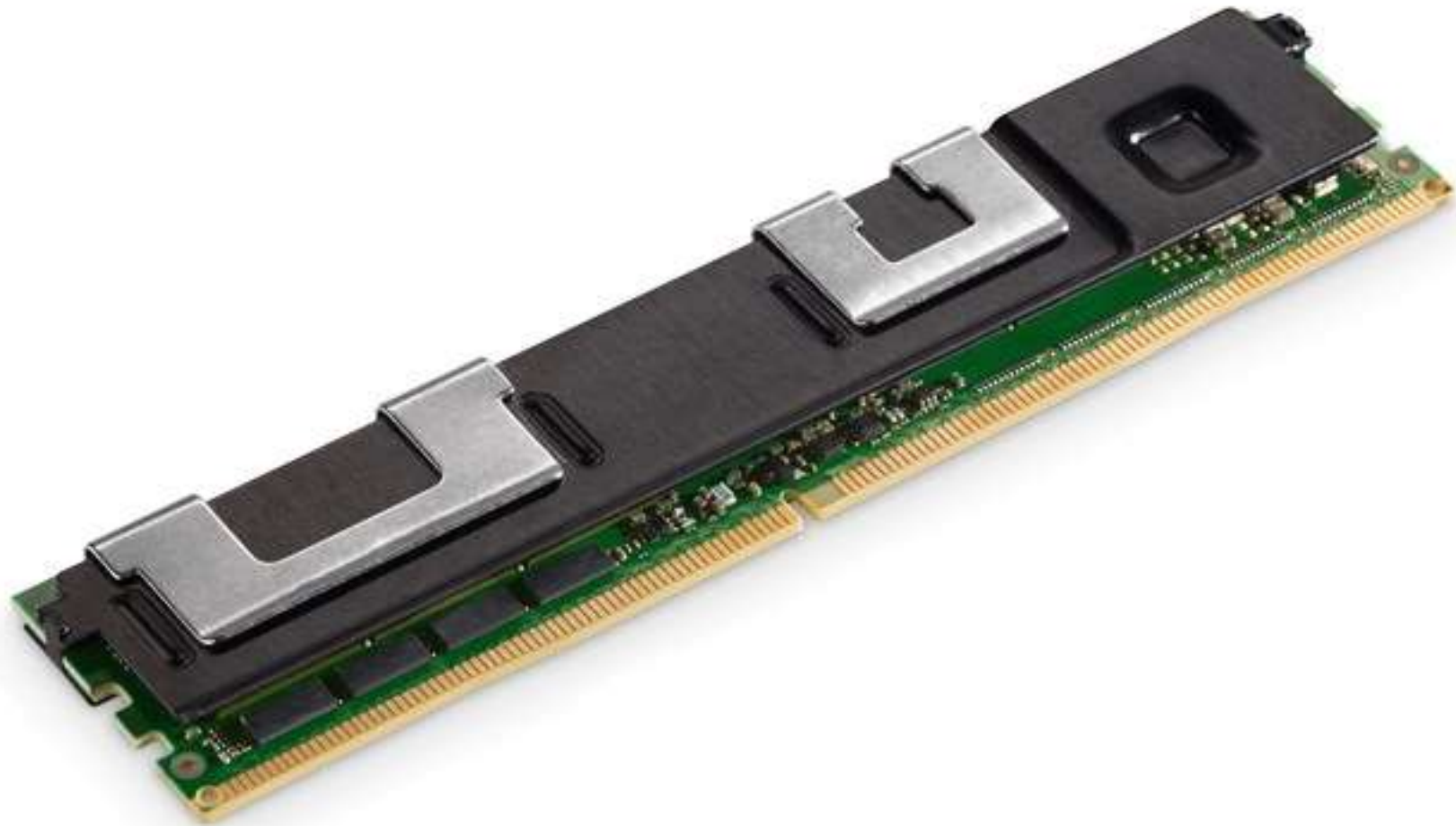


# In-node storage and memory-like I/O

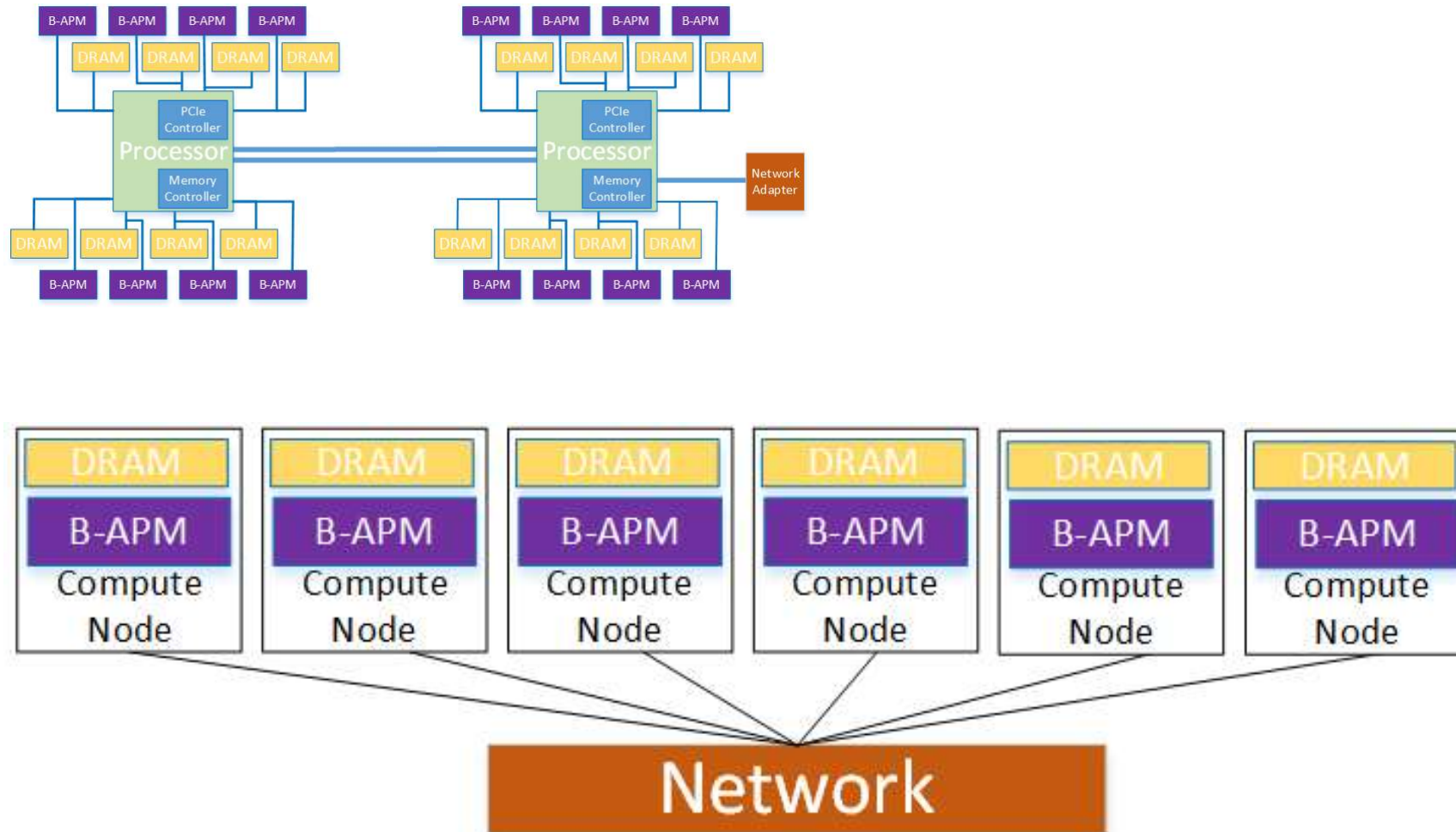
Adrian Jackson  
EPCC, The University of Edinburgh  
[a.jackson@epcc.ed.ac.uk](mailto:a.jackson@epcc.ed.ac.uk)  
[@adrianjhpc](https://twitter.com/adrianjhpc)

