Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Towards Integrated Performance & Power Modeling

NATHAN R. TALLENT, RYAN D. FRIESE, GOKCEN KESTOR, ANDRÉS MÁRQUEZ
ROBERTO GIOIOSA, DANIEL CHAVARRÍA-MIRANDA, DARREN KERBYSON

Pacific Northwest National Lab

Workshop on Modeling & Simulation of Systems & Applications

August 12, 2016

Tool Overview & Vision

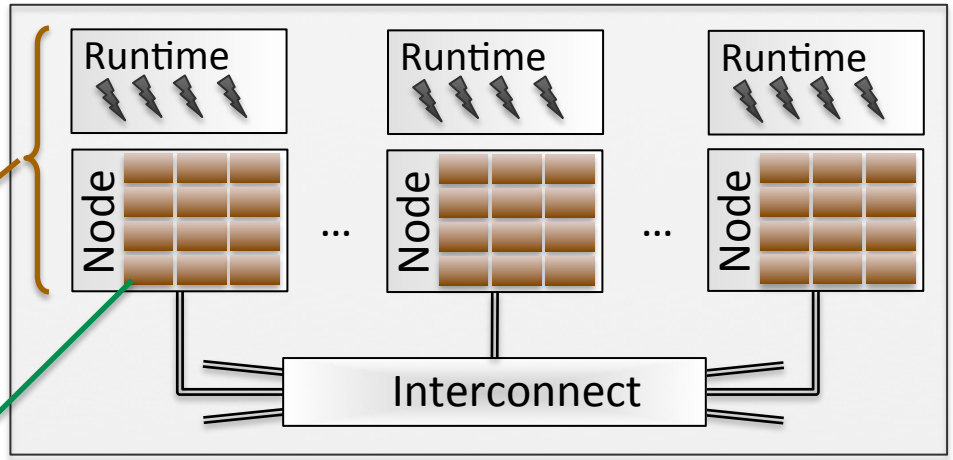
Palm generates performance models for an application and its subcomponents.

```
program nekbone
    !$pal model init
    call init_dim, call init_mesh, …
    !$pal mode
    call cg(…)
end
```

```
subroutine cg(…)
    !$pal loop n_cg = ${n_iter}
    do iter=1,n_iter
        …
    enddo
```

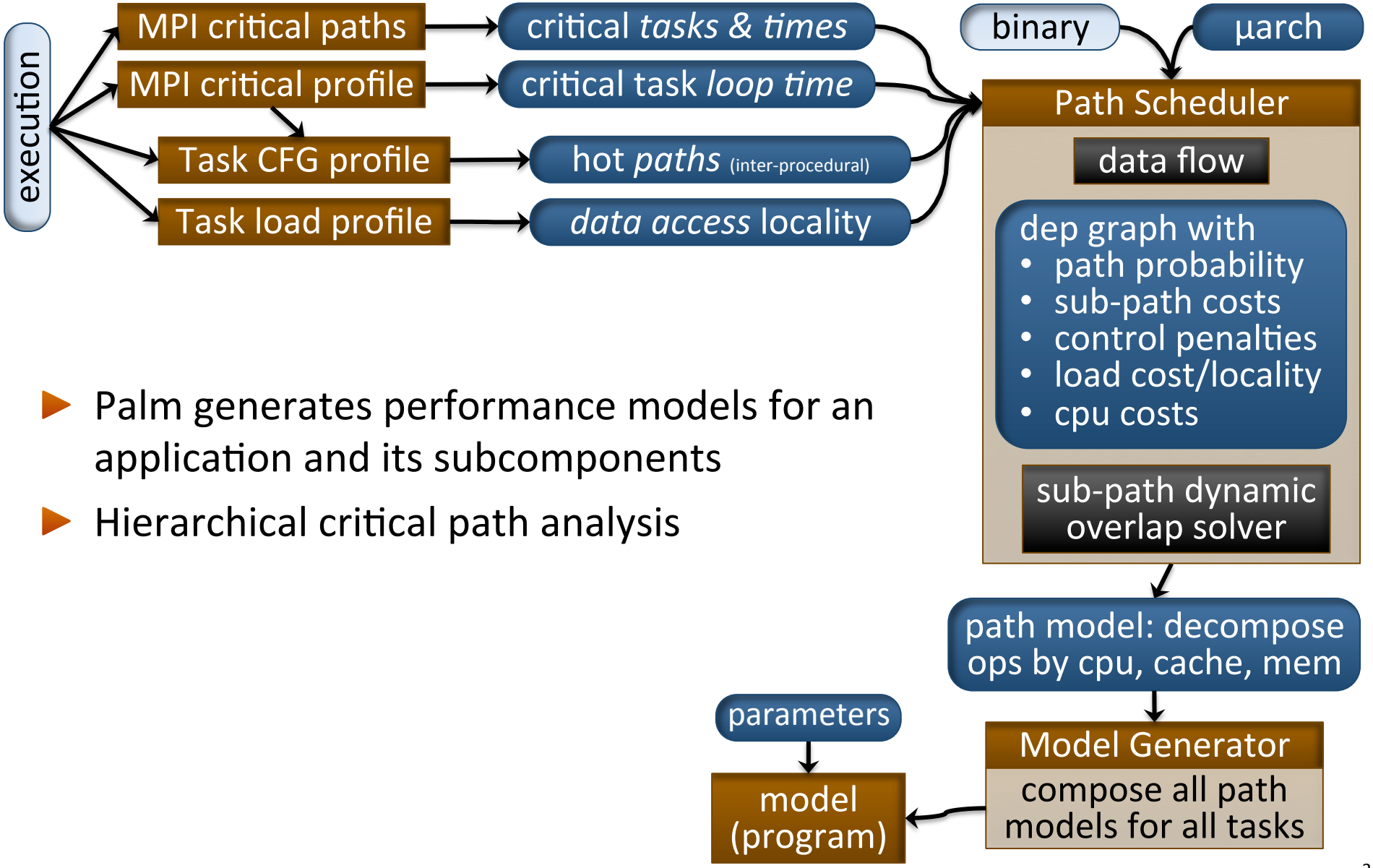
```
void halo_exchange(buf[n], n…)
    #pragma pal loop n_send = ${n}[max]
    for(i = 0; I < n; ++i)
        isend(…, buf[i]…);
```

