Simplified Interface to Complex Memory

Sean Williams, PhD
Los Alamos National Laboratory

Problem

- Exascale nodes expected to have complex, heterogeneous memory systems
- Users won't want to learn low-level technical details
- Heterogeneity is the enemy of portability
- Middleware requries unified low-level abstraction

Proposed Solution

Two-level library:

- Low-level interface:
  - Homogeneous abstraction over different memory devices
  - Classifies devices according to designer intent
  - Unified method of querying, allocating, and freeing
  - Intended for advanced users or middleware developers
- High-level interfaces:
  - Expose memory systems in intentional terms, rather than technical
  - Choose where to allocate, when to migrate
  - Many possible implementations
  - Basis for other research into memory management

Project Team

Point of Contact: Mike Lang (mlang@lanl.gov)
Roger Pearce (pearce7@llnl.gov)
Maya Gokhale (gokhale2@llnl.gov)
Simon Hammond (sdhammo@sandia.gov)
Sean Williams (swilliams@newmexicoconsortium.org)
Douglas Otstott (douglas.otstott@gmail.com)
Latchesar Ionkov (lionkov@lanl.gov)
Ada Gavrilovska (ada@cc.gatech.edu)
Greg Eisenhauer (eisen@cc.gatech.edu)
Thaleia-Dimitra Doudali (doudali@gmail.com)
Terry Jones (trjones@ornl.gov)
Aaron Welch (welchda@ornl.gov)

Preliminary Results

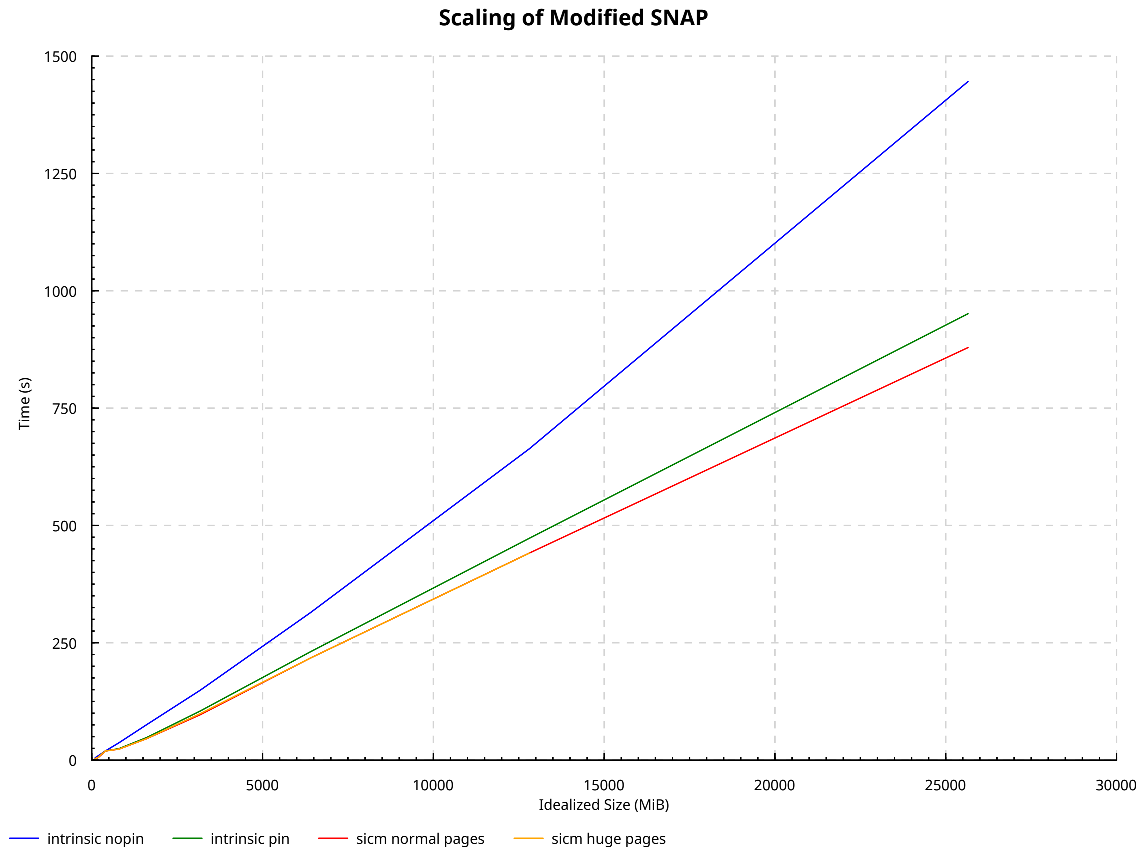
Early low-level interface:

- Currently provides abstraction over NUMA nodes, page size, and Intel Knights Landing high-bandwidth memory
- Written in C, with FORTRAN bindings

Added support to a proxy application:

- We use SNAP, a proxy particle transport simulation
- Tested NUMA pinning and huge pages

Results:



- Normal run, allocations with FORTRAN `ALLOCATE`
- Pin to single NUMA node, uses `ALLOCATE`
- Pin to single NUMA node, uses SICM
- Pin to single NUMA node, huge pages with SICM

Repository

https://github.com/lanl/SICM