epcc

Persistent Memory Exercises

Practical 2

Adrian Jackson

1 Introduction

In this exercise we are going to investigate the NUMA nature of the B-APM installed in the compute nodes in the system. We will still be accessing the B-APM as a filesystem, but we will require some modifications of the programs we have already used to ensure that the best performance is achieved when accessing the hardware.

Before starting this practical you should ensure you have already completed Practical 1 (https://github.com/NGIOproject/PMTutorial/blob/master/Exercises/practical1.pdf), which also has detailed instructions on accessing and using the system.

This practical is designed to highlight the fact that the persistent memory we are exploiting for high I/O performance here is embedded directly into the compute node and is attached to the individual processors in the node. This means that there are two regions of this B-APM memory, and accessing the correct region for the processor you are using ensures good performance.

2 NUMA-aware I/O

We should now have been able to run IOR using Lustre and also the filesystems mounted on the compute nodes (i.e. /mnt/pmem_fsdax0/), and streams using DRAM. For this practical the exercise is to modify the IOR source code so that it can exploit both the filesystems on the compute node (i.e. /mnt/pmem_fsdax0 and /mnt/pmem_fsdax1), and ensure that processes running on socket 0 use pmem_fsdax0 and those running on socket 1 use pmem_fsdax1.

To compute which socket a process is running on you can use this code:

```
unsigned long GetProcessorAndCore(int *chip, int *core){

   unsigned long a,d,c;

 __asm__ volatile("rdtscp" : "=a" (a), "=d" (d), "=c" (c));

   *chip = (c & 0xFFF000)>>12;

   *core = c & 0xFFF;

   return ((unsigned long)a) | (((unsigned long)d) << 32);;

}
```

You can then use that, with some string manipulation, to create the string `/mnt/pmem_fsdax0` or `/mnt/pmem_fsdax1`. This code will need to be added into the file `IOR.c`, in the function `GetTestFileName`. Alter this to create the filename based on the socket number.

Once you have made your modifications, recompile you code and re-run the fsdax test as outline in the Practical 1 exercise sheet. What performance do you now see from the B-APM using the fsdax filesystem?

fsdax Performance: _____