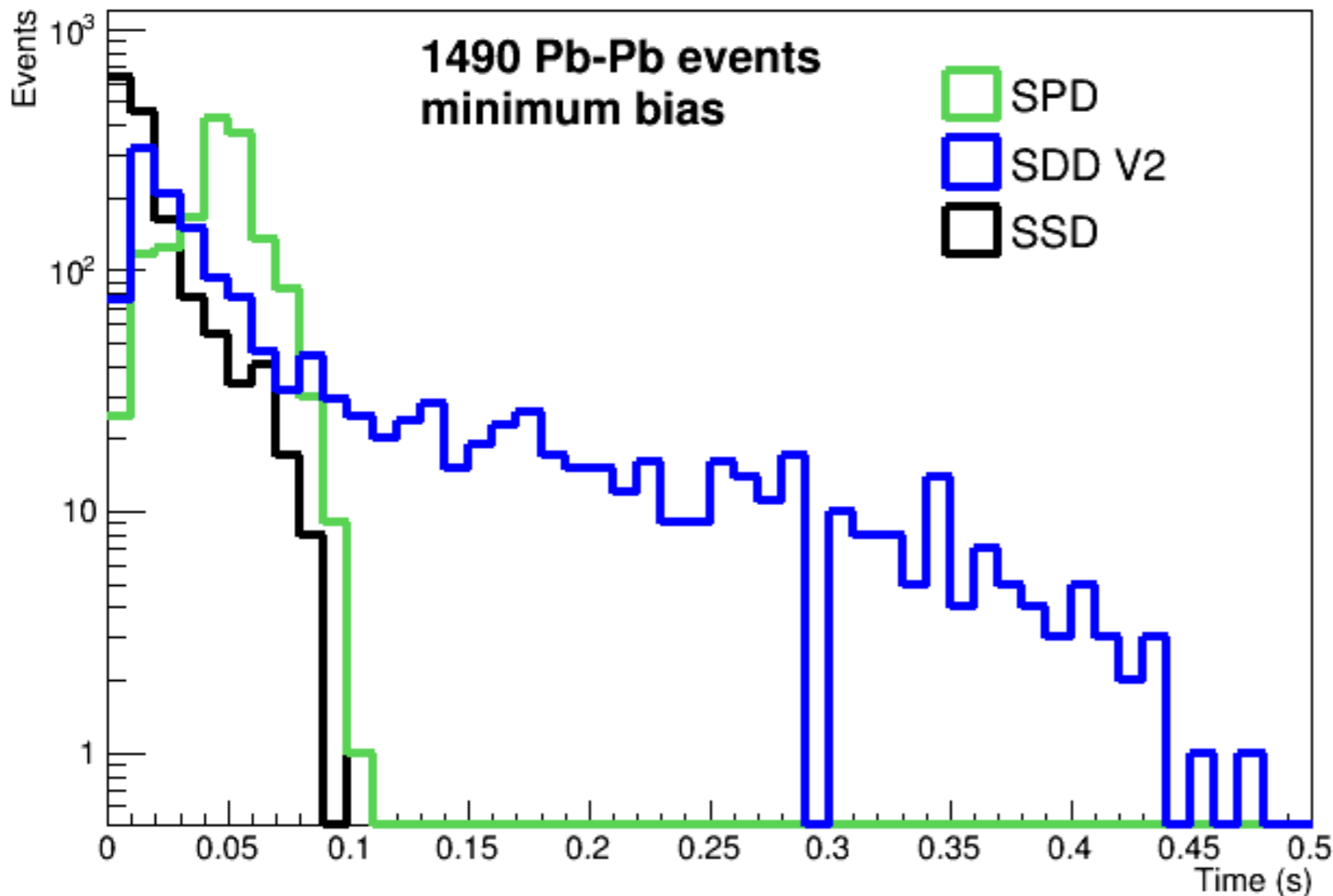
p-p data

Data set: 1 chunk from LHC12g
(~1200 events)



