Towards Integrated Performance & Power Modeling

Nathan R. Tallent, Ryan D. Friese, Gokcen Kestor, Andres Marquez
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.

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.

Program nekbone

```plaintext
!$pal model init
call init_dim, call init_mesh, ...

!$pal model cg(...)
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]...);
```
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.

Cilk++ code

```c
fib(n) {
    ...
    spawn fib(n-1);
    spawn fib(n-2);
    sync;
    ...
}
```

DAG

Timing information

1: 764569
2: 6780034
3: 655578
2912: 3455002
2913: 341123

Performance model

\[ \sigma^2(A) = \frac{1}{M} \sigma^2 \left[ 1 + 2 \sum \left( 1 - \frac{\mu}{M} \right) \delta_i \right] \]

Hardware contention model drivers

Real system measurements

Architecture topology

Scheduling algorithm

Configuration file

Representation of application

Representation of architecture

Task scheduler

Prometheus Input

Emulation Engine

Statistics

Emulated time
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
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

<table>
<thead>
<tr>
<th></th>
<th>Measured</th>
<th>Simulated</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem</td>
<td></td>
<td></td>
<td>8.6</td>
<td>15.1</td>
<td>10.9</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.2</td>
<td>14.6</td>
<td>9.7</td>
<td></td>
</tr>
<tr>
<td>FU</td>
<td></td>
<td></td>
<td>8.9</td>
<td>11.8</td>
<td>9.7</td>
<td>1.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.1</td>
<td>11.2</td>
<td>8.9</td>
<td></td>
</tr>
<tr>
<td>Cache</td>
<td></td>
<td></td>
<td>9.1</td>
<td>22.9</td>
<td>10.0</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.3</td>
<td>12.6</td>
<td>8.7</td>
<td></td>
</tr>
</tbody>
</table>

Measured vs. Simulated Power (W)
Integrating Performance and Power Modeling

MPI critical paths → critical tasks & times
MPI critical profile → critical task loop time
Task CFG profile → hot paths (inter-procedural)
Task load profile → data access locality

Path Scheduler
- data flow
- dep graph with
  - path probability
  - sub-path costs
  - control penalties
  - load cost/locality
  - cpu costs

Model Generator
- sub-path dynamic overlap solver
- path model: decompose ops by cpu, cache, mem
- parameters
- compose all path models for all tasks

Node

P-McPAT power predictions