# This paper's focus
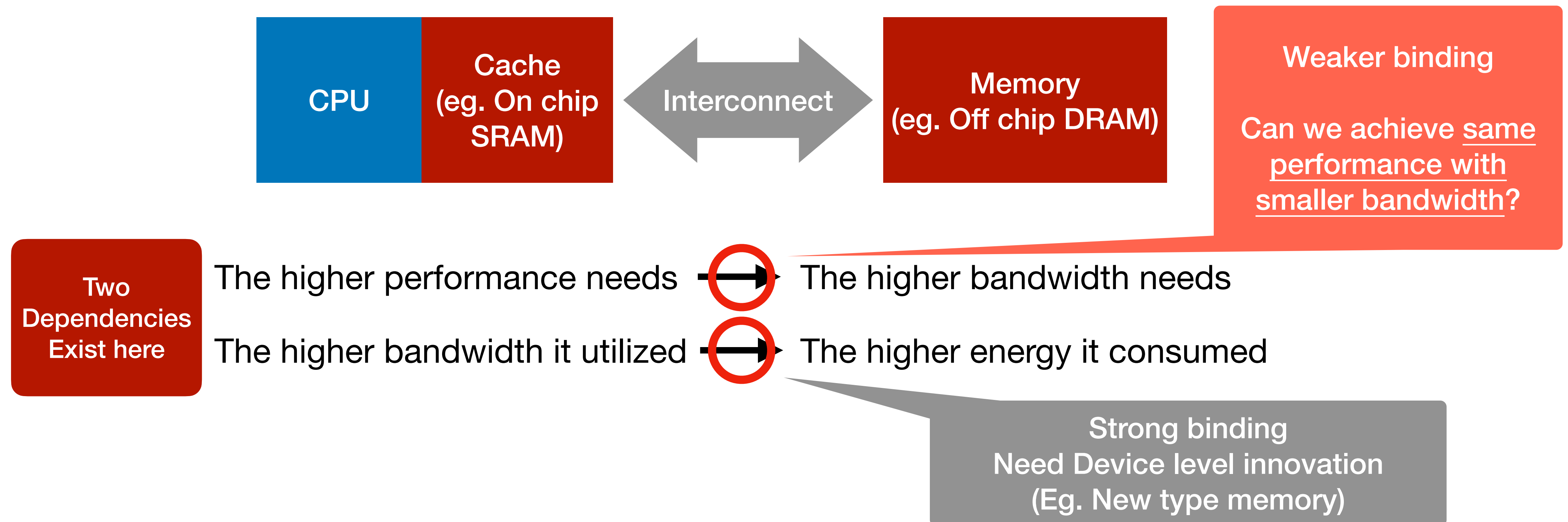## A more specific problem this paper works on
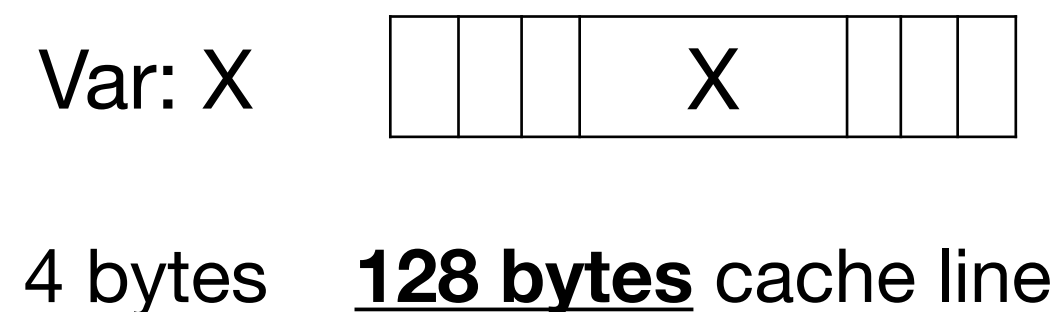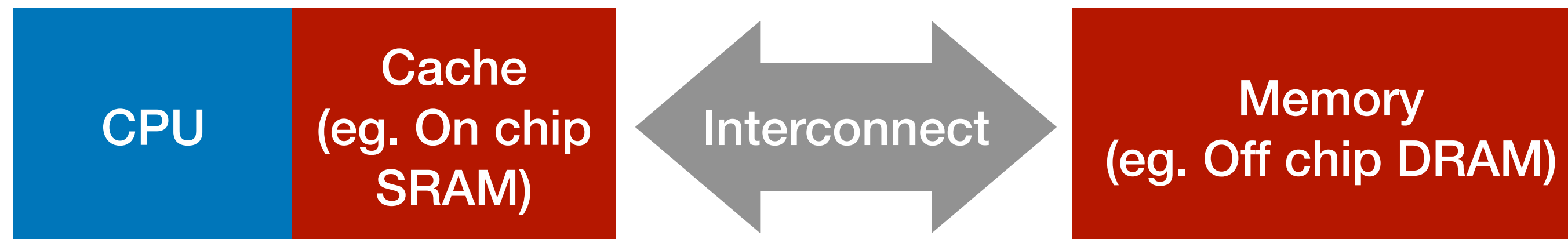
- A macro view