Computing performance

Execution time

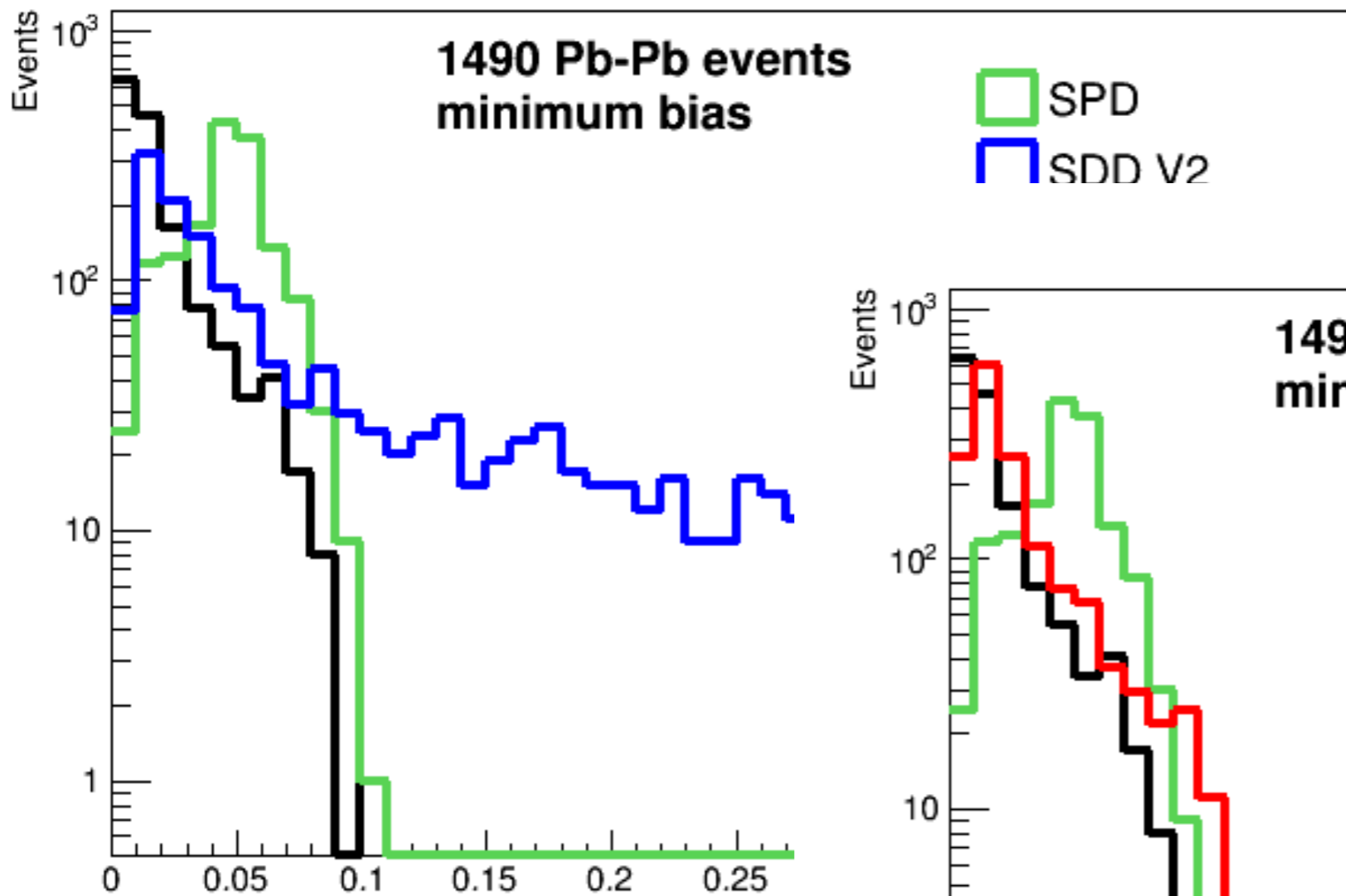


Long queue only