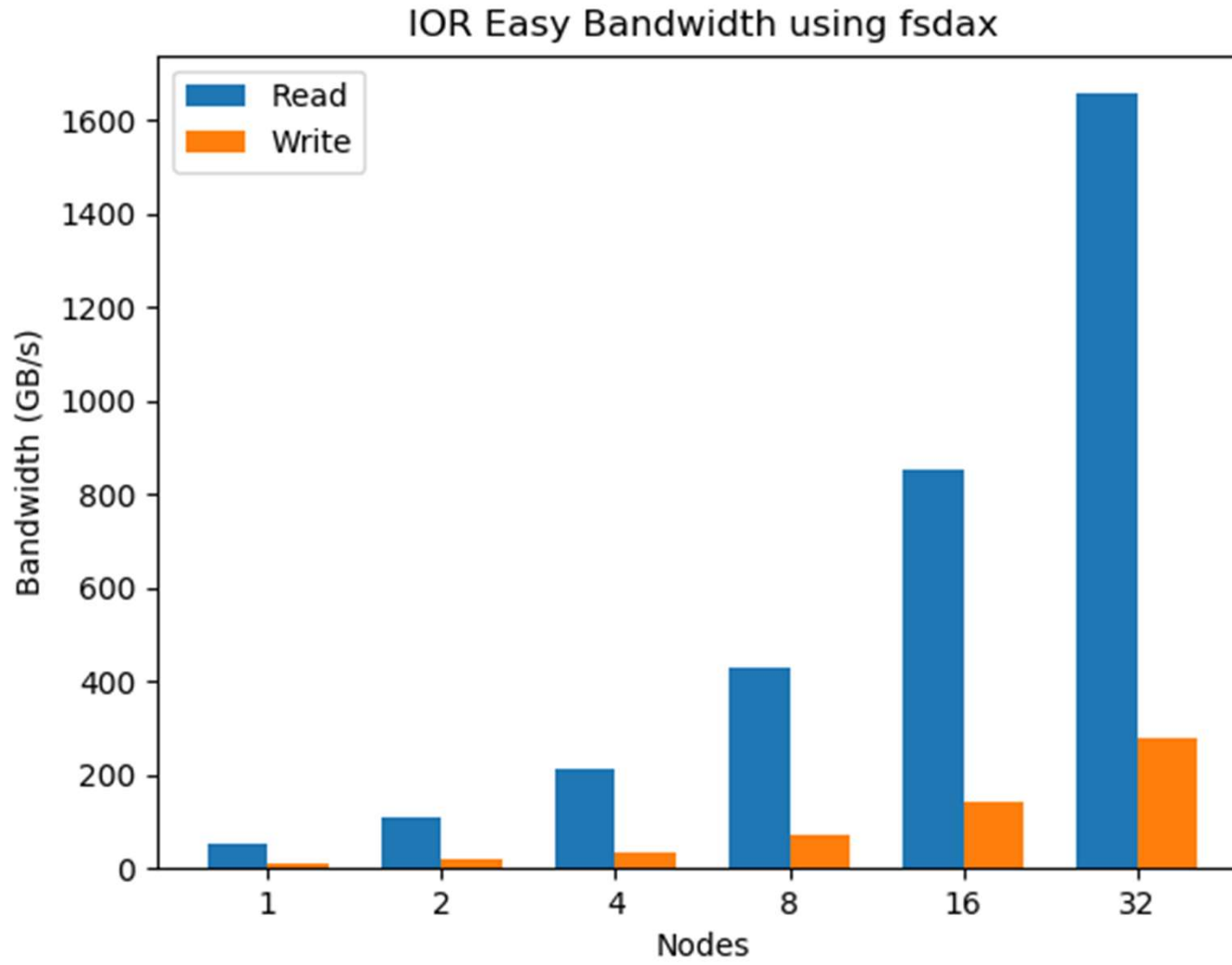
# NVRAM / B-APM

---

