Predicting and Alleviating Bottlenecks in Hybrid CPU and Memory Architectures

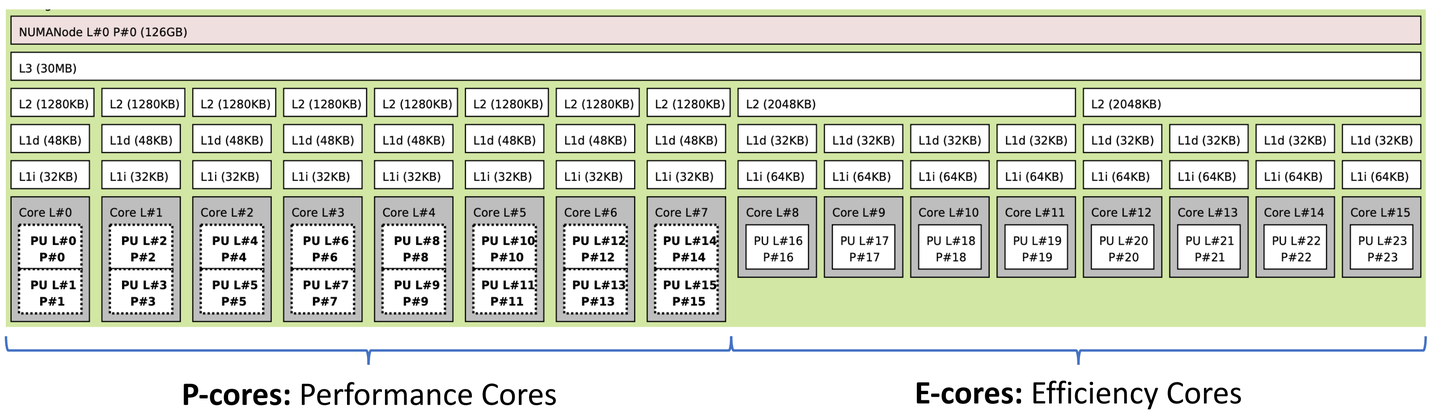
Nanda Velugoti, Kyle Hale [IIT]
Joseph Manzano, Nathan Tallent [PNNL]
nvelugoti@hawk.iit.edu

Hybrid CPU Architectures

In HPC, computing architectures usually tend to be homogeneous in nature and are specifically designed to be highly parallel and closely connect clusters of compute units. This is because in the HPC applications benefit from homogeneous systems with minimal OS noise (data parallelism, frequent bulk synchronization).

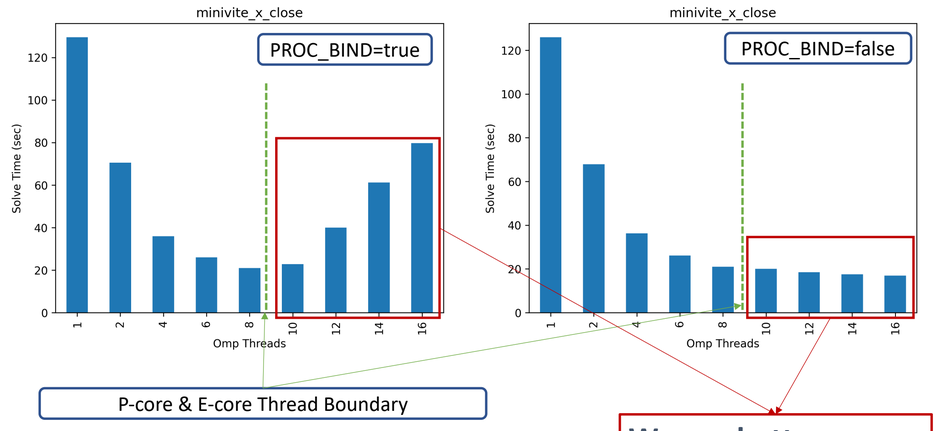
Hybrid architectures have been explored in *embedded and mobile devices* in the past. However, with the release of new Intel's P/E-core CPUs, **hybrid architectures have entered desktop/cloud/server devices**.

Output of Linux's *lstopo:* Alderlake Architecture 12th Gen Intel(R) Core(TM) i9-12900KF



P-cores: Performance Cores **E-cores:** Efficiency Cores

Impact of Hybrid CPUs



Testbed: 12th Gen Intel(R) Core(TM) i9-12900KF
Workload: *miniVite* (multi-threaded graph community detection)
Runtime Configuration:

- OMP settings: 16 threads,
- Disable hyper-threading using OMP_PLACES
- OMP_PROC_BIND=true/false

We see better scaling when threads are not bound to cores, i.e., thread imbalance in parallel workloads scale better in hybrid core architectures.