for SDD

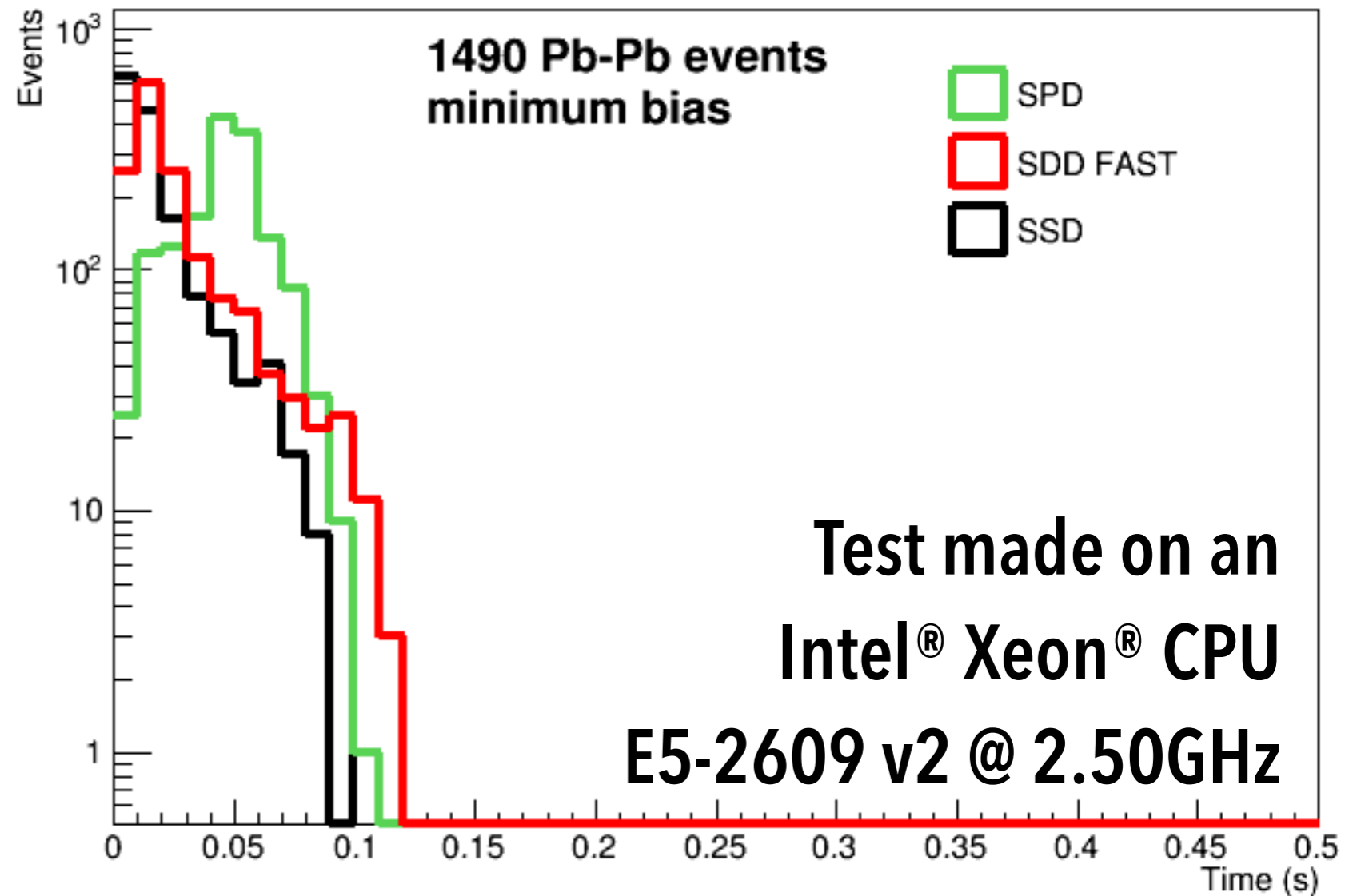
Test made on an
Intel® Xeon® CPU
E5-2609 v2 @ 2.50GHz

