Arigo improves or augments existing OS/R components for use in production HPC systems, providing portable, open source software that improves the performance and scalability and that provides increased functionality toexascale applications.

**AML**

**Overview**
- A library for application-aware management of byte-addressable memory devices
- Explicit placement and movement of data
- Designed as a collection of building blocks
  - Users can create custom memory management policies for allocation and placement of data across devices.
- Designed for deep, heterogeneous memory systems, featuring NUMA, HBM, or GPU memory.

**Impact**
- Improved performance of applications regarding memory usage on the complex compute nodes of exascale systems.
- Improved performance portability of applications across exascale systems.

**Before ECP**
- A proof-of-concept library then called DeepRAM
  - Focus on multi-level DRAM hierarchy on CPU
    - Software-managed scratchpad in NVRAM
  - Exploration of different migration mechanisms
    - User-space, kernel-space, hardware
    - Asynchronous using dedicated CPU threads

**Now**
- Production-quality implementation
- Major focus on GPUs, given the eventual architectures of first exascale systems.
- Integration into Exascale's X SBench
  - Interface to build custom memory mapping policies that are application-focused, on top of any GPU interface (OpenCL, CUDA, HIP, oneAPI).
- Duplication of latency-sensitive data across devices.
- Transformation, optimization of data layout on target accelerators.

**Future**
- Continuous improvements to application performance, including better use of GPU memory capacity, leading to better scaling.
- Towards a vendor-neutral, programming model.
  - Axiomatic memory management layer for future production systems.
- Increased use across exascale applications, and more portable performance across complex architectures.

**UMap**

**Overview**
- A library that enables user-space optimizations for memory mapping HWA devices into the complex memory hierarchy.
- Facilitates direct access to large data sets through virtual address spaces.
- Provides application-specific configurations suited to massive observational and simulation data sets.
- High-performance design features I/O decoupling, dynamic load balancing, and application-level controls.

**Impact**
- The UMap memory mapping abstraction is important for the blurring of the memory/storage hierarchy. UMap enables accessing file-resident data as memory.
- UMap breaks the dichotomy between memory and storage by providing a unified virtual memory interface and simplifying application code.
- UMap enables application-specific tailoring of the in-memory page cache and page size in user space.
- Successful use cases demonstrated in graph processing, database, metagenomics, and file compression applications.

**Before ECP**
- A proof-of-concept library then called PERMA
  - Focus on NUMA memories
  - Requires kernel modifications and root privileges.

**PowerStack**

**Overview**
- Holistic System Power Management for exascale with production-quality software
  - Kernel-level module for safe access to low-level registers with mmusr
  - Node-level: CPUs, GPUs Memory with a vendor-neutral open-source library, Variorum, which supports
    - Application-level performance optimizations with a task-aware runtime: Intel GEMOP and Conductor, as well as Kokkos support
    - Power-aware resource management and scheduling with SLURM and Flux
    - Large-scale power telemetry with LDMS
    - HPC PowerStack Initiative: Community-wide and international effort with industrial partners (Intel, AMD, IBM, NVIDIA, ARM), academic partners, and national labs.

**Impact**
- Improved performance and energy efficiency across facilities, as well as node-level, with users involved in the process
  - Facilities can make user workloads more energy-efficient and performant
  - Users can make flexible improvements to their use of compute nodes

**Before ECP**
- Sparse efforts existed with mmusr and ibmrnr, which were Intel-specific implementations
  - Power-aware scheduling prototype based on SLURM simulator
  - Runtime system prototype for optimization which was a research code, early version of Intel GEMOP

**Now**
- Production-quality power management software at all levels, ranging from the node-level all the way through system resource managers, across 15 architectures.
- Variorum integrations with Caliper, Kokkos, LDMS, Flux, Intel GEMOP allowing users and administrators to manage power easily at various levels in a vendor-neutral manner.
- Power management of large-scale workflows.

**Future**
- Policies to mitigate power swings to be integrated into resource managers
  - Optimization of science workflows and dependency graphs using integration with workload managers
- Power management with elastic scheduling
- Additional support for upcoming architectures and performance counters.

---

**Argo: Then and Now**

ANL: Pete Beckman (PI), Idbris Daoudi, Rinku Gupta, Kamil Iskra, John-Luke Navarro, Swann Perarnau, John Tramm, Brice Videau, Kazutoshi Yoshiki, LLNL: Maya Gokhale (co-PI), Eric Green, Keita Iwabuchi, Roger Pearce, Ivy Peng, Abhik Sarkar; Tapasya Patki (co-PI), Stephanie Brink, Aniruddha Marathe, Barry Rountree, Kathleen Shoga

University of Arizona: David Lowenthal and team

https://web.cels.anl.gov/projects/argo/

Developing vendor-neutral, open-source software for OS/R improvements.