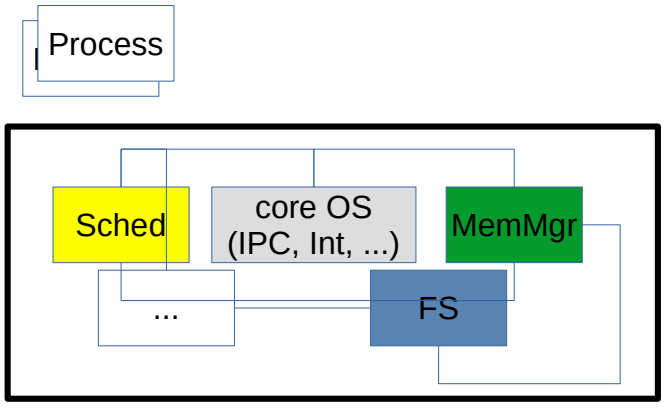
LIG's PDS/HPDA research projects

Alain Tchana (Grenoble INP, France),
Renaud Lachaize (Univ. Grenoble Alpes, France)

alain.tchana@grenoble-inp.fr

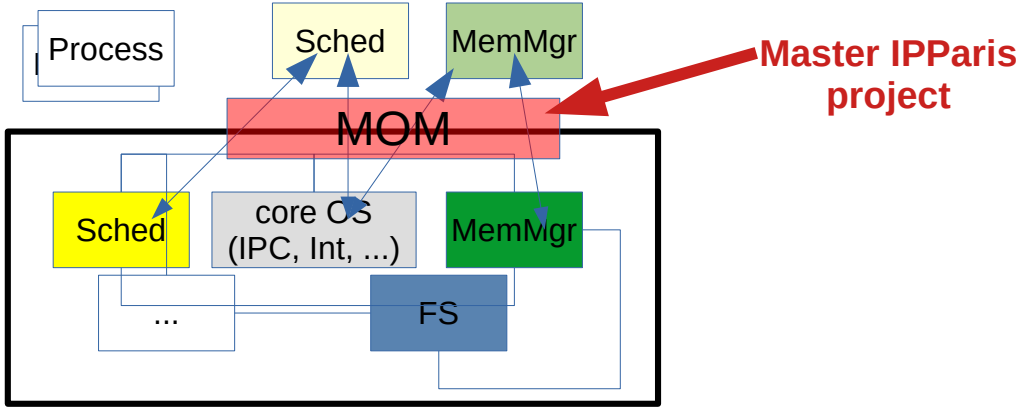
September 2023

Micro-Linux Kernel: Building a Message-Oriented Middleware for Kernel space – User space communication



Monolithic Linux

Custorm OS policies, no machine reboot, no kernel programing, in Python:), …..

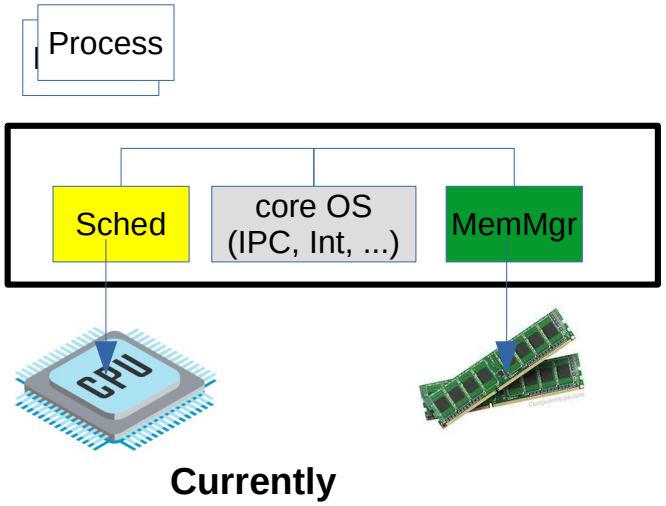


Microkernalization of Linux

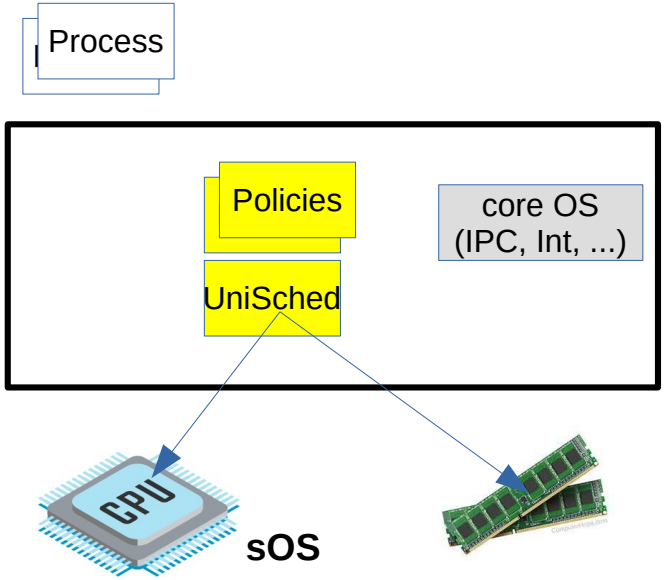
Scheduler : ghOSt SOSP'21
FS : uFS SOSP'21, Bento, FAST'21
IO : Snap, SOSP'19

MemMgr + holistic approach : In progress, ERODS (Grenoble)

sOS: OS's memory manager is actually a scheduler, so let's treat it this way for greater efficiency and velocity



Currently



sOS

The similarity between processor management and memory management:

- RAM is a collection of physical page (PP) frames <-> The CPU package is a collection of cores.
- Virtual pages (VP) virtualize physical pages <-> Threads virtualize physical cores.
- Memory allocator maps VP to PP <-> The scheduler maps threads to cores.
- Swapping in/out <-> Context switching.
- A thread can release memory (e.g., madvise MADV_DONTNEED) <-> A process can yield.
- Page fault <-> Tick timer.
- VP pinning to a NUMA node <-> CPU affinity.

XOS: the end of the reign of the process-thread duo

- Existing **mainstream OS abstractions** for separation (threads, processes+IPC) are **insufficient for modern applications**.

- **Newer research proposals** for first-class OS abstractions **are proliferating** **but** are **not converging** and **do not compose well**.

- The **rigid abstractions** exported by OS APIs are **leaking into application/library code**, which impedes their potential for (static and dynamic) adaptation.

xOS: Goals & approach

We aim at **designing and building an OS prototype** (“xOS”), through the following steps:

- Define **central “pivot” OS abstractions** for expressing separation concerns + communication model.
- Refine & optimize **mapping of existing abstractions** to the pivot model, with appropriate toolchain and OS support.
- Support **assisted/automated choice of adequate isolation** abstraction based on high-level guidelines and constraints.
- Support **fully dynamic switch** between different separation abstractions and communication channels.

xOS: The VIC architecture