| CPU | Cache (eg. On chip SRAM) | ⟷ Interconnect | Memory (eg. Off chip DRAM) |

**Weaker binding**

Can we achieve <u>same performance with smaller bandwidth?</u>

**Two Dependencies Exist here**

The higher performance needs ➔ The higher bandwidth needs

The higher bandwidth it utilized ➔ The higher energy it consumed

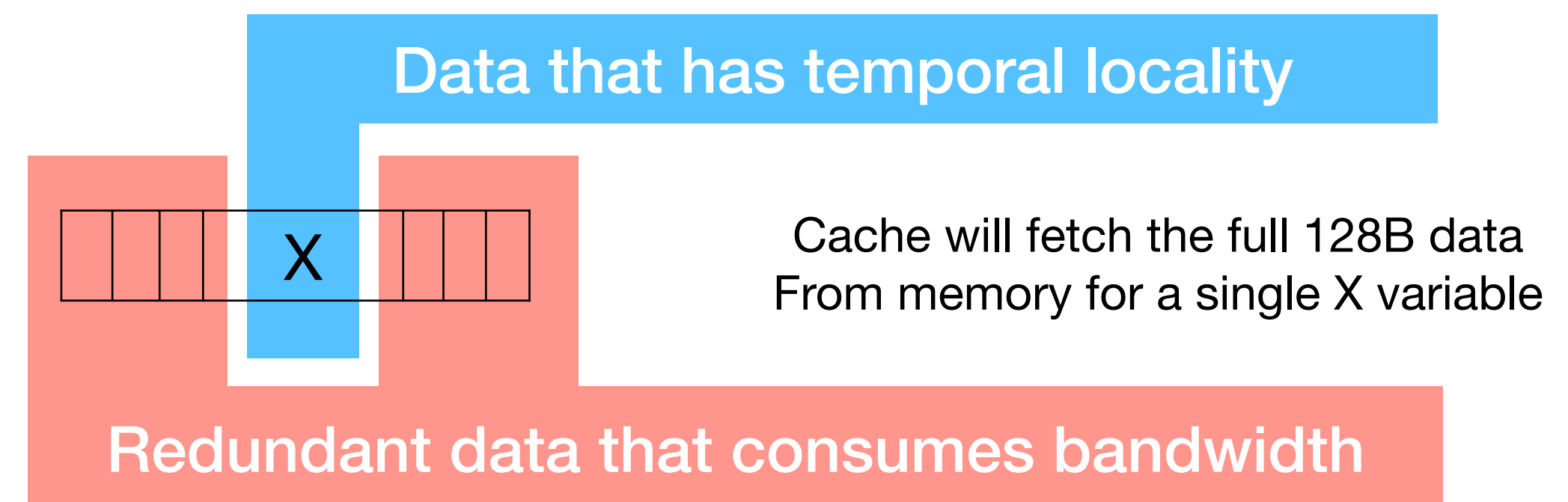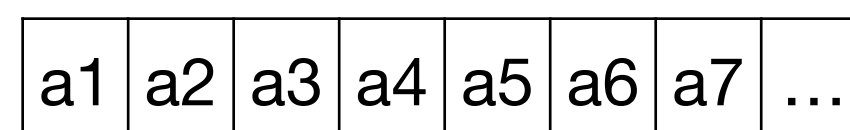**Strong binding**
Need Device level innovation
(Eg. New type memory)

# This paper's focus
## The breaking point - is there any WASTE of bandwidth?

- "Useless or redundant fetches which increase memory traffic"

CPU | Cache (eg. On chip SRAM) | Interconnect | Memory (eg. Off chip DRAM)

Var: X       X

4 bytes    **128 bytes** cache line

Then why we set it so big when we design this cache?

ANS: large array with strong spatial locality
can benefit from large cache line size

| a1 | a2 | a3 | a4 | a5 | a6 | a7 | … |

Data that has temporal locality

X

Cache will fetch the full 128B data
From memory for a single X variable

Redundant data that consumes bandwidth

Conclusion: traditional single cache design is a **compromise**
between different access needs of different types of data.
*It is just this compromise that creates the margin to shrink bandwidth.*