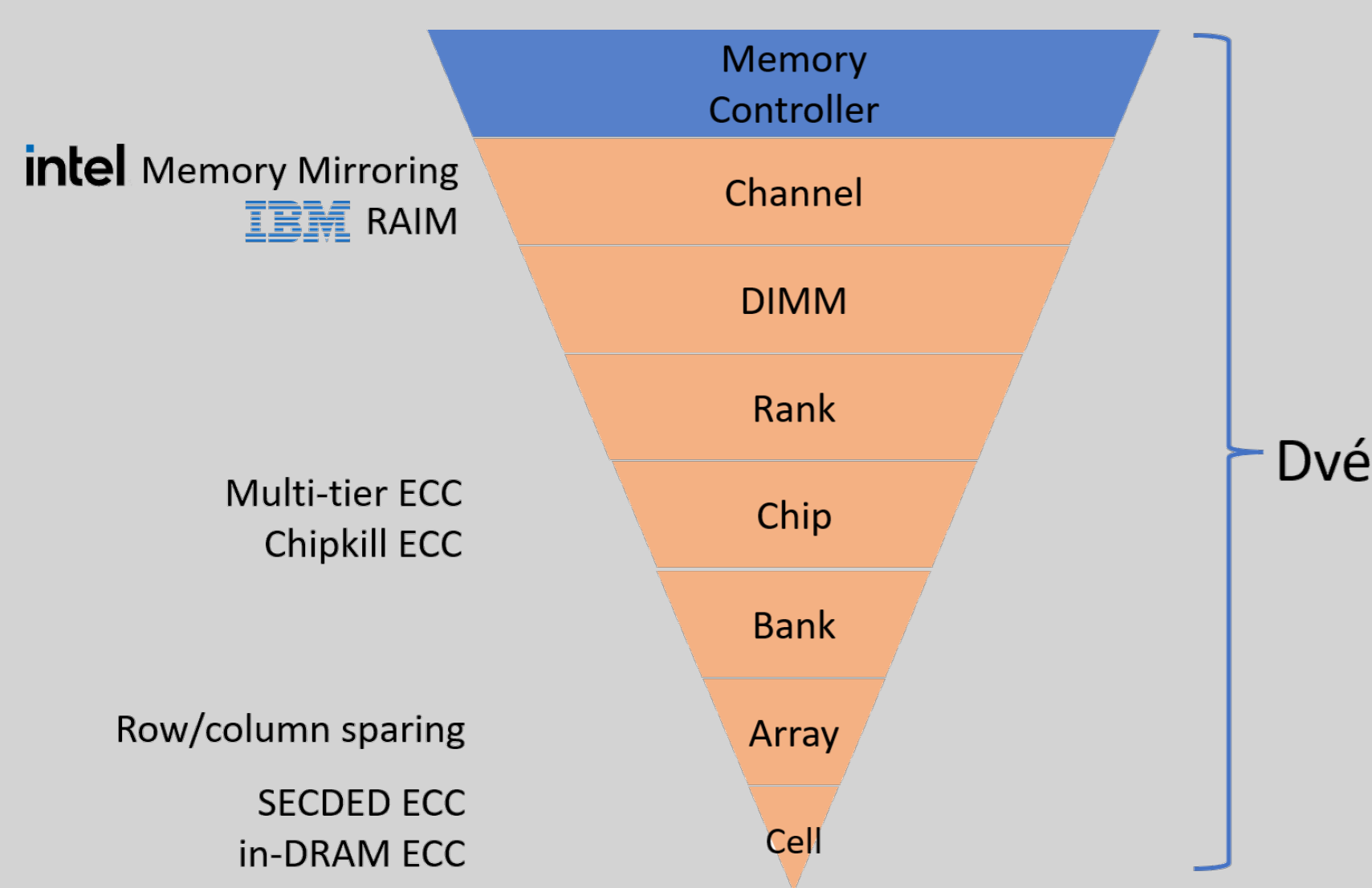


## Motivation and Design space



### PROBLEM

- ▲ Rapid increase in memory errors - variety and granularity
- ▲ Incremental increases in error protection mechanisms - to catch up with new errors
- ▲ Performance overheads - additional checks in critical path
- ▲ No flexibility in memory reliability + Extensive memory under-utilization

