

# Performance Measurement on Heterogeneous Processors with PAPI

Willow Cunningham, **Vince Weaver**

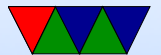
[vincent.weaver@maine.edu](mailto:vincent.weaver@maine.edu)



ProTools Workshop — 18 November 2024

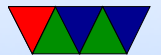
# Hopefully you've heard of PAPI

- Performance API
- Open-source cross-platform performance measuring tool nearing 25 years old
- Unlike most other tools allows “caliper” measurements in source code
- Currently does not support heterogeneous processors



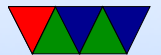
# What are Heterogeneous Processors?

- Also called Hybrid
- Multi-core processors where the cores are not identical
  - ARM has had this for years: big.LITTLE
  - Intel since Alder Lake has P-cores (performance) and E-cores (efficiency)
- **Note, this is not the same as CPU/GPU heterogeneous computing**



# Why have Heterogeneous Processors?

- Usually for power reasons:  
When not doing much, move jobs to the “efficient” cores and shut down the rest
- If a program needs a boost, move it to the “power” core

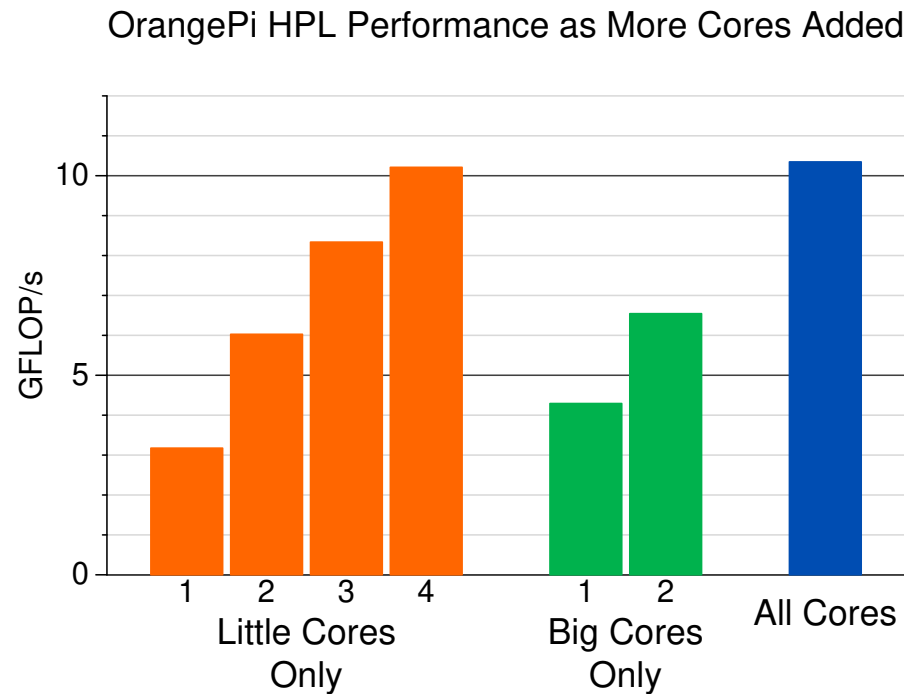


# Complications

- Scheduling: when to move processes between cores?  
Who decides? (it's hard enough to schedule identical cores)
- Differing capabilities: the cores are often different models of chips (E-cores are essentially Intel Atom)
  - E-cores on Raptor Lake do not support top-down events
  - E-cores do not support AVX512



