Aakash Saini(2018CSM1001)

May 7, 2019

- **Observations:** Paper proposes **Helios**, an OS to simplify the task of writing, deploying, and tuning applications for heterogeneous platforms. The important part of Helios are ***satellite kernel***, which export a single, uniform set of OS abstractions across CPUs of disparate architectures and performance characteristics. Helios uses **positive platform affinity** values to call the list of possible satellite kernels that are eligible to host a particular process. I/O services are accessed using the files which are made transparent via *remote message passing* and which are extending a standard micro kernel message-passing abstraction to satellite kernel infrastructure. Helios avoids the unnecessary goals by exporting a single, unified namespace. The *namespace* serves multiple purposes. Helios retargets applications to available ISAs by compiling from an intermediate language. For the simplification of deployment and tuning of application performance, an affinity metric for the developers is exposed by Helios. A hint to the OS is provided using the affinity for deciding whether a process would benefit from executing on the same platform as a service it depends upon. Addition of CPU and DRAM to the satellite kernels require three basic hardware primitives: a timer, an interrupt controller, and the ability to catch an exception (i.e., trap). Without these primitives, Helios could not implement basic services, such as scheduling, directly on the programmable device. Authors developed the satellite kernels for an XScale programmable I/Ocard and for cache-coherent **NUMA** architectures. Helios achieves up to a 28% performance improvement by offloading tasks to the XScale I/Ocard. On a mail-server benchmark, 39% improvement in performance was achieved by splitting an application into multiple NUMA domains.

- **Conclusions:** Helios is an OS designed for heterogeneous systems and multi-programming environment. It uses the satellite kernels for simplifying the programming environment. by providing the affinity between the deploying and performance tuning applications. Helios extends satellite kernels to NUMA architectures, treating NUMA as a shared nothing multiprocessor. Helios removes the kernel as a bottleneck to scaling up performance in large multiprocessor systems.

- **Limitations:** Although satellite kernels proved to be useful tools, their deployment has been limited to only one programmable device which is a problem. For the proposed OS there is no compiler which can create DirectX textures for interrupts. In a heterogeneous environment, the placement of application scan have a drastic impact on performance.

- **Future work:** In future processes can be allowed to span NUMA domains by moving CPUs and memory between satellite kernel. Helios can be ported to Intel Larrabee graphics card and benefits of the satellite kernels to an application also can be measured. Helios can be ported to the programmable device such as the GPU. Also, a compiler can be designed to create DirectX textures from CIL, for providing the interrupts which are required by the satellite kernels.