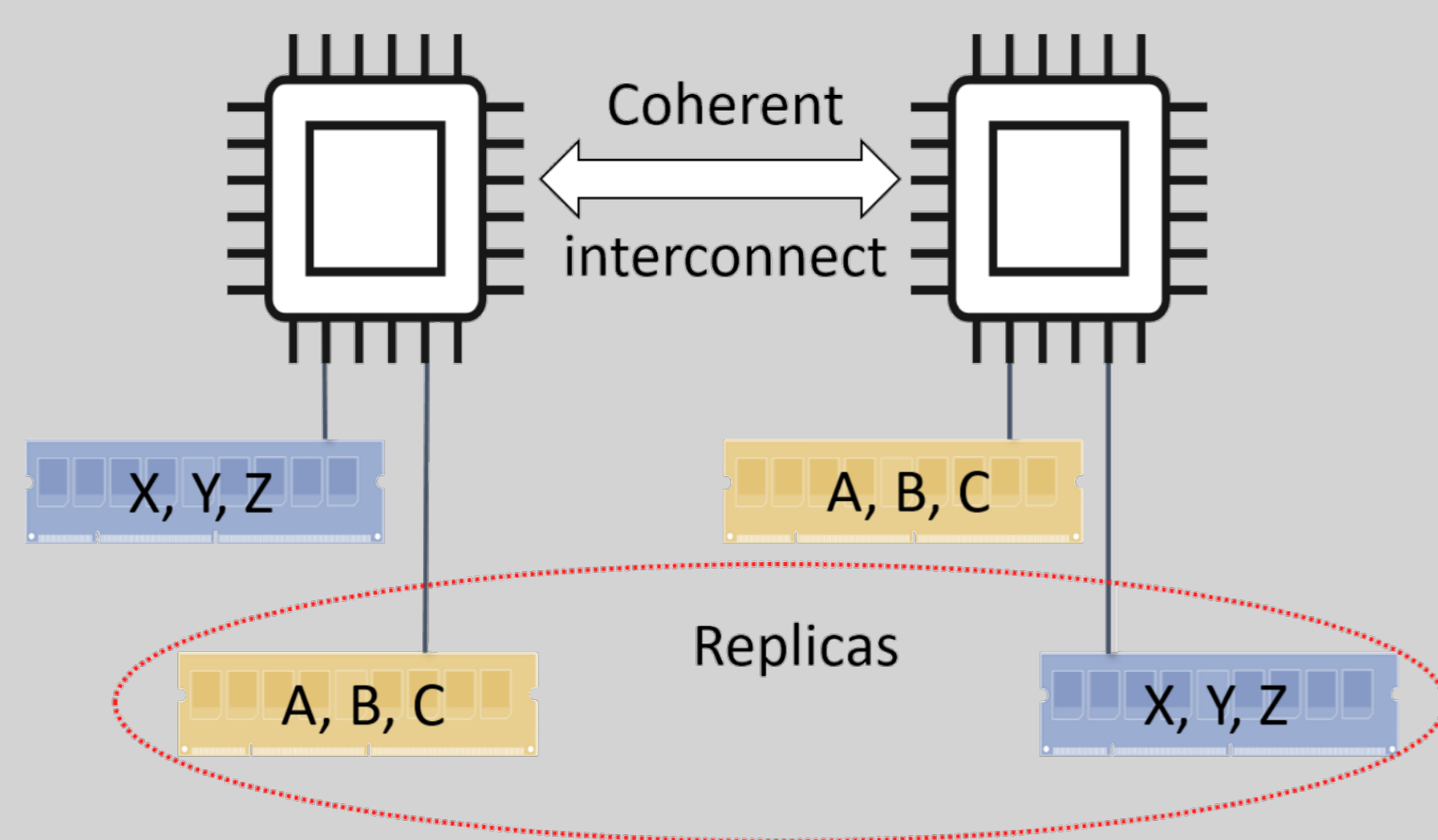
### KEY IDEA

- 💡 Memory protection at the highest end point i.e., Memory Controller – subsuming all other error types
- 💡 Hardware-driven active replication of data across cache-coherent NUMA nodes

