Welcome to ALMOS Project

ALMOS (Advanced Locality Management Operating System) is an open-source 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 is also deeply used in ?TSUNAMY - an ANR national research project aiming to co-design a new security solution for information processing on multi/many-core processors. ALMOS has been originally developed as a part of the TSAR MEDEA+ European project (?#2A718).

ALMOS allows several shared-memory parallel applications to scale up to +1024 cores on the ?TSAR (Tera-Scale ARchitecture) single-chip cc-NUMA many-core.

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 the 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 resources and to increase the locality of the kernel's memory access when referring to its own data structures. A cluster-manager is a manager of hardware resources of a cc-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-manager contains a per-core managers. Each core-manager has its own events-manager and a multi-policies scheduler server. In this distributed scheme, the kernel has no notion of resources global state. Instead, it has a decentralized mechanism for discovering from where to allocate the requested resources, taking in account the locality of memory access of the requester thread. This decentralized mechanism and its related policies are built using the DQDT (Distributed Quaternary Decision Tree). The DQDT is a distributed wait-free infrastructure based on a set of resource usage indicators and integrates the locality awareness needed by all kernel decisions regarding the tasks/threads placement, memory allocation and cores load-balancing. It constitutes a common framework for building 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 their local kernel process of their cluster. 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 of the page tables and the application's code; 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 provide a native support to PGAS (Partitioned Global Address Space) programming model found in HPC oriented languages like UPC (Unified Parallel C).

Publications

- Ghassan Almaless. Operating System Design and Implementation for Single-Chip cc-NUMA Many-Core. Ph.D thesis, UPMC, France, 2014. Abstract, Résumé, Thesis FR, Slides FR.

- 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. Paper.

- 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. Poster.

- 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 Operating System on TSAR Single-Chip cc-NUMA Many-Core. Seminar, ALSoC/LIP6, 2012. Slides.

- 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.

Available User-land Libraries

ALMOS comes with the following user-land libraries:

- System
  - ♦ C library based on ?dietlibc.
  - ♦ PThreads support library.

- Parallel Programming
  - ♦ GNU OpenMP ?libgomp.
  - ♦ Standford Map&Reduce ?Phoenix.

- Math & Compression
  - ♦ Sun Microsystems standard math library ?fdlibm.
  - ♦ General purpose compression library ?zlib.

- Cryptography *[experimental]*
  - ♦ Some cryptography functions from the OpenSSL project ?libcrypto.

- Graphics
    - A very small implementation of a subset of OpenGL for embedded systems or games ?TinyGL.

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 Tutorial page to get started and to run your applications on ALMOS/TSAR target.

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.

History & Contributors

A brief history of ALMOS development and a list of contributors can be found in HistoryAndContributors.

Contact Information

You may want to subscribe to the ?almos-users mailing list. If you have any idea or proposal about the deployment of ALMOS or a potential joint work, please feel free to write to Ghassan Almaless (first.last@lip6.fr) the designer of ALMOS and the project leader.