Prometheus explores the effects of task-based scheduling under different concurrency, task-placement, and platform scenarios.



P-McPAT provides detailed power predictions for novel technology and architecture pairings.

Palm: Generate Application Performance Models



- Palm generates performance models for an application and its subcomponents
- Hierarchical critical path analysis

Results: Strong scaling models of irregular tasks

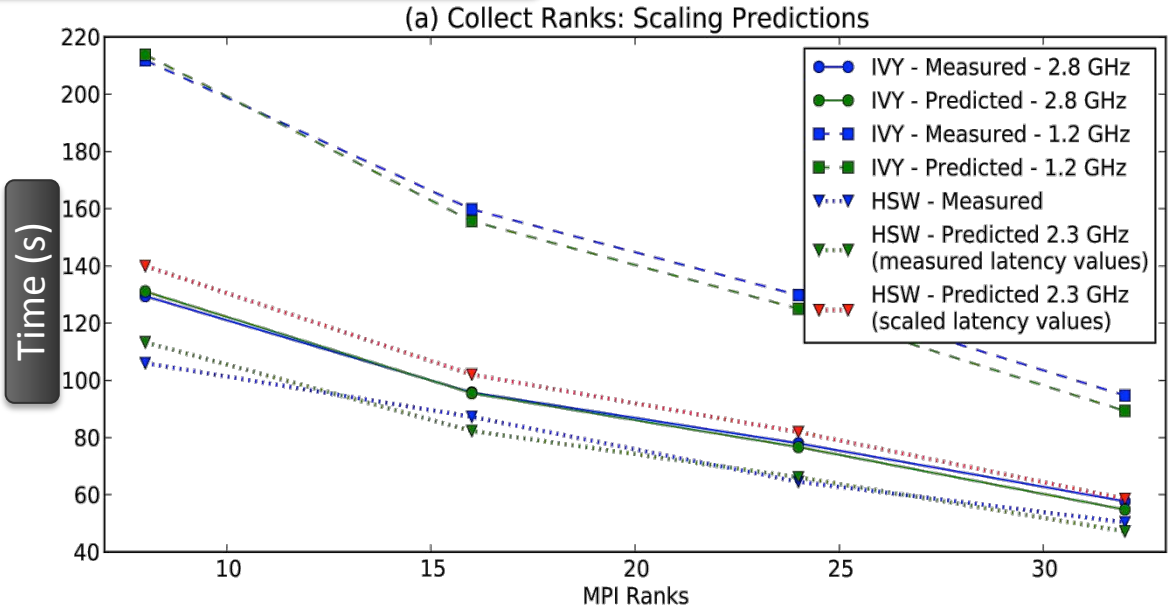
- Generate task models at *one* scaling configuration
  - IvyBridge, 2.8 GHz, DDR3-1866 [Intel E5-2680 v2]
  - Smallest number of ranks only
- Collect MPI critical task parameters at several scaling points
- Predict strong scaling sequence
  - IvyBridge, 2.8 GHz, DDR3-1866
  - IvyBridge, 1.2 GHz, DDR3-1866
  - Haswell, 2.3 GHz, DDR4-2133 [Intel E5-2698 v3]

