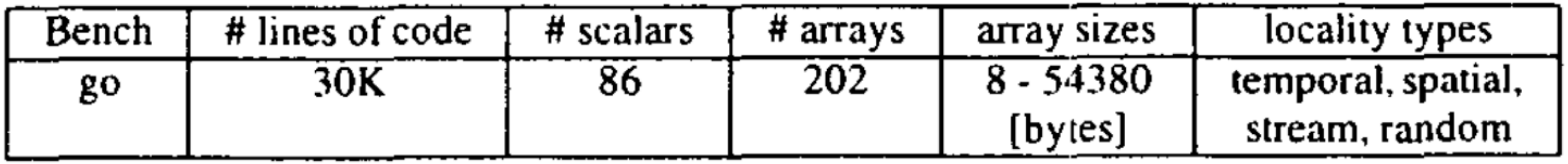
A single case

Design a traditional cache for GO benchmark

- Setup: 8KB 2way set associative cache (cache line size configurable)
- Benchmark: Go game playing



Bench	# lines of code	# scalars	# arrays	array sizes	locality types
go	30K	86	202	8 - 54380 [bytes]	temporal, spatial, stream, random

- design target: achieve a good hit rate and lower bandwidth;
- Design independent variable: cache line size.

A single case

Design a traditional cache for GO benchmark

- design target - achieve a good hit rate and lower bandwidth
- The optimal design: 8 words line size
- Total memory Bandwidth needed: 0.32B/cycle

A strong rule:
Lager the cache line size is,
higher the memory bandwidth the cache will ask for.
Because you need to feed the cache
with a larger window of data every time
(When fill/replace/WB).

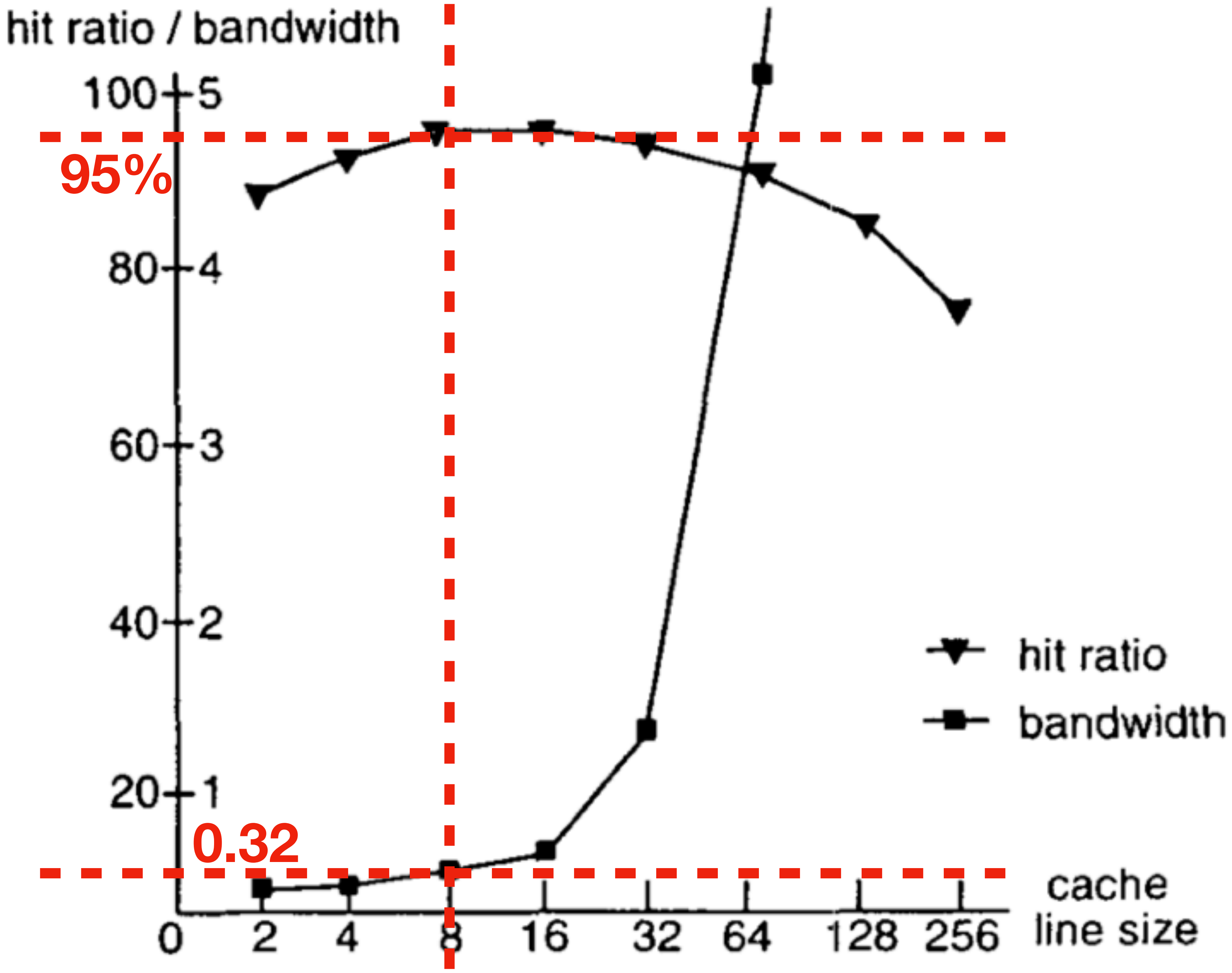


Figure 2. The cache hit ratio and memory bandwidth variation with the cache line size for the go benchmark