Blocking vs Non-Blocking Communication Between Quantum Chemistry Applications and Middleware

- GAMESS has been integrated with NICAN middleware tool.
- The existing communication is via Sockets API. (Blocking communication)
- Non-blocking functionality is implemented. It eliminates the overhead of processing by NICAN.
- Threading is used to acheive the Non-blocking functionality. A thread spawned for each function call by GAMESS is serviced by NICAN in parallel with other threads.
- For performance testing of GAMESS, wall clock times of several runs in different scenarios including both Blocking and Non-blocking modes are recorded.

Test Bed

Machine Used

- Dynamo

(iprt, IASTATE)

→35 Dual-Quad core Intel Xeon nodes
→16 GB RAM/node
→1.5 TB scratch/node

Molecule Used

- AT.cc-pVDZ

Input Combinations

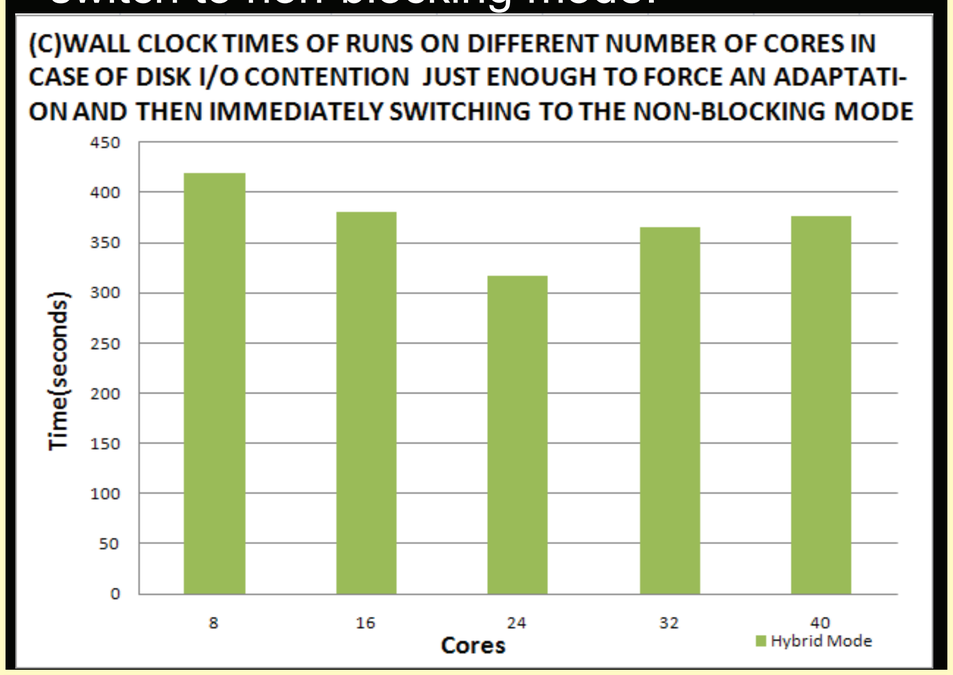
- MP0
- Conventional

Data Collected

- Wall clock times

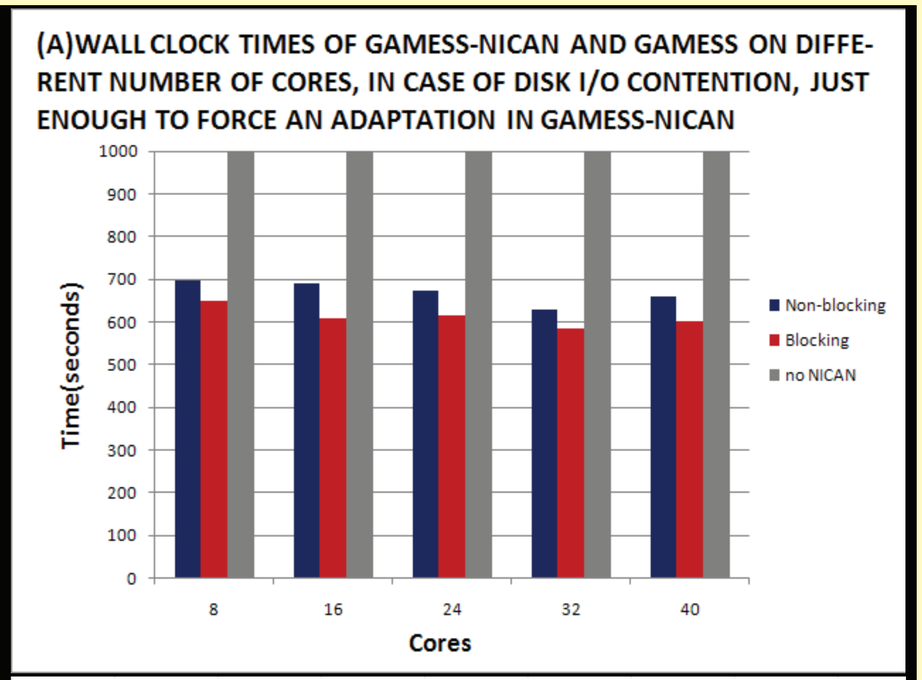
A Better Approach

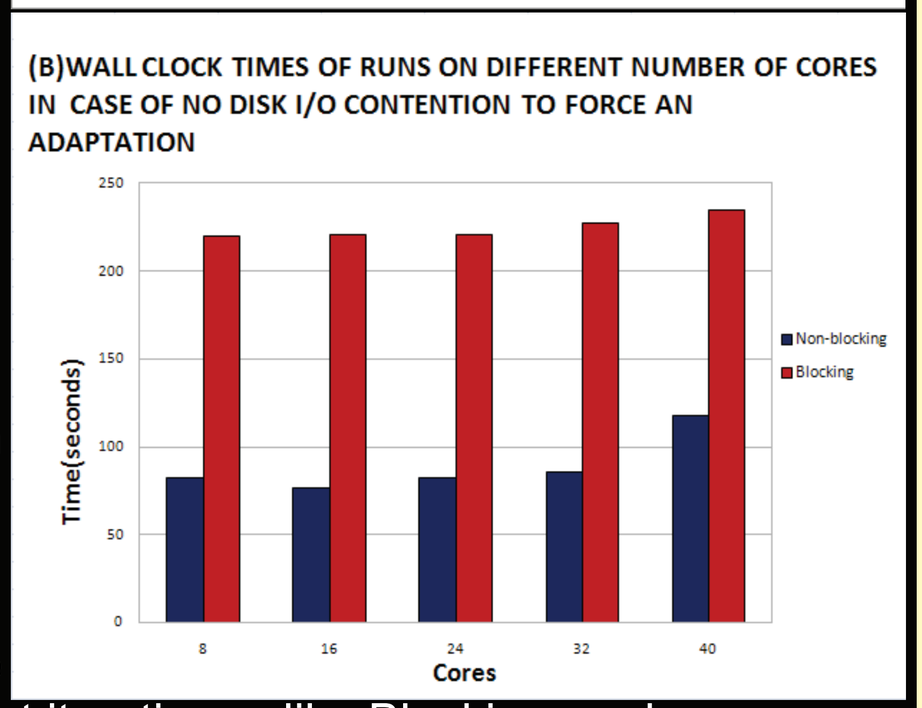
- Blocking mode of communication is used until an adaptation is invoked.
- Later, once an adaptation is invoked, switch to non-blocking mode.



Experiment Setup

- Disk I/O Contention may exist through-out the run-time of a job. This is achieved using a file system bench-mark utility IOzone.
- NICAN takes 10 seconds to process a request and return(this can be attributed to the load on NICAN and a weak Network connection).
- Runs are conducted on 8, 16,24,32,40 cores on a set of 5 compute nodes on the Dynamo.





Observations

- Non-Blocking mode in graph (A) does not perform better as there is an overhead of an iteration running in conventional mode during Disk I/O contention.This is due to the fact that GAMESS is not able to see the adaptation until next iteration unlike Blocking mode.
- GAMESS without NICAN took more than 1000 seconds (graph A).
- Non-blocking mode in graph (B) performs better as there is no need for an adaptation, averting the overhead of processing by NICAN.
- Relative structure of the graph (B) does not change whatever time NICAN takes for the processing of a request.
- Relative structure of the graph (B) is not altered even if there is Disk I/O Contention that does not force GAMESS to adapt.

Contact Information: Masha Sosonkina, masha@scl.ameslab.gov
or Sai Kiran Talamudupula, kiran123@slc.ameslab.gov