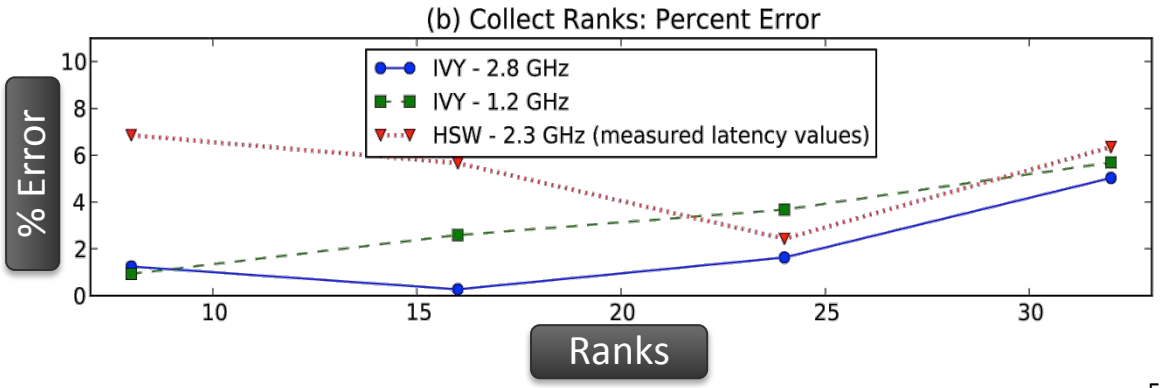
PageRank's Key Tasks: Strong Scaling

- MPI implementation of Page Rank
- Power-law graph as input (11 M vertices; 1.3 B edges)
- Load imbalance

Challenges: 'CollectRanks'
- aggregates communication
- inlined code: C++ map lookup, insert, iterate
- unbiased branches, indirect data accesses
- calls: hash, new/delete