Execution time

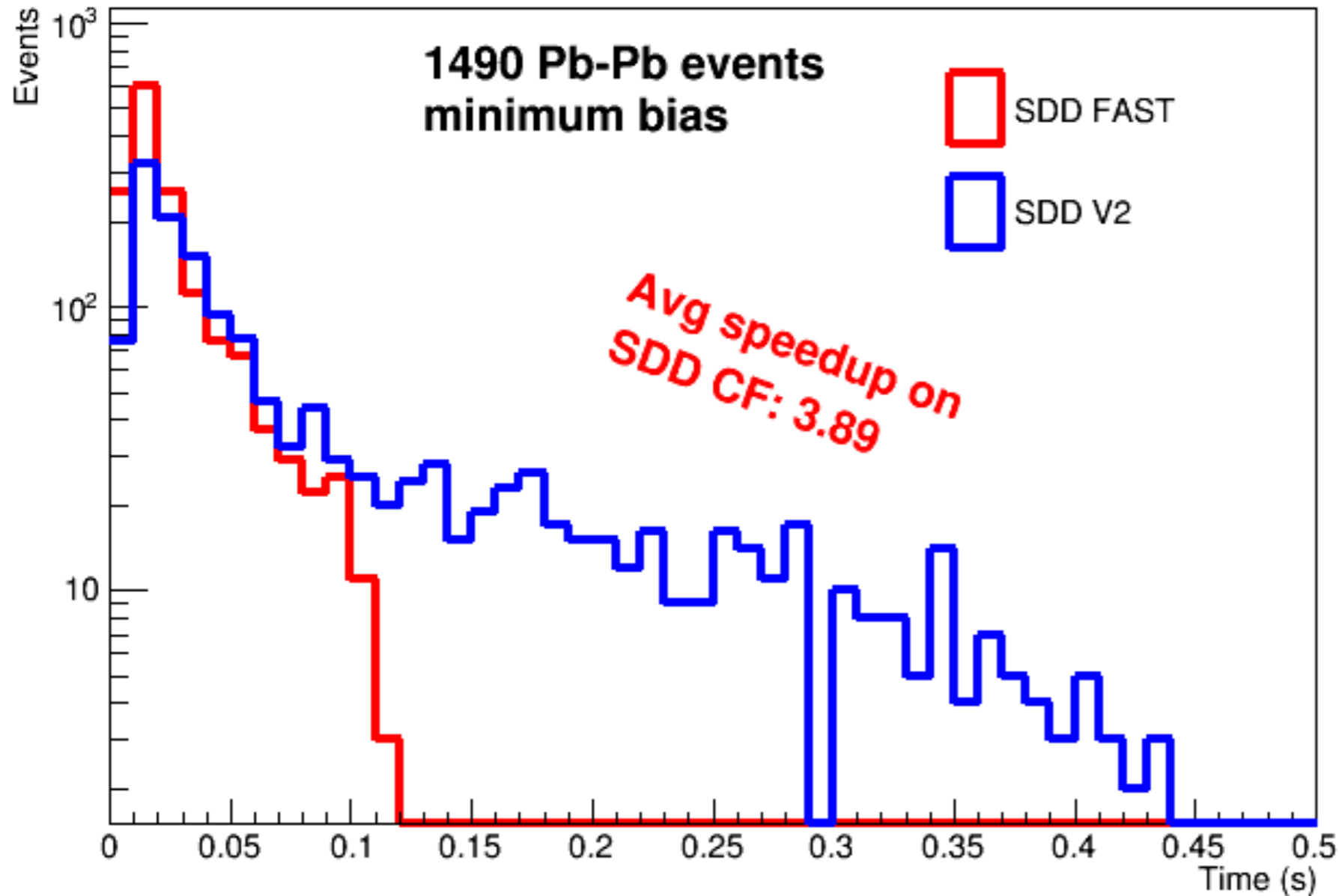
FAST algo is
comparable with others



Long queue only
for SDD

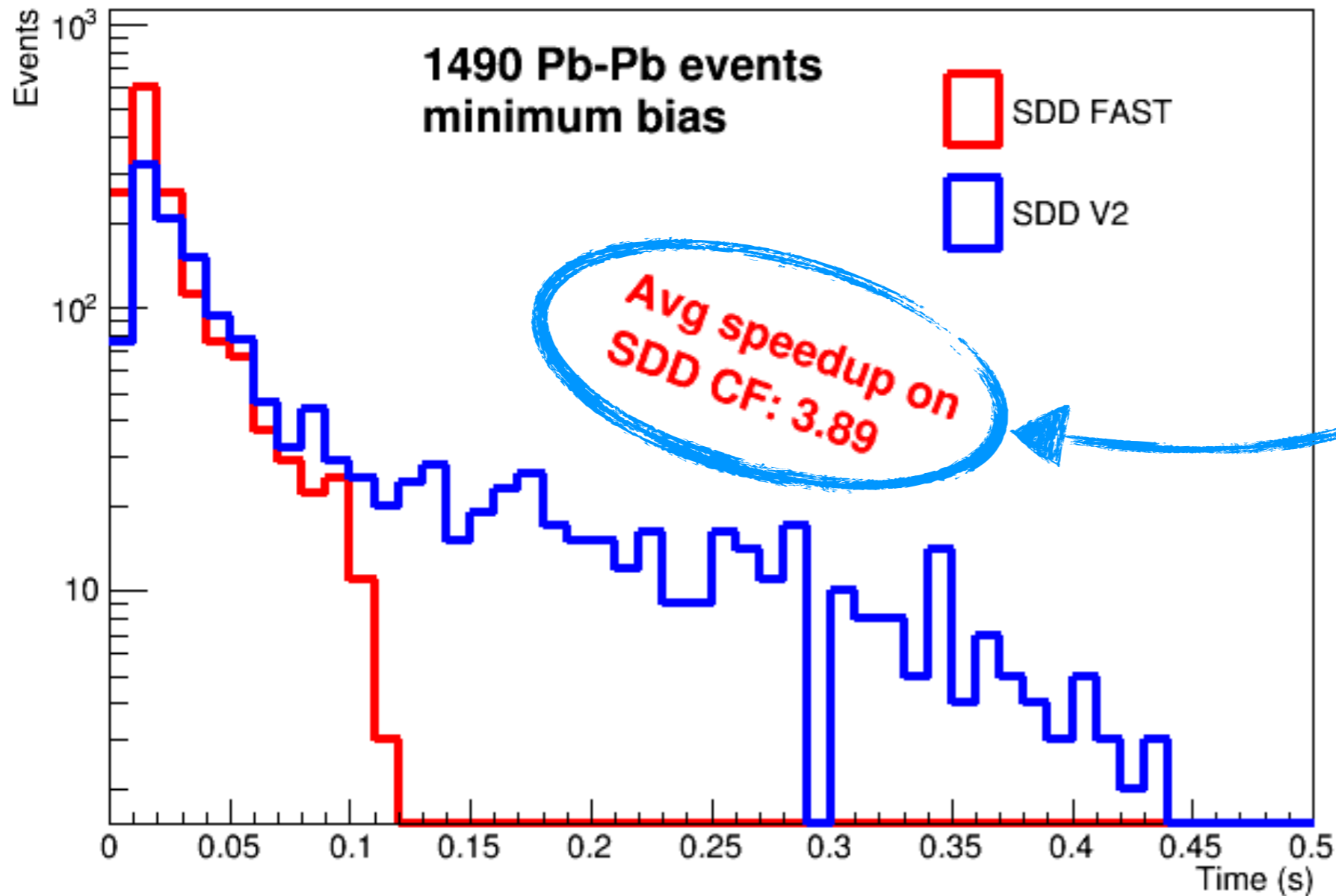


Speedup



Test made on an
Intel® Xeon® CPU
E5-2609 v2 @ 2.50GHz

Speedup



Fast algo is in the HLT reconstruction now

Test made on an
Intel® Xeon® CPU
E5-2609 v2 @ 2.50GHz

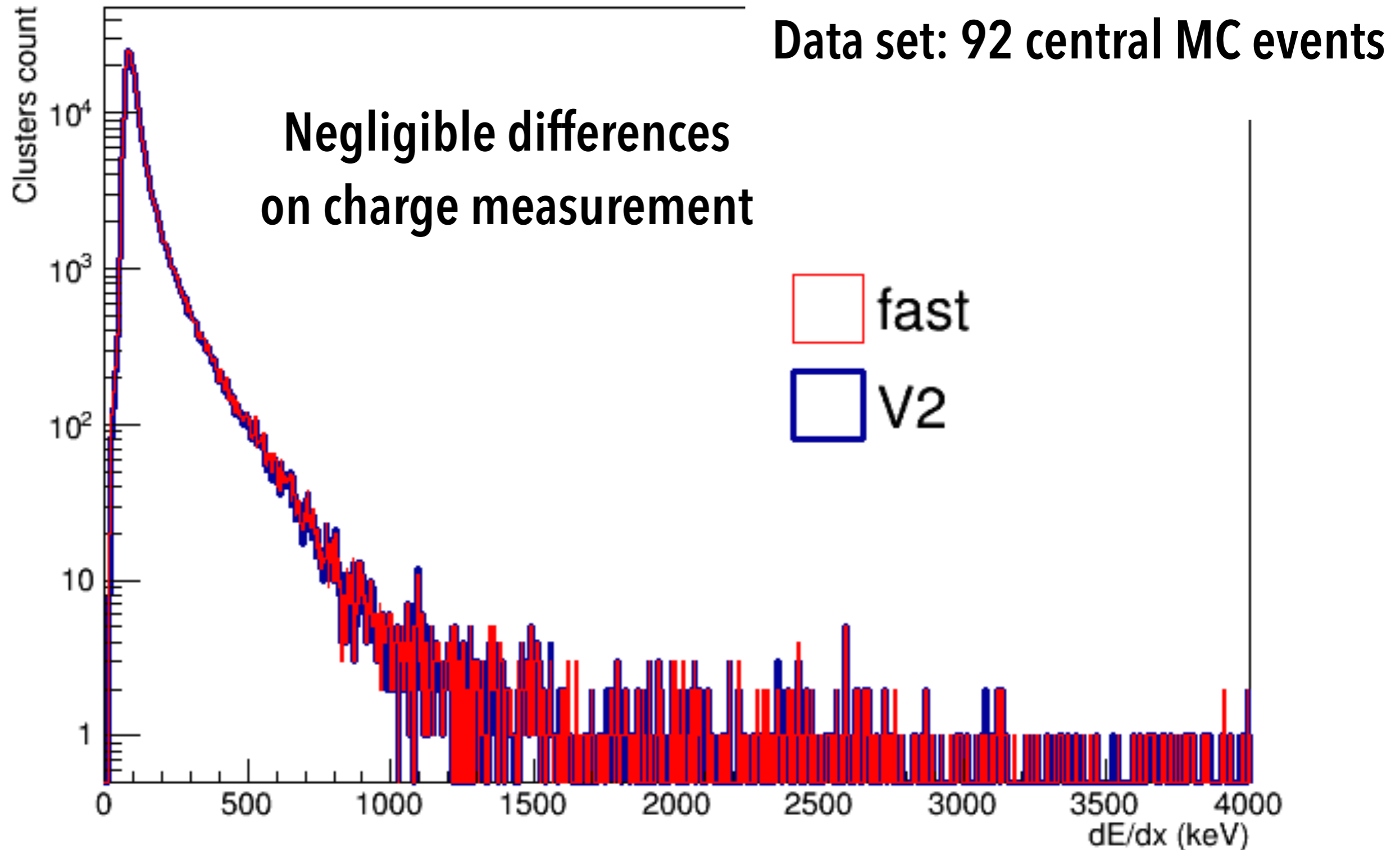
Summary

Summary

- **Committed a new class in AliRoot**
`$ALICE_ROOT/ITS/AliITSClusterFinderSDDfast.*`
- **Reconstruction performance is the same of the v2 CF**
- **The speedup satisfies HLT requirements**
- **From 2014/09/10 the fast CF is the default for SDD offline reconstruction (commit b4bd769 on master branch)**
- **Not ported to the current release, it will go to the next one**
- **Activated this morning in the HLT reconstruction (commit 8700980 on master branch)**