## Dvé = Reliability + Performance + On-demand operation

### 1 Improved reliability by Data Replication

INSIGHT: Full data replicas, kept far apart and disjoint within a system



- Replicate data across different sockets in a NUMA system
- Detection - existing ECC, strong detection-only codes, temperature sensors etc.
- Correction - read replica

### Quantifying reliability benefits

Analytically modeling DRAM error rates

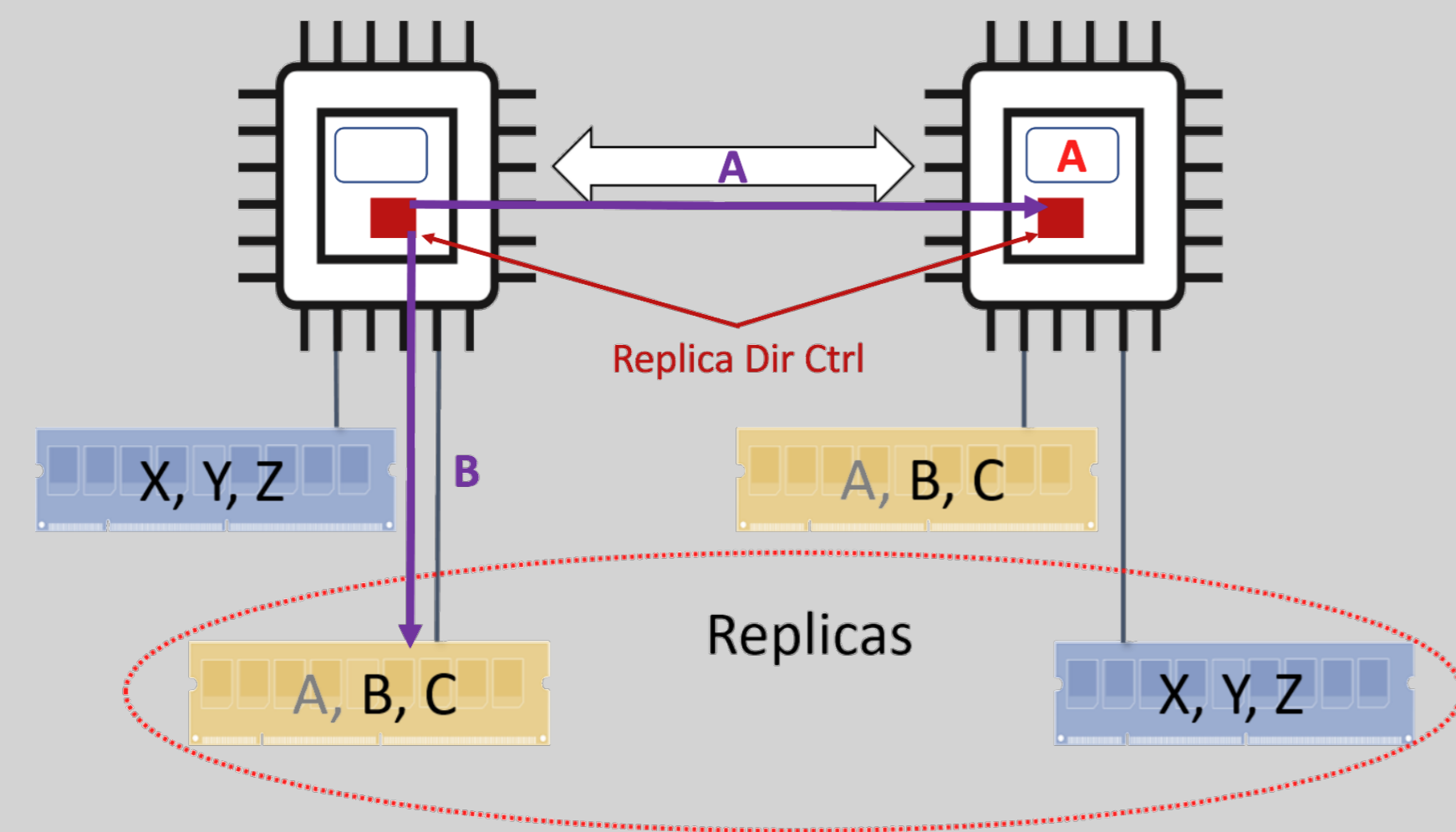
- Compute error rates per billion hours of operation
- Based on DRAM device FIT rate obtained from field studies
- DUE (Detected Uncorrectable Errors), SDC (Silent Data Corruption Errors)