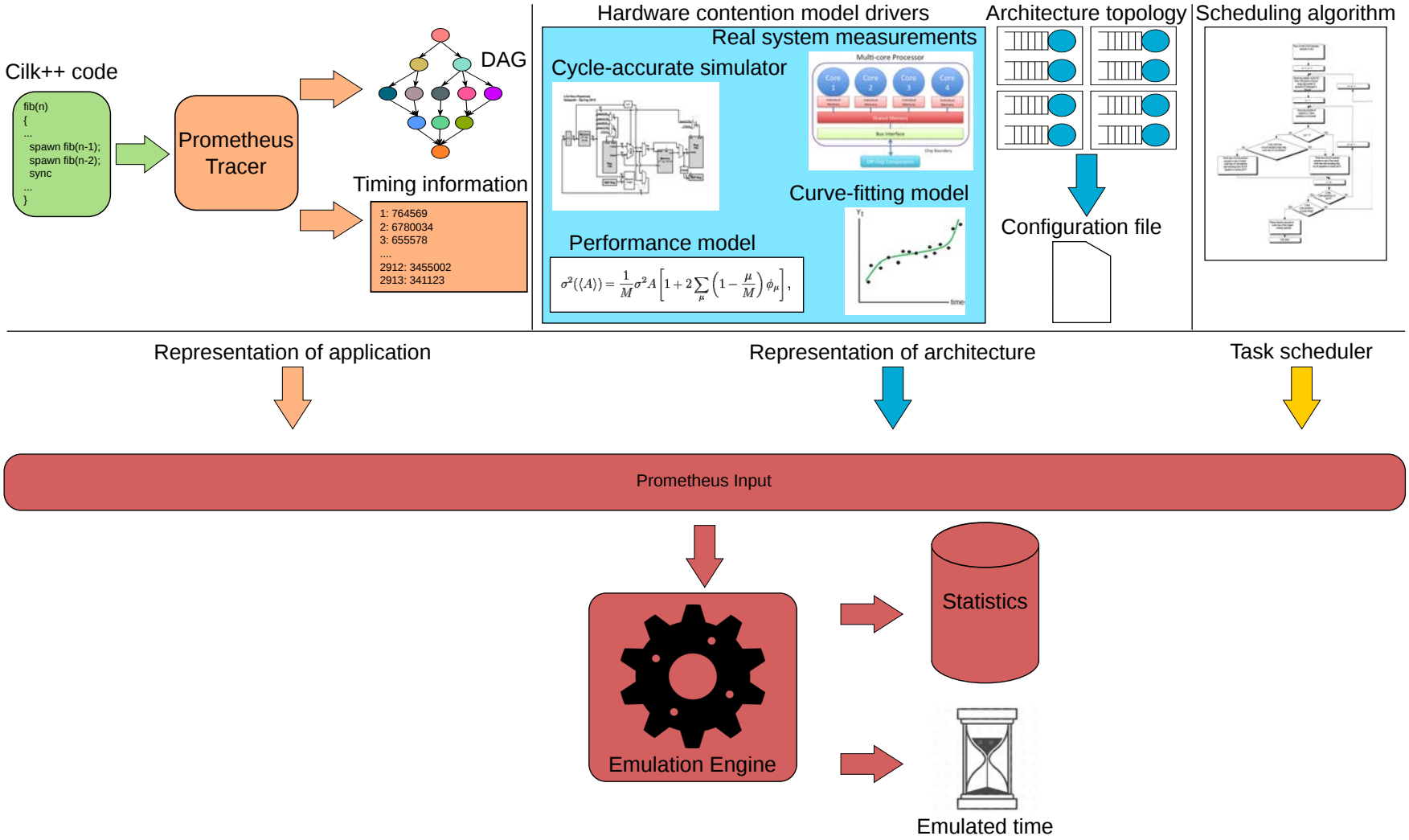
On Haswell, re-generate data access parameter values





Prometheus: Effects of Task-based Scheduling

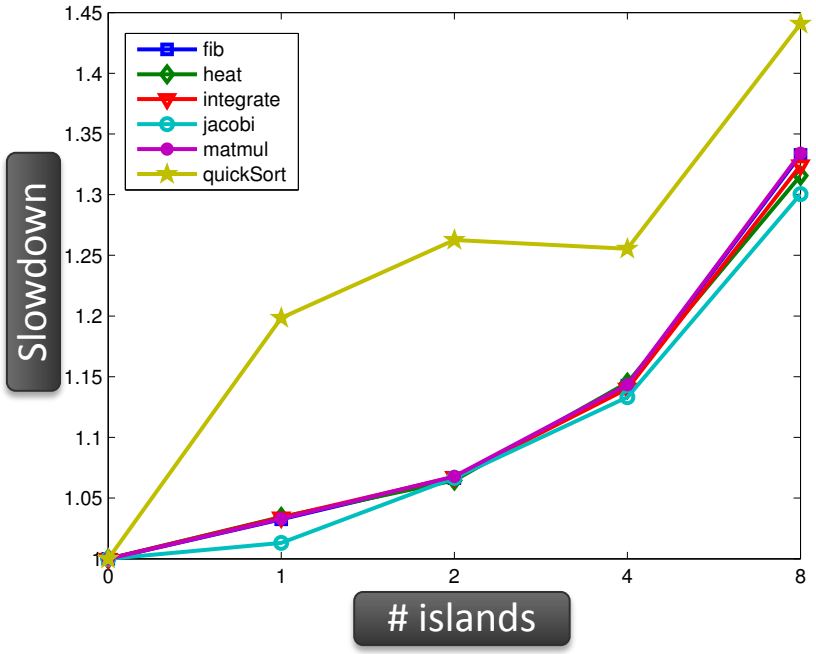
Prometheus explores the effects of task-based scheduling under different concurrency, task-placement, and platform scenarios