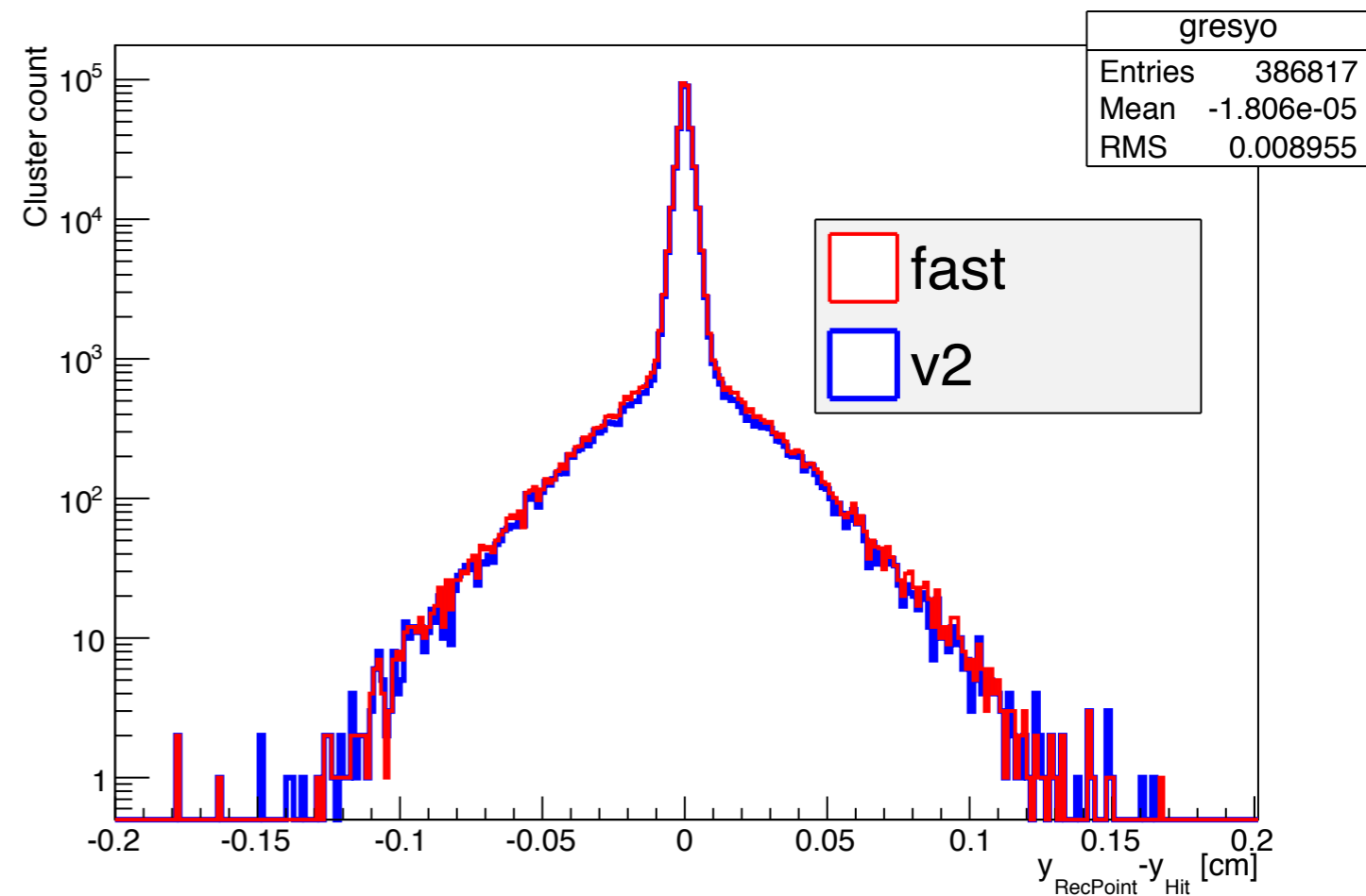