Comparison against	DUE Rate Improvement	SDC Rate improvement
Chipkill (Dvé equipped with TSD)	4×	≈ 10 <sup>6</sup> ×
IBM RAIM (Dvé equipped with Chipkill)	172×	0.63×
Intel Memory Mirroring (Dvé equipped with TSD + temp scaled FIT rate)	11%	1×

- Intuition: “parallel-n” system (Dvé) vs “k-out-of-n” systems (ECC/RAIM)
- Lower bound analysis of the reliability benefits

### 2 Improved performance using Coherent Replication

INSIGHT: Route memory requests to nearest replica, when safe

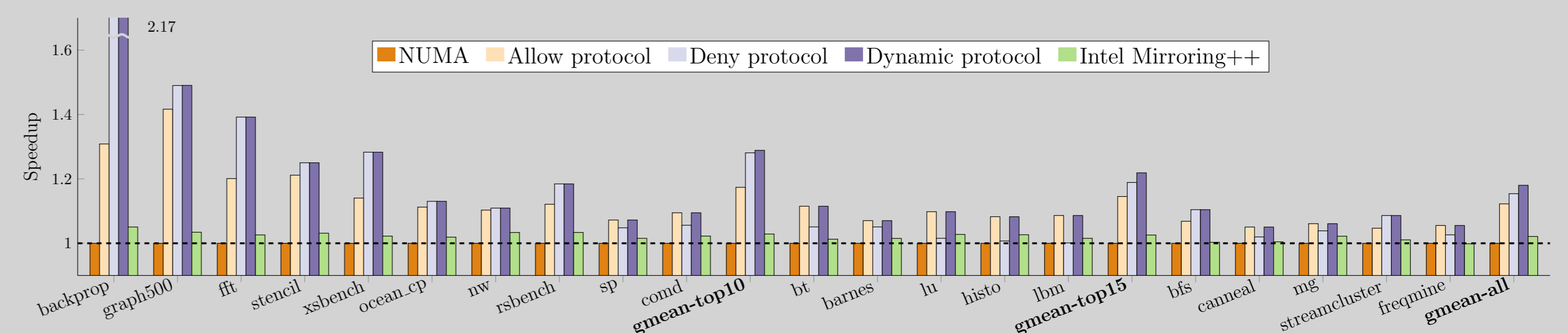


- Provide coherent access to both replicas during fault-free operation - for performance
- Maintains replica in sync - for reliability
- Builds hierarchically over existing cache coherence protocols
- 2 families of protocols to achieve coherence replication  
allow-based (lazy/pessimistic pull scheme), deny-based (aggressive/optimistic push scheme)