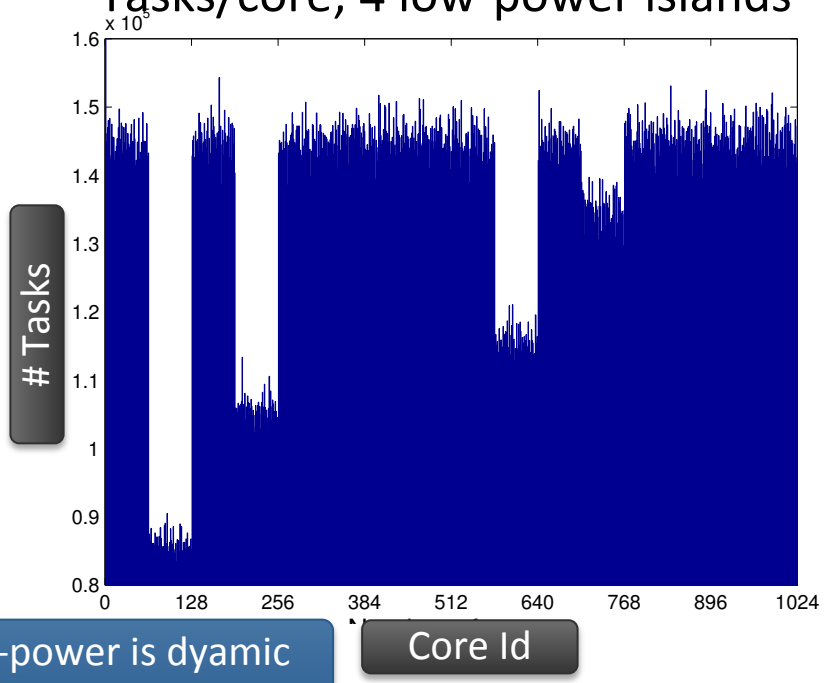
Case study: Power-constrained systems

- Emulate heterogeneous, power-constrained exascale systems
  - 1,024 total cores, 16 voltage islands, 64 cores/island
  - Vary the number of voltage islands in low-power mode
  - Low-power mode cores run at ½ max frequency
- Automatic task balancing contains performance degradation

Slowdown with *n* low-islands



Tasks/core; 4 low-power islands



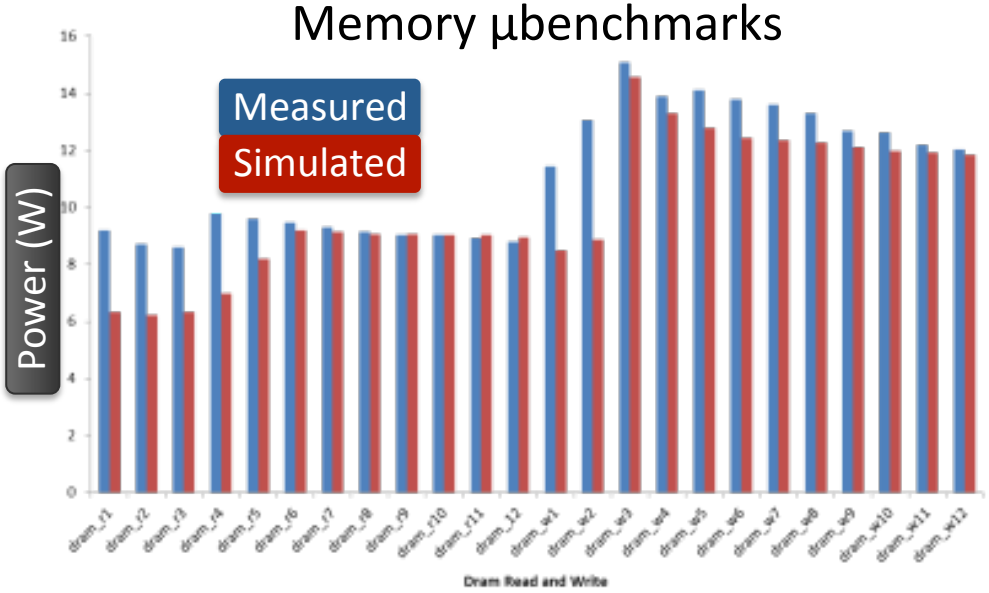
- Low-power is dyamic
- Tasks of different sizes