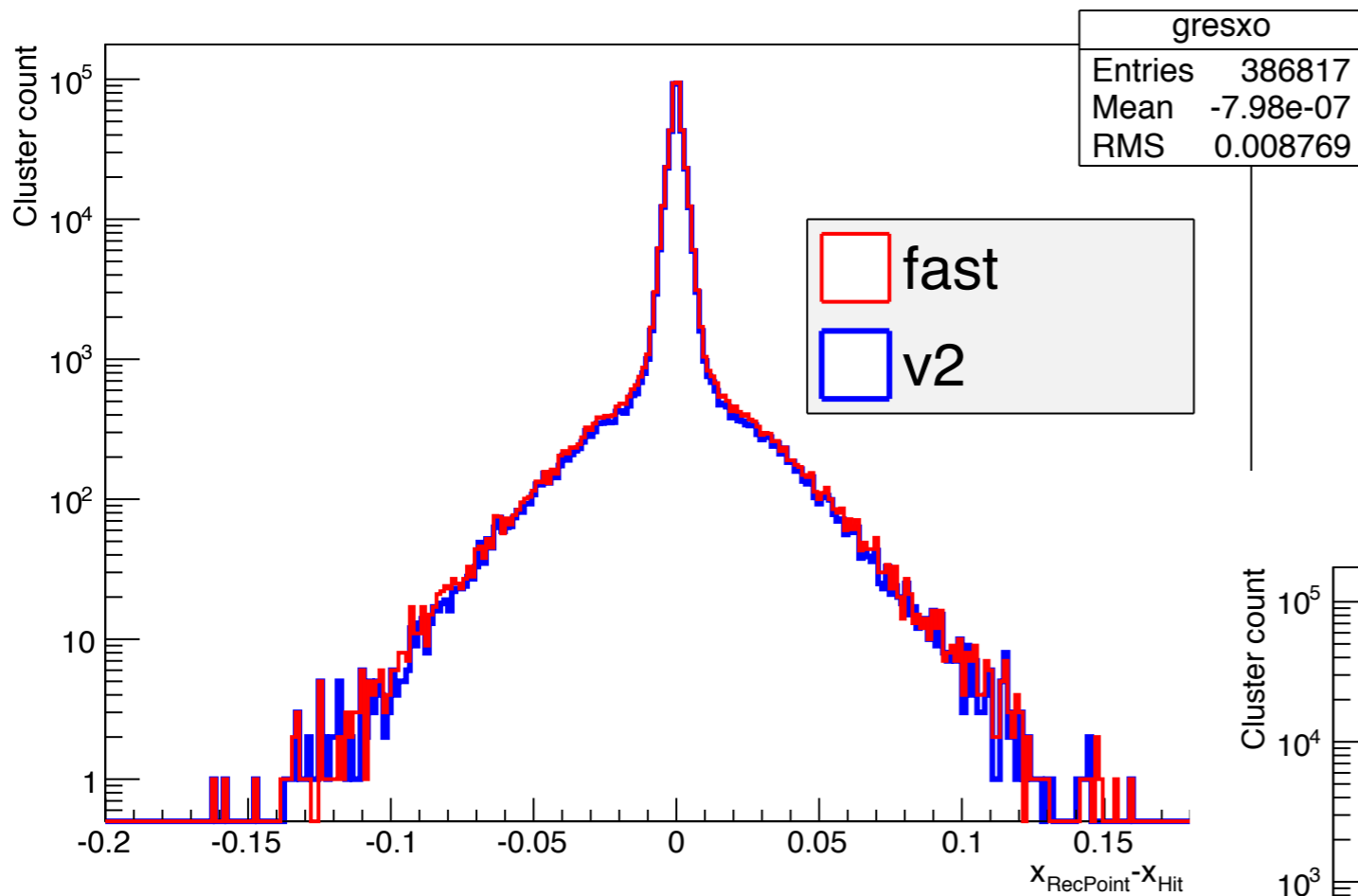
Backup