### Evaluating performance gains

Simulator driven methodology

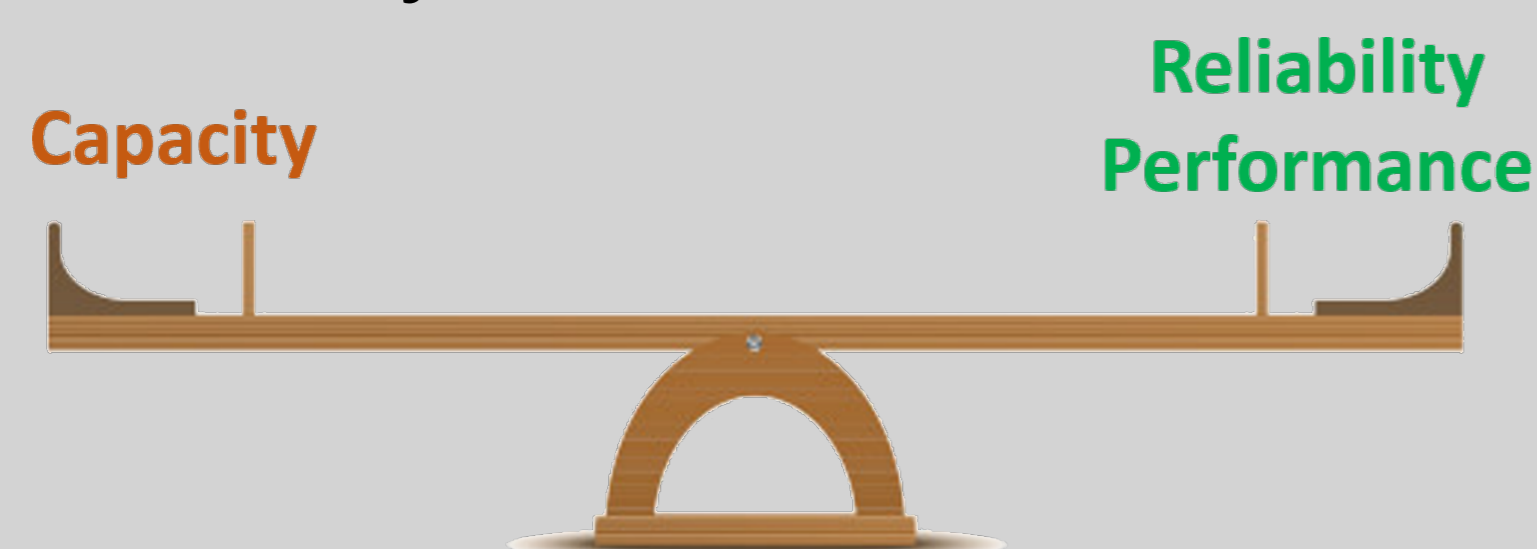
- SynchroTrace gem5: Intel Icelake-like, 2 sockets (16-cores), 2x 8GB DDR4 DRAM
- Workloads: Multi-threaded benchmarks  
NAS PB, Parboil, Rodinia, PARSEC, SPEC 2017, assorted HPC workloads
- Comparison against:  
Baseline NUMA - requests routed to node where data is housed  
Improved Intel Memory Mirroring - load balance reads between mirrored channels



- 12% (allow), 15% (deny) and 18% (dynamic) geomean speedup over baseline
- 13% geomean speedup over an improved Intel Memory Mirroring scheme

### 3 Flexible trade-off between capacity and reliability

INSIGHT: Opportunistically use the idle or under-utilized memory capacity



- Dynamically uses the idle memory capacity present in servers today
- Overheads applicable as and when demanded by the application, not fixed at design time

### Providing on-demand reliability

Mapping between physical address ↔ replica physical address

Dynamic Scheme:

- Where? Carve/manage space for replication - max DRAM resident set size, balloon drivers
- How? Mapping replica page pairs - OS managed Replica Map Table, walked in hardware
- When? Enable/disable as a soft setting by control plane (per-VM, per-container, kernel-only) or specified by app at malloc

OR

Static scheme:

- System wide replication: Fixed function mapping

## Summary and takeaways

- Dvé is a unique design point with new trade-offs in DRAM reliability design space  
Robust, holistic protection, decoupled from error detection to correct any error
- Dvé provides higher reliability + performance for workloads that do not require entire memory capacity

🔗 Artifacts - simulator source code, verified full protocol, FAQs, etc.  
<https://github.com/adarshpatil/dve> and <https://adar.sh/dve>