P-McPAT: Node-level Power Modeling

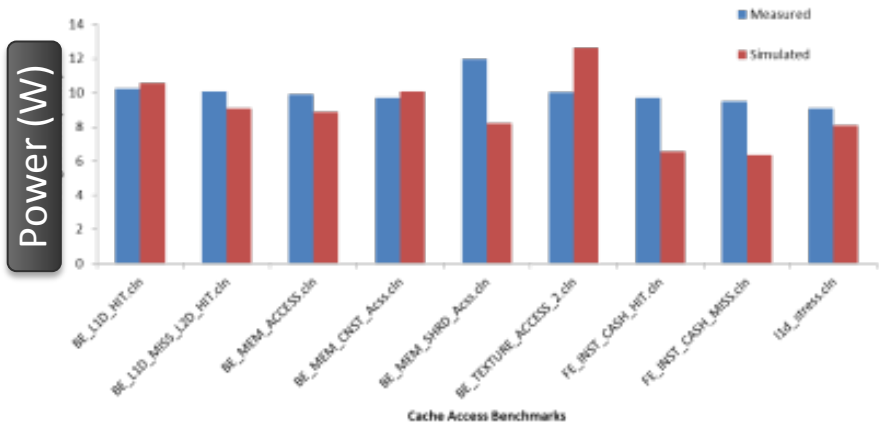
- Detailed power predictions of novel technology & architecture
  - e.g., GPU Kayla running at 7nm NTV
- Works well with leading performance simulation tools
  - Gem5: LOCs like Alpha, ARM, SPARC, MIPS, POWER, x86
  - GPGPU-SIM: TOCs like NVIDIA
- Mature: CACTI → McPAT → P-McPAT
  - CACTI, FinCACTI: Technology and array layout/routing modeling
  - McPAT: Multicore Power, Area, and Timing
    - Logic and architecture modeling, builds on CACTI
    - orphaned after HP ends support
  - PNNL Enhancements
    - New technology nodes: 7nm FinFet
    - New operation modes: STV (super) and NTV (near)

P-McPAT Validation: NVidia Kepler

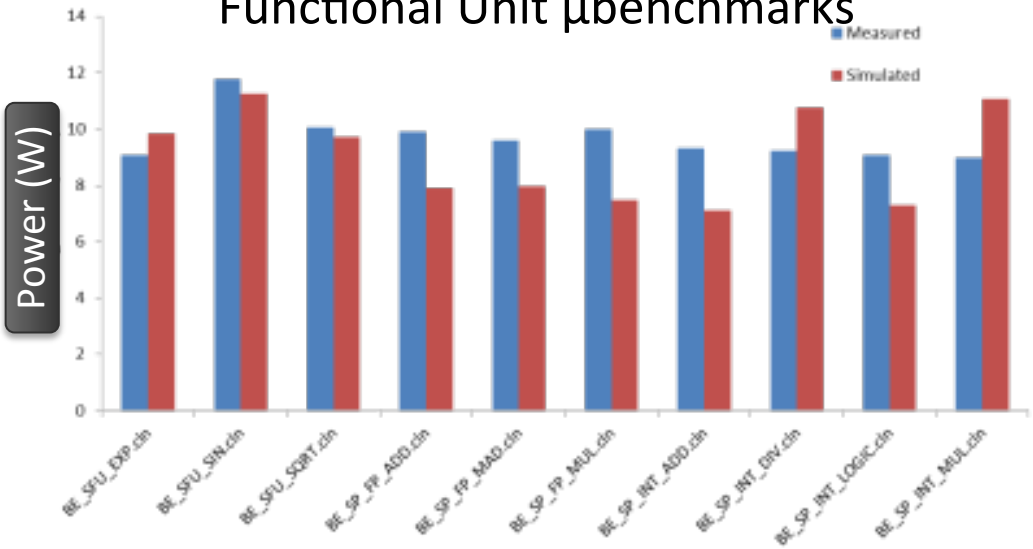
Memory µbenchmarks



Cache µbenchmarks



Functional Unit µbenchmarks



Measured Simulated

Measured vs. Simulated Power (W)				
	Min	Max	Mean	RMSE
Mem	8.6	15.1	10.9	1.6 (16%)
	6.2	14.6	9.7	
FU	8.9	11.8	9.7	1.7 (19%)
	7.1	11.2	8.9	
Cache	9.1	22.9	10.0	2.2 (25%)
	6.3	12.6	8.7	

Integrating Performance and Power Modeling