Pb-Pb MonteCarlo

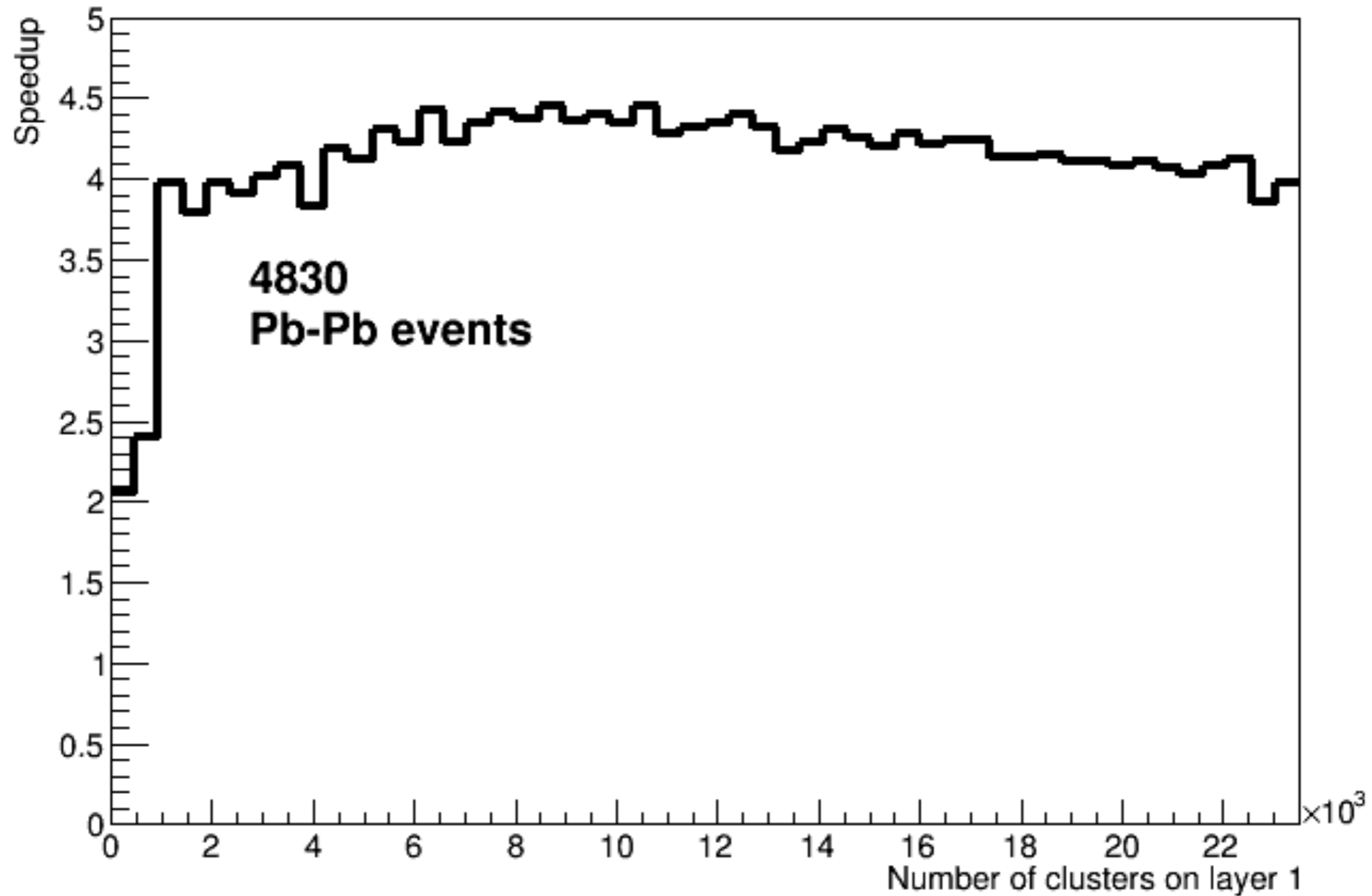


Pb-Pb MonteCarlo

Data set: 18 central MC events



Speedup Vs. Multiplicity



None of the two algorithms is occupancy-dependent