Memento
Energy-Efficient Memory Placement

Timo Hönig, Ruhr-Universität Bochum
Andreas Polze, Hasso-Plattner-Institut
cs-os-memento@lists.ruhr-uni-bochum.de
https://informatik.rub.de/boss/research/memento/

Starting Point

- Novel memory technologies (NVM, HBM, GDDR, etc.) bring new hardware diversification, enabling new application areas in the software. [1]
- Established resource abstractions in system software or programming models largely hide unique characteristics (e.g., persistence guarantees, low power demand), preventing optimisation strategies on application or system level.
- Details should no longer be completely hidden but raised to a meaningful context for application development or system resource management.
- The Memento project aims to address shortcomings of the current state of the art with methods for analysing both program code at development time, its operational characteristics at runtime, as well as characteristics of memory resources at system setup time.

Memento Notation

- Understanding a workload's memory-access behaviour is vital for placement decisions. [2]
- Alongside static and dynamic analysis, programmers may wish to supply functional and non-functional requirements.
- We propose persisting the memory-access requirements in an abstract ***Mem**ory Be**h**aviour **N**otati**o**n* (Memento).
- Our API design builds on our preceding research with NUMA-aware data placement. [3]

```
// uninitialised fields default to 0.f,
// identifying irrelevant requirements
MemRequirement LatencyReq {
    weight_latency = 3.f,
    weight_randomAccess = 1.f };

// allocations will be associated with LatencyReq
MemGuard guard(LatencyReq);
Index *jumpListA = new Index [4096];
Index *jumpListB = new Index [8192];
```

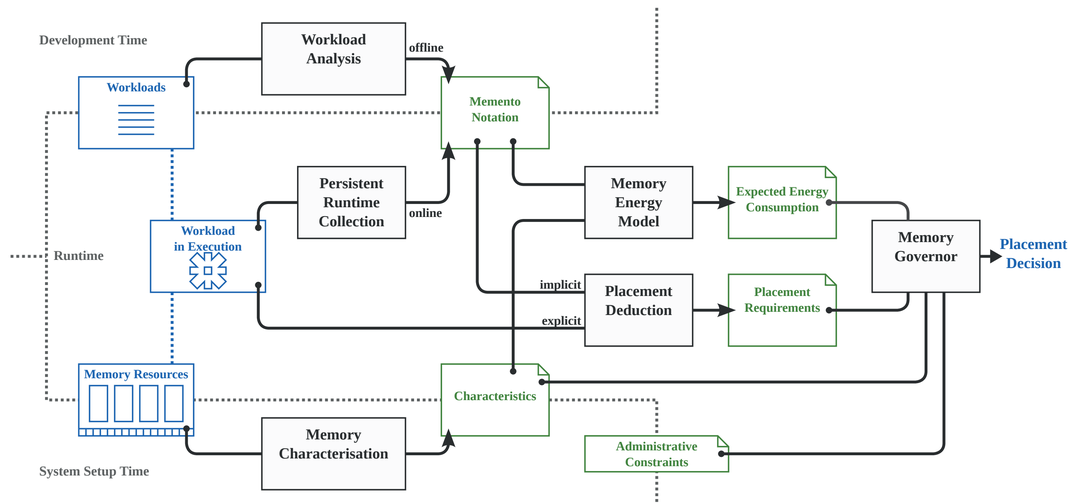
Envisioned API for specifying desired memory characteristics explicitly.

Memory Governor

- The Memento *Memory Governor* automatically makes energy-efficient memory-placement decisions.
- As part of the OS, it has a complete overview of the system's software and hardware and can implement and *enforce* system-wide placement strategies.
- It considers resource requirements, hardware characteristics, estimated energy demand, and administrative constraints.

Project Objectives

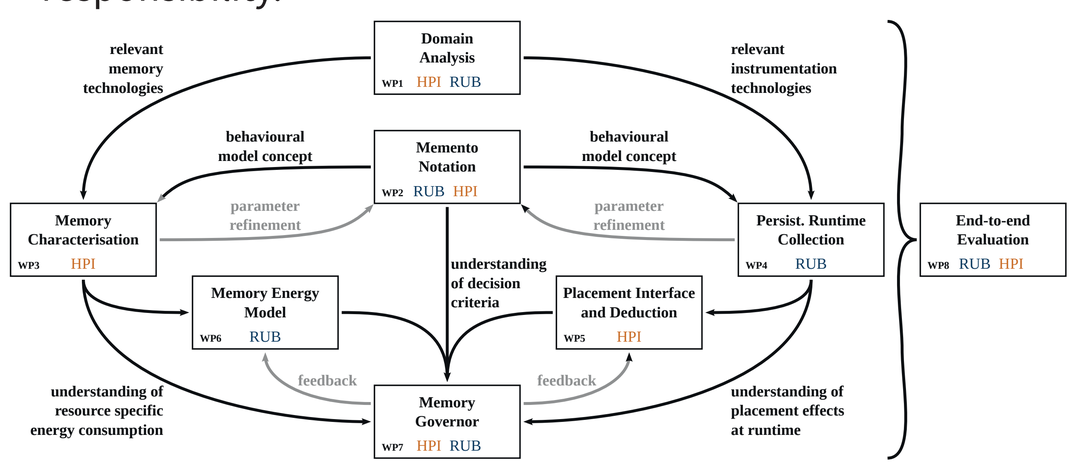
- Investigate and develop new tools, system-software designs, and evaluation metrics
- Appropriate characterisation of hardware and software properties
- Model hard- and software's memory energy demand
- Implement memory-placement interfaces
- Make automatically energy-efficient placement decisions



Structure of the Memento approach: proceeding from a system environment (blue) is a set of tools/mechanisms (black) leads to optimised placement decisions. Intermediate artefacts are denoted in green.

Project Structure and Interdependencies

- The work program consists of eight work packages, focussing on the different expertise of both groups involved.
- Essential work packages (WP1-2, 7-8) have shared responsibility.



Logical dependencies and interfaces between work packages. Grey backward edges indicate cyclic dependencies, where findings in one work package trigger new insights in another.

References

[1] C. Eichler, H. Hofmeier, S. Reif, T. Hönig, J. Nolte, and W. Schröder-Preikschat. Neverlast: An NVM-centric operating system for persistent edge systems. In *Proceedings of the 12th ACM SIGOPS Asia-Pacific Workshop on Systems (APSys'21)*, 2021.

[2] T. Patel, A. Wagenhäuser, C. Eibel, T. Hönig, T. Zeiser, and D. Tiwari. What does Power Consumption Behavior of HPC Jobs Reveal? In *Proceedings of the 34th IEEE International Parallel and Distributed Processing Symposium (IPDPS'20), pages 799–809*, 2020.

[3] W. Hagen, M. Plauth, F. Eberhardt, F. Feinbube, and A. Polze. PGASUS: a framework for C++ application development on NUMA architectures. In *Proceedings of the Fourth IEEE International Symposium on Computing and Networking (CANDAR'16)*, pages 368–374, 2016.

funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 502228341