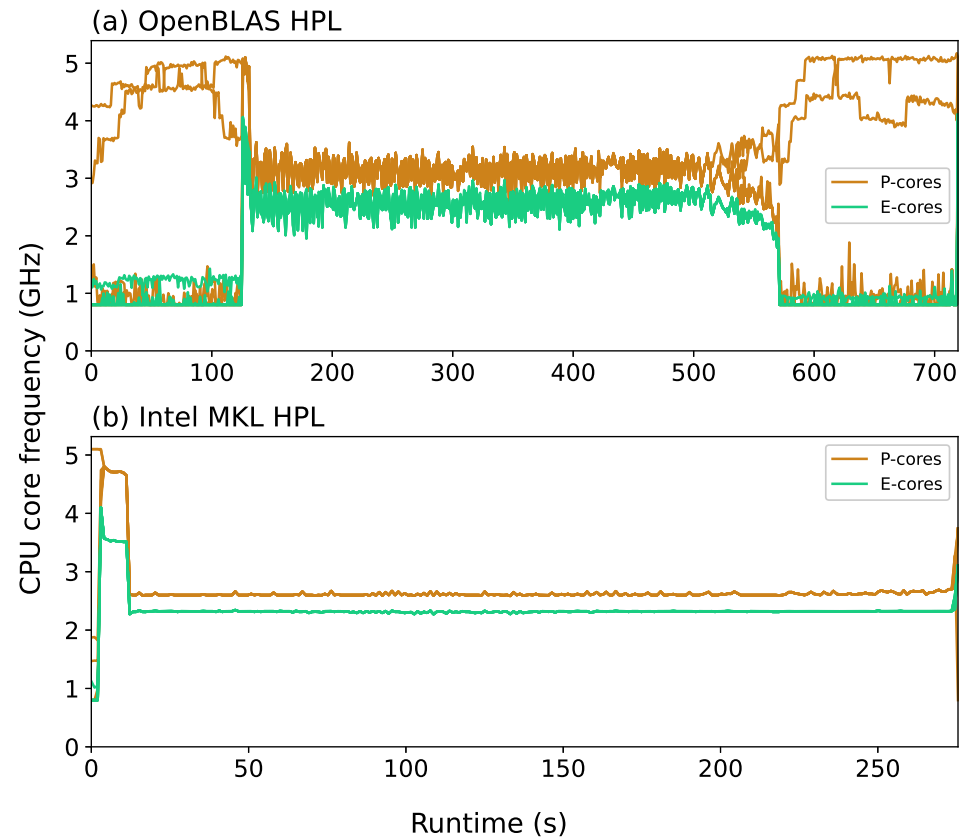
# Motivation: ARM big.LITTLE (A72/A53)



The problem here is big cores are throttled down due to overheating.



# Motivation: Intel Raptor Lake (8/16P, 8E)



Power throttling is involved, but also differences in LLC cache misses.

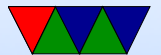
OpenBLAS best: P-core only, 356 GFLOPs

IntelMKL best: P+E, 457 GFLOPs



# Implementation: How does perf work?

- Has separate PMUs for each type of core
- If you open an “instructions” event, opens one event for each type of core
- Linux will only enable the appropriate event as necessary
- When done you have two counts, one for each core type
- Seems like a reasonable thing to implement in PAPI



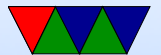


# Challenge: Detecting Heterogeneous Processors

- There are lots of ways to do this on Linux
- None of them work in all cases
- Intel makes this especially difficult

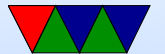
*We probably ought to publish a table of native model numbers. . . but I don't know if there are plans to do that. Likely that Wikipedia will get to that before Intel does.*

*– Tony Luck (Intel), linux-kernel, 21 October 2024*



# Challenge: libpfm4 support

- PAPI uses libpfm4 for event names
- Helped test and get hybrid support into libpfm4
- Intel Alder/Raptor Lake support there
- ARM big.LITTLE nearly there (required re-write)



# Challenge, PAPI changes

- PAPI perf\_event component only allows adding events from one perf-pmu at a time
- Need to change to allow a PAPI EventSet to contain multiple perf-groups so that multiple perf-pmus can be measured as one single EventSet
- There are a lot of deep seated assumptions in the PAPI code that make this difficult, but we do have some working prototype code
- Ongoing work to avoid overhead regressions



# Biggest Challenge, Presenting Transparently to Users

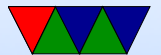
- If a user adds PAPI\_TOT\_INS on a heterogeneous machine, what should they get?
- Should it be combination of P and E cores?
- Should we force user to specify separate events?  
PAPI\_TOT\_INS:P? But then what on normal machine?
- Should they be forced to use native events on hybrid?

`adl_glc::INST_RETIRED:ANY, adl_grt::INST_RETIRED:ANY`



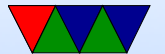
# Other Challenges: Processor Detection

- There are other parts of PAPI that can't express hybrid CPUs well
- `PAPI_get_hardware_info(void);`
- The sysdetect component



# Future Work

- Make sure no regression in overhead
- Make sure whatever we decide on is forward-thinking.  
For example, Some ARM processors already have three levels of hybrid



# Questions?

`vincent.weaver@maine.edu`  
`https://web.eece.maine.edu/~vweaver/`

Also can come talk to us at the poster session

Supported by NSF Grant No. CSSI-2311709

