This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations - the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and very-literate platforms, to support the nation’s exascale computing imperative.

Management of Deep Memory Hierarchy in the Exascale Era

ANL: Nicolas Denoyelle, Swann Perarnau, Brice Videau
LLNL: Maya Gokhale, Marty McFadden, Roger Pearce, Eric Green, Keita Iwabuchi
UCMerced and LLNL: Kai Wu, Dong Li

Incorporating novel memory types into exascale systems and applications

Within Argo memory thrust we are working on the best software techniques to incorporate complex new memory types into the memory hierarchy. We are currently pursuing two strategies: a transparent DRAM cache for NVRAM devices (UMap) and a managed scratchpad for high-bandwidth memory (AML).

UMap

Overview
- User Level Memory Map (UMap) is a memory map approach to incorporating NVRAM into complex memory hierarchy
- Exploit large virtual address space to access data sets directly
- Well suited to analyzing large observational and simulation data sets as if in memory
- Application (class) specific user mode handler migrates pages between far and near memory
- UMap handler is used in asteroid detection algorithm and graph traversal algorithms

Features
- User space flexibility, ease of use, self-defined handler (multiple files, distributed data sets)

UMap handler design
- Uses userfaultfd low-overhead protocol to allow applications to handle page fault in user space
- Fault is sent to the user space handler as a message, and handler loads page from file(s) or (future work) network
- Handler runs in application context, customizable for page size (multiple of system page) and page buffer size

Applications
- Transient detection
  - Detecting Near-Earth Asteroids (NEAs) through imaging data sets generated by optical camera
  - UMap combines many 2-D tiled images to form 3-D volume for search algorithm

- Out of core graph algorithms
  - Generate edge or adjacency list representation of graph from raw data (e.g. Wikimedia commons edits)
  - Traverses graph to compute features of interest for community detection

Prefetching/pipelining results
- Dense Linear Algebra
  - KNL in flat/quad mode
  - 3 overlapping layers of tiling:
    - Pipeline over MCDRAM-size tiles
    - OpenMP-task based, input size specific blocking to maximize bandwidth
    - Inner autotuned microkernel for AVX2 performance
  - As good as or better than vendor-provided libraries on very large inputs (bigger than MCDRAM)

AML

Overview
- Explicit, application-aware memory management:
  - Descriptive API for application-level data access,
  - Explicit placement and movement of data,
  - Abstract topology and memory device complexity.
- Collection of building blocks:
  - Generic: few assumptions about user application, hardware-oblivious,
  - Customizable: application users can specialize the inner implementation of each offered abstraction,
  - Composable: mix and match as needed.
- Locality optimizations for current and future hardware generations:
  - Static allocations with application insights,
  - Asynchronous (re-)mapping to optimize data locality on the go (data layout, HBM management, NUMA locality).
- Ongoing discussions with PETSc team, RIKEN, INRIA, BSC

Key Components
- Topology & hardware management (NUMA, hwloc, SICM)
- Data layout descriptions (application-specific)
- Tiling schemes (incl. ghost areas)
- Data movement facilities (move_pages, memcpy)
- Pipelining helpers (scratchpad, asynchronous requests)

UMap version 0.0.3 has been open sourced on github.
1. Umap URL: https://github.com/LLNL/umap
2. Umap apps URL: https://github.com/LLNL/umap-apps

AML is open source and is available here: https://xgitlab.cels.anl.gov/argo/aml

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations - the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and very-literate platforms, to support the nation’s exascale computing imperative.