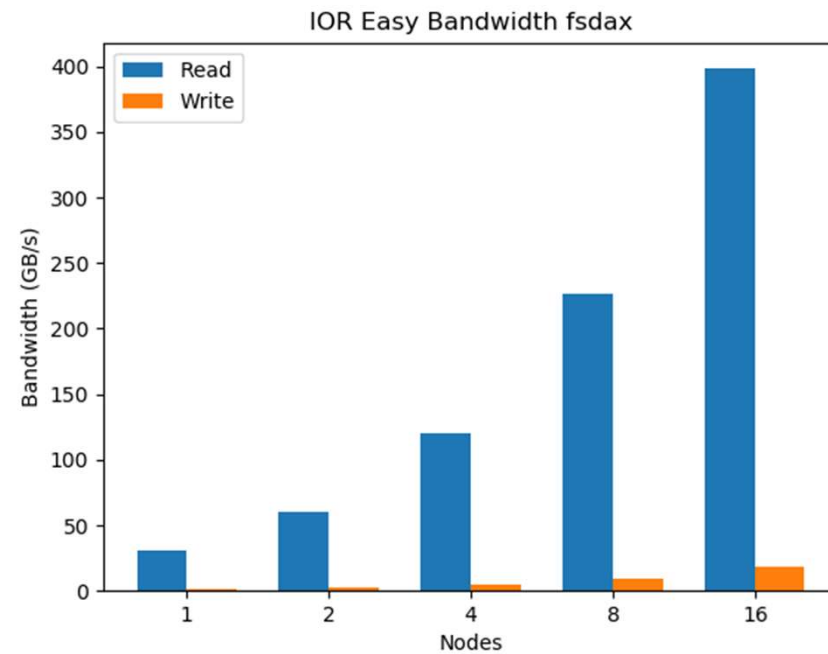
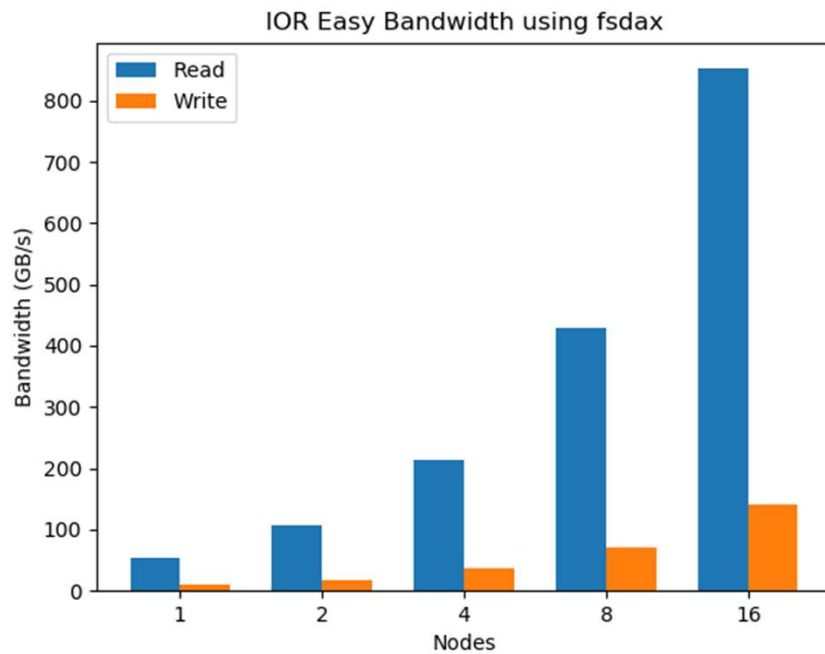


