Welcome to ALMOS Project

1. 1. Outline of ALMOS's Kernel Design
   2. Publications
   3. Getting ALMOS
   4. Running ALMOS on the TSAR many-core

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 is currently used in several research projects and Master degree courses at LIP6 and UPMC. Its development is currently a work-package in the SHARP (Scalable Heterogeneous ARchitecture for Processing) MEDEA+ European project (?#CA109) which is an HPC oriented project leaded by Bull. ALMOS has been originally developed as a part of the TSAR MEDEA+ European project (?#2A718) 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.

Outline of ALMOS's Kernel Design

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 as well as kernel-level threads. 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. The goal is to reduce the contention on the ressources and to increase the locality of the kernel's memory access when refering to its data-structres. 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 data-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 global-state notion of resources. 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 its related policies are implemented in the DQDT (Distributed Quaternary Decision Tree). The DQDT is a wait-free mechanism based on a set of 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 in order 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. ALMOS threading model is compatible with PThreads standard and it allows the kernel to provides a native support to PGAS (Partitioned Global Address Space) programming model found in HPC oriented languages like UPC (Unified Parallel C).

Publications

- Ghassan Almaless and Franck Wajsburt. On The Scalability of Image and Signal Processing Parallel Applications on Emerging cc-NUMA Many-cores. In *Proceedings of the international conference on the Design and Architectures for Signal and Image Processing*, Karlsruhe, Germany, IEEE, 2012. [to appear].

- Ghassan Almaless and Franck Wajsburt. On the Impact of Many-Cores Small Caches on the Scalability of Shared-Memory Highly Multi-Threaded Single-Applications". In *Posters of the 3rd Asia-Pacific Workshop on Systems*, Seoul, South Korea, 2012.

- Ghassan Almaless and Franck Wajsburt. Does Shared-Memory, Highly Multi-Threaded, Single-Application Scale on Many-Cores? In *Proceedings of the 4th USENIX Workshop on Hot Topics in Parallelism*, Berkeley, CA, 2012. ?paper.

- Ghassan Almaless. ALMOS : un système d'exploitation pour manycores en mémoire partagée cohérente. In *Proceedings of the 8th French conference on Operating Systems (CFSE), the French chapter of ACM-SIGOPS, GDR ARP*, Saint-Malo, France, 2011. ?paper.

- Ghassan Almaless and Franck Wajsburt. ALMOS: Advanced Locality Management Operating System for cc-NUMA Many-Cores?. In *Proceedings of the 5th national seminar of GDR SoC-SIP*, Lyon, France, 2011. ?paper.

Getting ALMOS

You can get ALMOS source code from its GIT repository:

```
git clone https://www-soc.lip6.fr/git/almos.git
```

The current code hosted by this repository runs only on the TSAR simulator. The available version of ALMOS provides multi-processes and multi-threads support. The latest version of ALMOS will be soon tracked by this repository.

Running ALMOS on the TSAR many-core

For convenient reasons, a stand-alone and ready-to-use distribution of ALMOS targeting the TSAR virtual prototype is available. Please refer to this documentation page to get started and to run your applications on ALMOS/TSAR target.