**1. Motivation**

- High Performance Computing applications have dataset sizes that often exceed the most commonly available DRAM capacities.
- Emerging memory technologies that are much cheaper, such as Non Volatile Memory, are used to extend the memory space creating a *heterogeneous memory subsystem*.
- Data in Non Volatile Memory will incur higher access latencies, affecting the application performance, slowing it down compared to an ideal case when all data could fit in DRAM.
- Existing solutions **reduce the performance slowdown** by prioritizing allocations of the most frequently accessed objects in DRAM. They have limited utility in a shared hardware setup.

**2. Problem Statement**

**Problem Statement**: How to prioritize DRAM object allocations, so as to reduce the performance slowdown across all collocated applications?

**3. Observations**

- Not all applications are slowed down in the same degree when accessing Non Volatile Memory.
- Not all data objects of an application help reduce the performance slowdown when allocated in DRAM.

**4. CoBenefit Metric**

- Need for a metric that can capture the priority order of DRAM object allocations across all collocated applications.

<table>
<thead>
<tr>
<th>RunTime</th>
<th>Objects in DRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>All</td>
</tr>
<tr>
<td>t(O)</td>
<td>object O</td>
</tr>
<tr>
<td>S</td>
<td>None</td>
</tr>
</tbody>
</table>

- Normalize: \( F = \frac{S}{F} \)
- Scale: \( S = 0 \)

**5. CoMerge Solution**

- CoMerge prioritizes DRAM object allocations following the global CoBenefit descending order.
  - **Lower runtime** across all collocated applications.
  - **Higher DRAM utilization**.

**6. Future Directions**

- OS level solution that dynamically places and migrates data objects (or parts of them) across the heterogeneous memory substrate.

- Reduce overall system cost.
  - Determine the least amount of DRAM that is crucial for performance.
  - Leverage the fact that Non Volatile Memories offer access latencies that bridge the gap between DRAM and Storage (Flash / DDR).