# In-node storage





# NUMA effects



Single mount point used  
48 processes per node

# Dual (multi) mounts

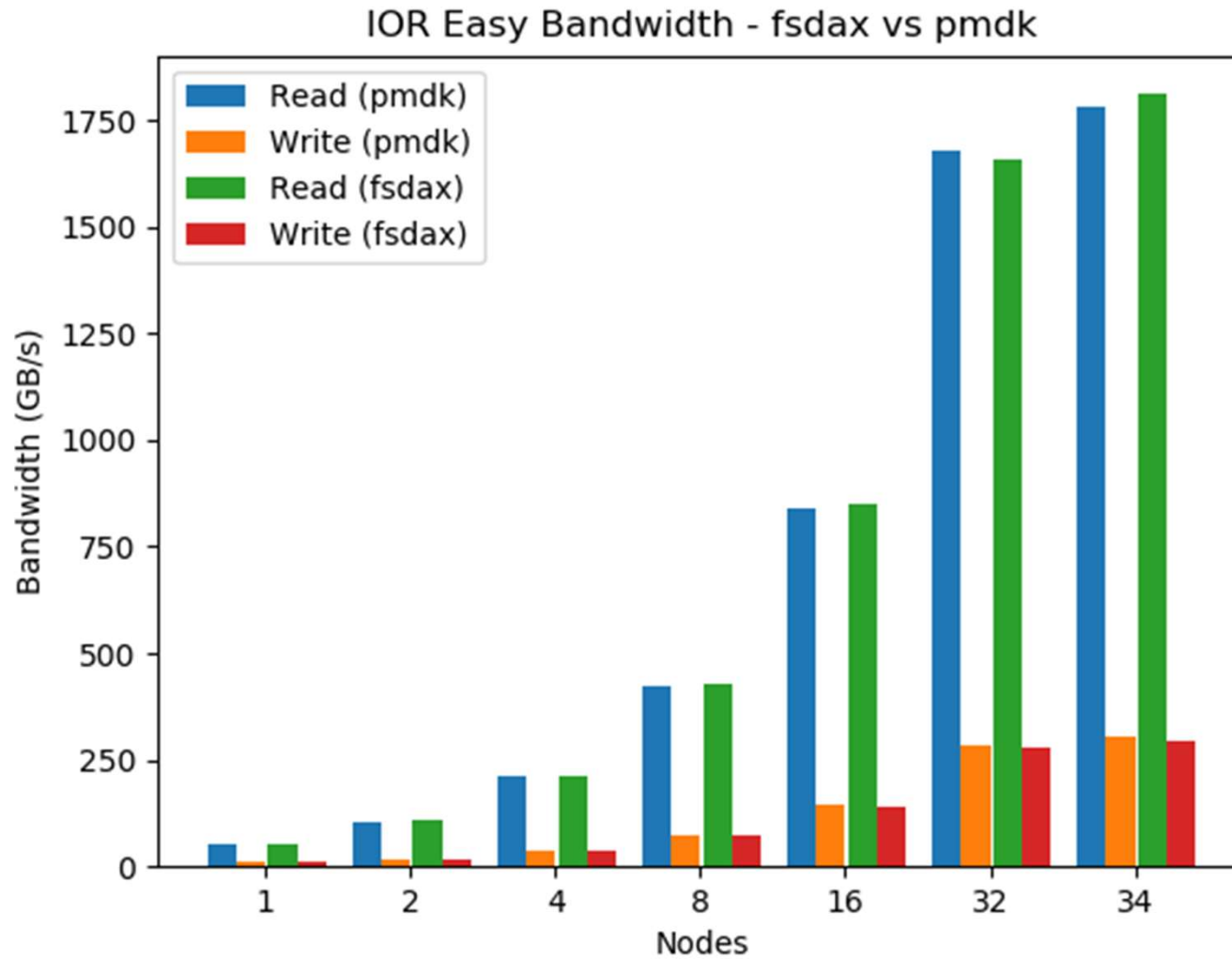


```
IOR START
  api=MPIIO
  repetitions=1
  multiFile=0
  interTestDelay=10
  readfile=1
  writefile=1
  filePerProc=1
  checkWrite=1
  checkRead=1
  keepFile=0
  quitOnError=0
  segmentCount=1
  singleXferAttempt=0
  individualDataSets=0
  ...
  useO_DIRECT=1
  showHints=1
  transferSize=128
