

Welcome to ALMOS Project

ALMOS (Advanced Locality Management Operating System) is a research operating system currently under development at the [?SoC](#) department of the [?LIP6](#) Laboratory ([?UPMC Sorbonne Universités](#)). This new research operating system has been started from scratch and it is targeting cc-NUMA multi/many-cores. It is intended to investigate the scalability of the different components of an operating system on a current and future many-cores. ALMOS has been originally developed as a part of the TSAR [?MEDEA+ European project](#) and its kernel allows a shared-memory parallel applications to scale up to +1024 cores on the [?TSAR](#) (Tera-Scale ARchitecture) single-chip cc-NUMA many-core. Work is in progress to port ALMOS to AMD Opteron (Interlagos) 64-cores cc-NUMA machine.

The design of ALMOS Kernel

Design of ALMOS kernel In a cc-NUMA architecture, the locality of memory access impacts directly both the scalability and the power consumption (energy by moved bit). The main challenge is to enforce the locality of memory access made by threads of parallel applications. Although the locality enforcing needs a fine management of hardware resources (mainly cores and physical memory), ALMOS aims to hide the hardware topology and the management of its resources to user's applications. This allows POSIX shared-memory parallel applications as well as legacy applications to benefit from the performances offered by a many-core.

The design of ALMOS kernel has a distributed approach articulated around objects named cluster-managers. A cluster-manager is a manager of hardware resources of a hardware NUMA node (mainly cores and physical memory). A cluster-manager can locally supply its homed threads with any type of memory-objects including physical pages and kernel-level objects. A cluster-manger contains a per-core mangers each of which has its own events manger and a multi-policies scheduler server. In this distributed scheme, the kernel has no notion of a global resources state. Instead it has a decentralized approach to discover from where to allocate the requested resources in respect to the locality of memory access. This decentralized mechanism and policies are implemented in the DQDT (Distributed Quaternary Decision Tree). The DQDT is a wait-free mechanism based on resources usage indicators and integrates the locality awareness in all kernel decisions regarding to the tasks/threads placement, memory allocation and cores load-balancing. It constitutes a common kernel-level framework to build strategies to manage the resources of other kernel sub-systems like files and I/O.

The kernel of ALMOS replicates its code in each cluster by creating a per-cluster kernel-process. The threads of a cluster-manager are attached to its local kernel-process. The kernel of ALMOS has a hybrid notion of threads which is based on a new organization of processes virtual address space. This new organization enables the kernel to replicate both the pages-tables and the application 's code and therefore enforcing the locality of the per-core TLBs and I-cache misses.

PROJECT WEBSITE IS UNDER CONSTRUCTION