Introduction

- **Power consumption** is one of the major concerns in HPC
  - Desired features: power capping and power shifting [1]
- **Memory bandwidth/capacity limitations** are also severe
  - Promising solution: hybrid main memory architecture
    - DRAM + NVRAM (Intel Optane SSD[2])
    - MCDRAM + DDR4 (Intel Knights Landing[3])

**Goal of this research**

A sophisticated power management scheme for hybrid memory based systems

Motivation

**Key observation:** the effective bandwidth decreases as the footprint size scales (Fig. 1) on a hybrid memory system

As a result, the performance bottleneck changes depending on the footprint size (Fig. 3)

**Objective:** Maximizing performance (Perf) under a total power constraint ($P_{total}$) by controlling CPU/memory power ($P_{cpu}/P_{mem}$) for a given footprint (or problem) size ($S$)

- **Formulation:**
  \[
  \text{max } \frac{\text{Perf}}{P_{cpu} + P_{mem}} \leq S \text{ s.t. } P_{cpu} + P_{mem} \leq P_{total}.
  \]
- **Power Shifting Result**
  - **CPU intensive case** (Fig. 5): Shifting power from $P_{mem}$ to $P_{cpu}$ improves performance because the CPU becomes less critical
  - **Memory intensive case** (Fig. 6): Shifting power from $P_{mem}$ to $P_{cpu}$ improves performance – the DRAM needs less power due to frequent NVRAM accesses

**Synthetic streaming code**

- **Power Shifting Result**
  - **CPU intensive case** (Fig. 5): Shifting power from $P_{mem}$ to $P_{cpu}$ improves performance because the CPU becomes less critical
  - **Memory intensive case** (Fig. 6): Shifting power from $P_{mem}$ to $P_{cpu}$ improves performance – the DRAM needs less power due to frequent NVRAM accesses

**Power Shifting Result**

- **Objective:** Maximizing performance (Perf) under a total power constraint ($P_{total}$) by controlling CPU/memory power ($P_{cpu}/P_{mem}$) for a given footprint (or problem) size ($S$)

**Solution**

Footprint Aware Power Shifting: Shifting power between $P_{cpu}$ and $P_{mem}$ in accordance with the footprint size ($S$)

- We should allocate more power on the bottleneck component, which highly depends on the footprint size (Fig. 3)

Experiment

We test various combinations of ($P_{cpu}/P_{mem}$) and choose the best one for each footprint size ($S$) under a given $P_{total}$ ($=P_{cpu}+P_{mem}$)

**Evaluation Setting**

- **System Configuration:** Summarized in Table 1
- **Power Management:** Running Average Power Limit (RAPL)[5]
- **Workloads:** FFT, Lulesh, and the synthetic streaming code shown in Fig. 2 (Streaming)

**Environment**

- **System Configuration:**
  - CPU: Xeon Gold 6154 Processor (Skylake), 18 cores, 3.0GHz, TDP 200W x2 sockets
  - Memory: DRAM: DDR4-2666 x12 modules, 12ch, 192GB
  - NVRAM: Intel Optane SSD P4800X, 375GB, 2.4GB/s(read), 2.0GB/s(write)
  - Data management: IMDT [2]
- **OS:** Cent OS 7.4
- **Compiler:** Intel C++/Fortran Compiler 17.0.4
- **Options:** -O3, -qopenmp

**Table 1. System Configuration**

<table>
<thead>
<tr>
<th>CPU Package</th>
<th>Xeon Gold 6154 Processor (Skylake), 18 cores, 3.0GHz, TDP 200W x2 sockets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory System</td>
<td>DRAM: DDR4-2666 x12 modules, 12ch, 192GB NVRAM: Intel Optane SSD P4800X, 375GB, 2.4GB/s(read), 2.0GB/s(write) x2 cards</td>
</tr>
<tr>
<td>Data management</td>
<td>IMDT [2]</td>
</tr>
<tr>
<td>OS</td>
<td>Cent OS 7.4</td>
</tr>
<tr>
<td>Compiler</td>
<td>Intel C++/Fortran Compiler 17.0.4</td>
</tr>
<tr>
<td>Options</td>
<td>-O3, -qopenmp</td>
</tr>
</tbody>
</table>

**Results**

- **Footprint-aware power shifting is promising to improve the performance of power constrained hybrid memory based systems**

**Future Work**

- Developing a software framework, a performance model, and a power allocation algorithm to realize our proposal
- Including NVRAM power management
- Evaluating with other hybrid memory based systems such as Intel Knights Landing

References