  reorderTasksConstant=0
  dualMount=1
  testFile=/mnt/pmem_fsdax

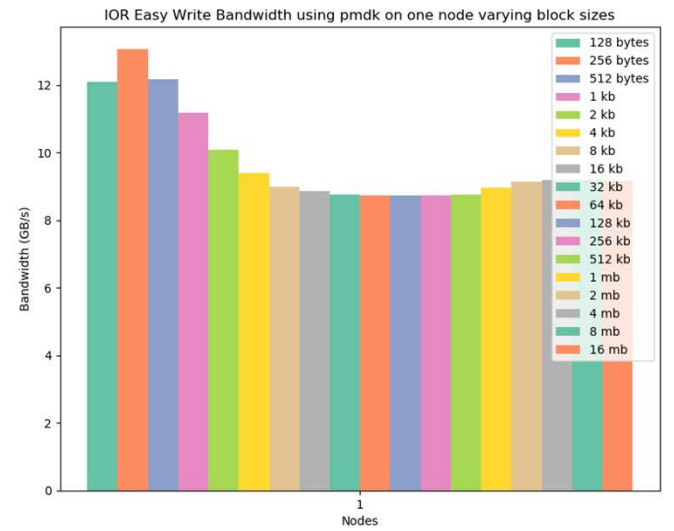
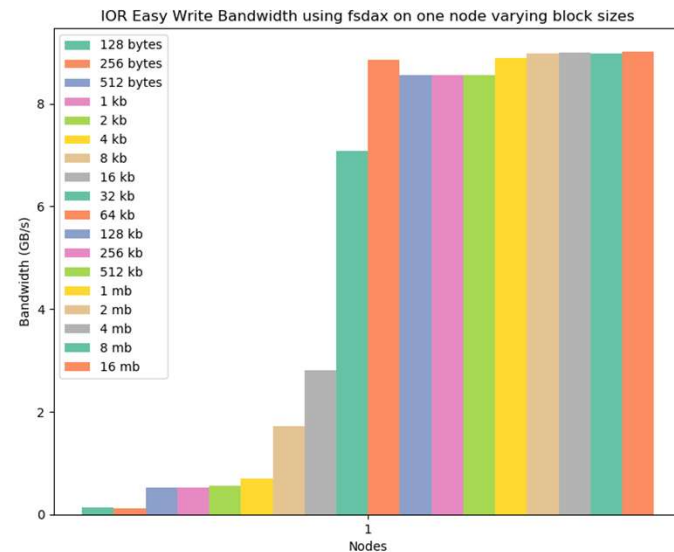
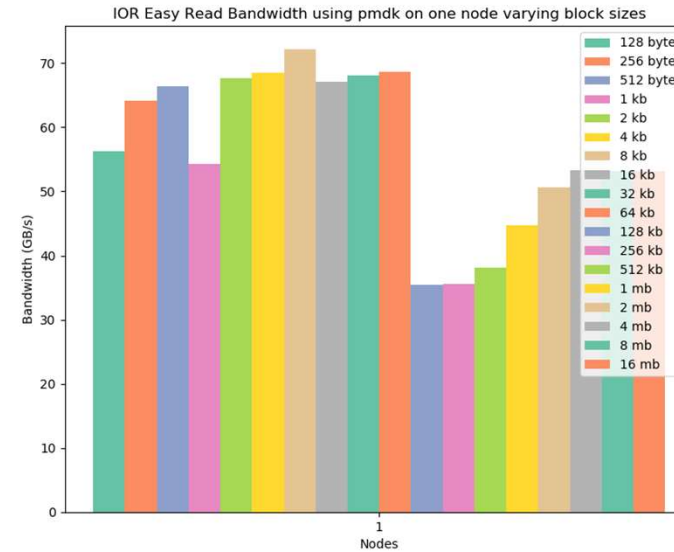
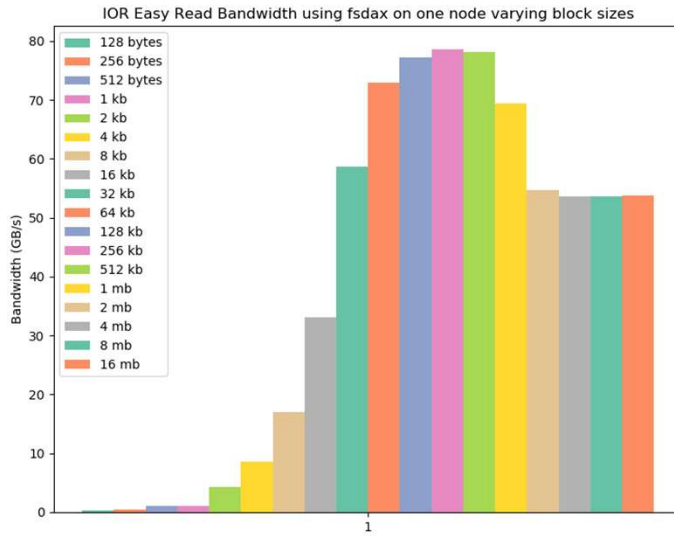
  if(test->dualMount) {
    GetProcessorAndCore(&socket, &core);
    sprintf(tmpString, "%s%d/%s",
            initialTestFileName, socket, "data");
    strcpy(initialTestFileName, tmpString);
  }

RUN
IOR STOP
```