Capturing Power Profiles

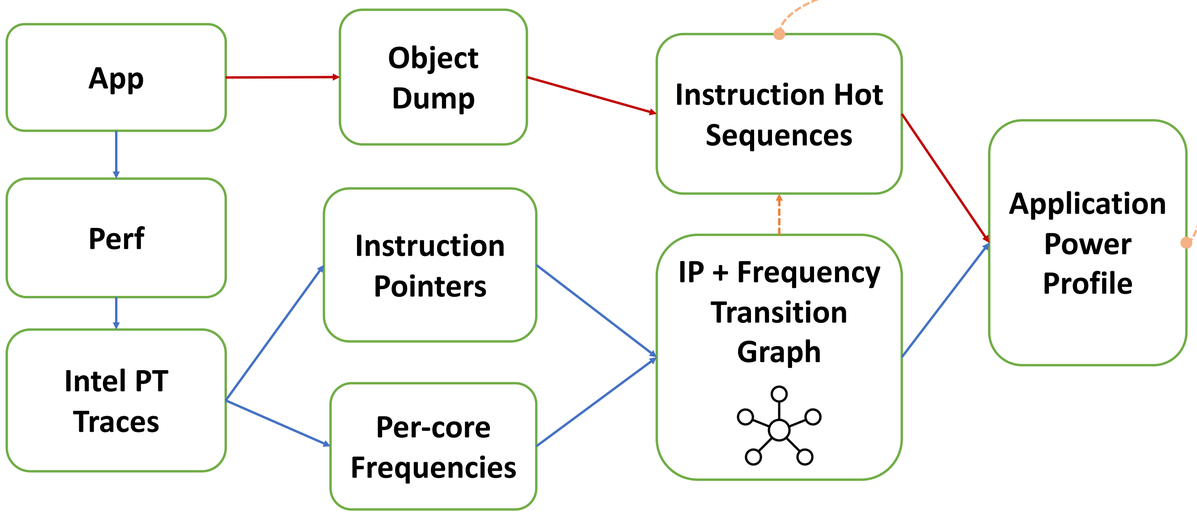
Apart from CPU/memory effects, we also wanted to study power characteristics of various parallel workloads, as power efficiency is the main selling point of hybrid architectures.

IP + Frequency Transition Graphs: These are essentially a form of (directed) state transition graphs, where the **vertex represents a particular Instruction Pointer** and **edges represents several frequency changes** occurred between two IP vertices. Here, the change in frequency can be either to higher value ("up") or to a lower value ("down").

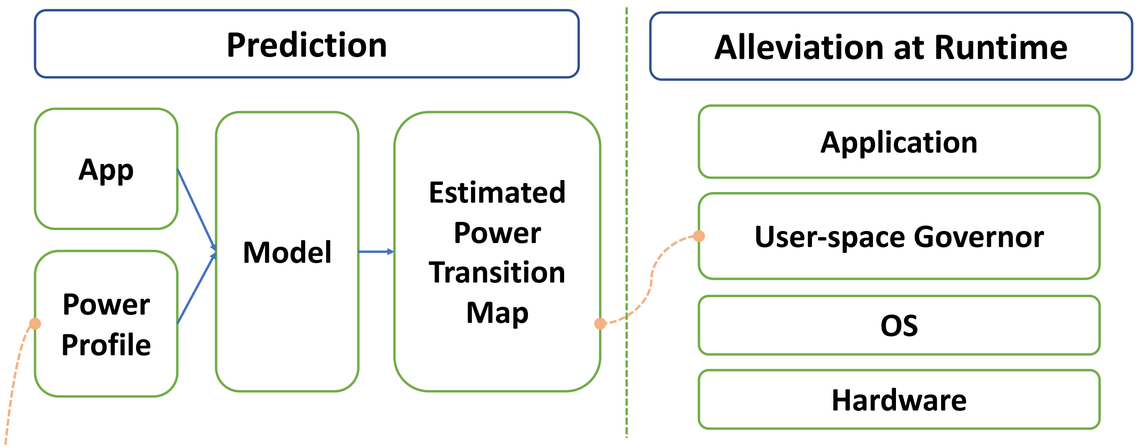
Instruction Hot Sequences: These are the sequence of instructions in the application binary (assembly) that are correlated with IP and frequency values from transition graphs.

Example of an Instruction Hot Sequence:

```
...
test
cmp xx xx
mov xx xx
Cmp xx xx
je
...
```



Current Work: Predict and Alleviate



In our current efforts, we are exploring on how we could use the constructed power profiles of a parallel application and to better inform the runtime on when to make the power/frequency transitions.

- **Prediction Model:** The goal here is to properly parse the profile information and identify the **"hot zones"** of the app and output an optimal (yet estimated) power/frequency transition map particular to that app. We can follow some approaches in MemGaze[1] tool
- **Alleviation Process:** We use the resulting optimal power transition map, to **explicitly change** the power/frequency of a core during the application runtime. To do this, we can use DVFS (Dynamic Voltage and Frequency Scaling) features[2].

Summary/Takeaways

In summary,

- We introduce hybrid architectures and their importance in executing parallel workloads.
- We show the impacts of hybrid CPUs on parallel workloads.
  - Parallel applications with **work imbalance → better (strong) scaling** when **threads are not pinned to cores**.
- We present an approach to capture power characteristics of an application
  - IP + Frequency **Transition Graphs → to represent app's power/frequency utilization** over time.
  - Instruction **Hot Sequences → to represent "hot zones",** i.e., which part of app is causing such power/ frequency changes.
- We then present our current work on prediction and alleviation approaches
  - Utilize the profiled information to **predict optimal power transition pattern.**
  - Implement a user-space power **governor to explicitly enforce the optimal pattern at runtime.**

Related Work

[1] Kilic, Ozgur O., et al. "MemGaze: Rapid and Effective Load-Level Memory Trace Analysis." *2022 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2022.

[2] Wamhoff, Jons-Tobias, et al. "The turbo diaries: Application-controlled frequency scaling explained." *USENIX Annual Technical Conference*. 2014.

Acknowledgement(s): This work has been supported and sponsored by Pacific Northwest National Laboratory (operated by BATTELLE MEMORIAL INSTITUTE), Richland, WA 99354.