TeraCache: Efficient Caching over Fast Storage Devices

Iacovos G. Kolokasis\textsuperscript{1,2}, Anastasios Papagiannis\textsuperscript{1,2}, Foivos Zakkak\textsuperscript{3}, Shoaib Akram\textsuperscript{4}, Christos Kozanitis\textsuperscript{5}, Polyvios Pratikakis\textsuperscript{1,2}, and Angelos Bilas\textsuperscript{1,2}

\textsuperscript{1}Foundation for Research and Technology – Hellas GR, \textsuperscript{2}University of Crete GR, \textsuperscript{3}Red Hat Inc., \textsuperscript{4}Australian National University AU

Fast Storage Devices Available but Analytic Stacks Not Ready

Increasing Memory Demands for Analytics
- Analytic servers use caches for avoiding recomputation (compute caches)
- Cache size is often several times the input dataset size
- DRAM scaling is limited (more $ per GB)
- Analytics resort to high capacity, fast storage devices, such as NVMe

Merging On-Heap and Off-Heap Benefits

Pros
- No Serialization
- High GC Time

Cons
- Low GC Time
- High Serialization

TeraCache: Best of Both Worlds!

Machine Learning Workloads
- RDD caching is critical in Spark
- GC and serialization introduce significant overhead
- TeraCache improves ML workloads performance by 25% over the state-of-the-art