---



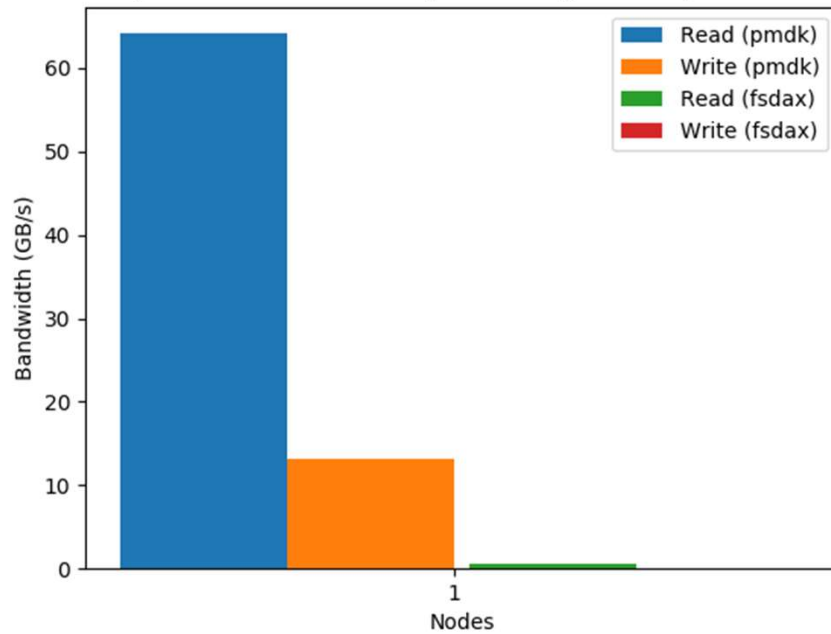
# IOR - Data block sizes



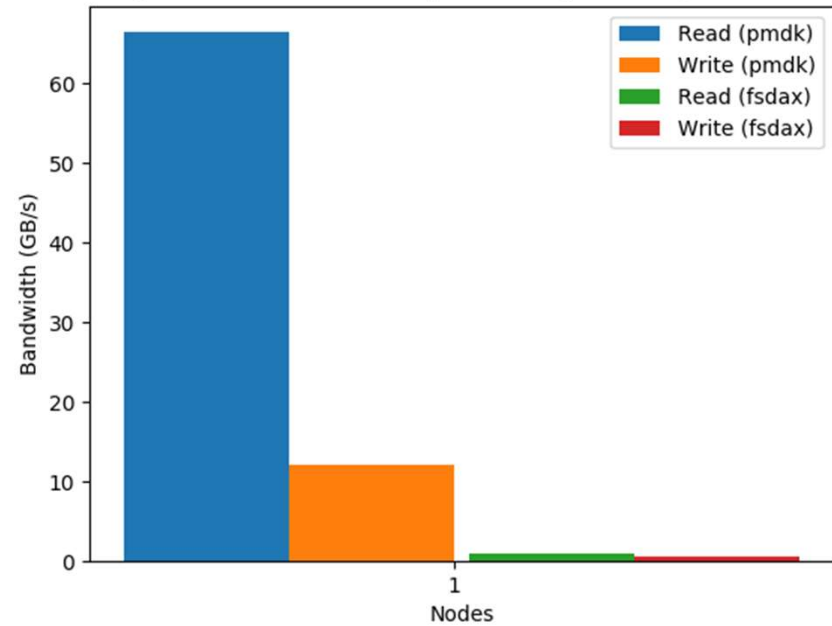


# Data access sizes

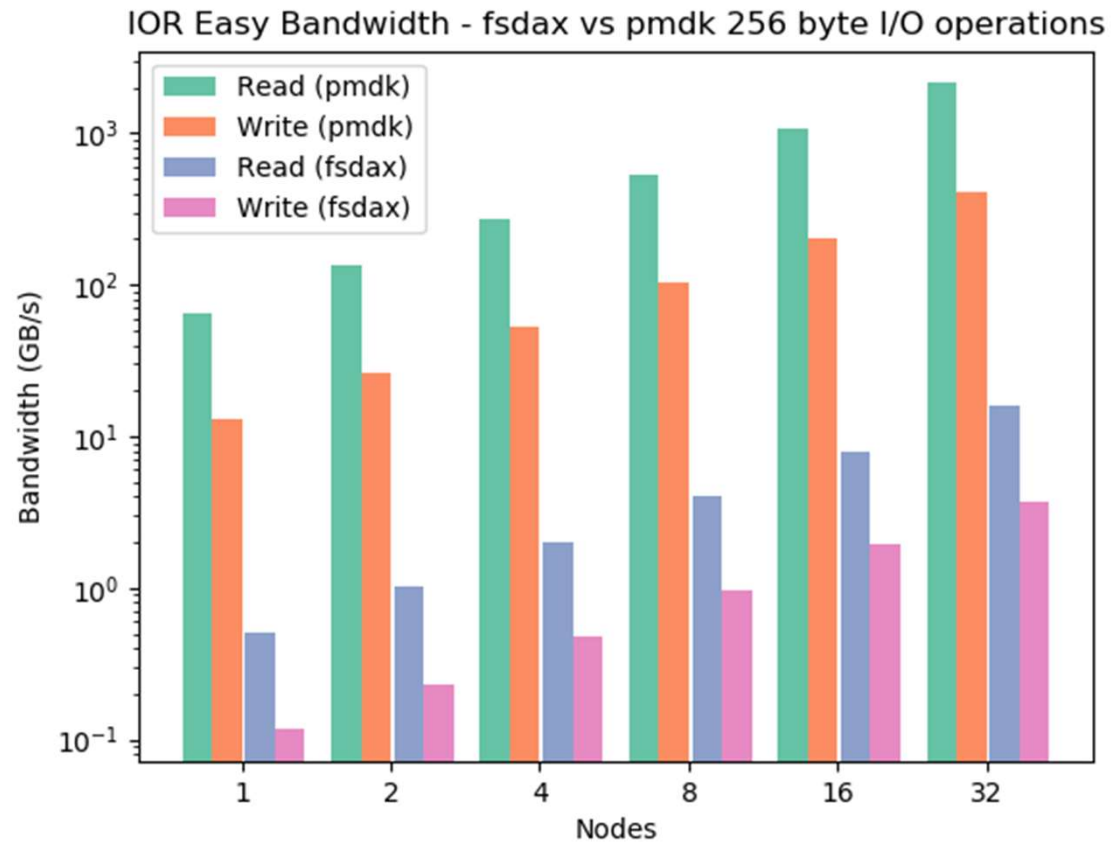
IOR Easy Bandwidth - fsdax vs pmdk using a 256-byte transfer size



IOR Easy Bandwidth - fsdax vs pmdk using a 512-byte transfer size



# Data access sizes



- In-node storage
    - Contention impact
    - Overall resource usage
  - NUMA I/O issues
    - Dual (or more) mount points
  - Small (or a range of) I/O operation sizes
    - Recognising different I/O usage modes/models
  - Moving beyond POSIX file interfaces
-