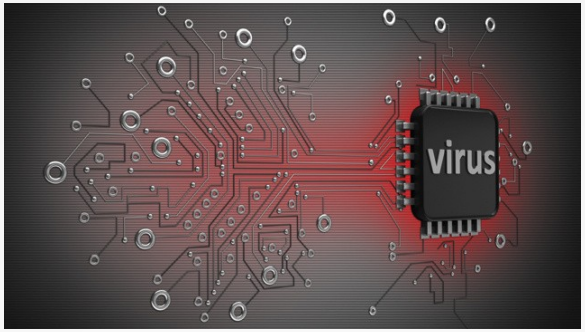
Power Containers



An OS Facility for Fine-Grained Power and Energy Management on Multicore Servers

Kai Shen, Arrvindh Shriraman, Sandhya Dwarkadas, Xiao Zhang, Zhuan Chen

Goals

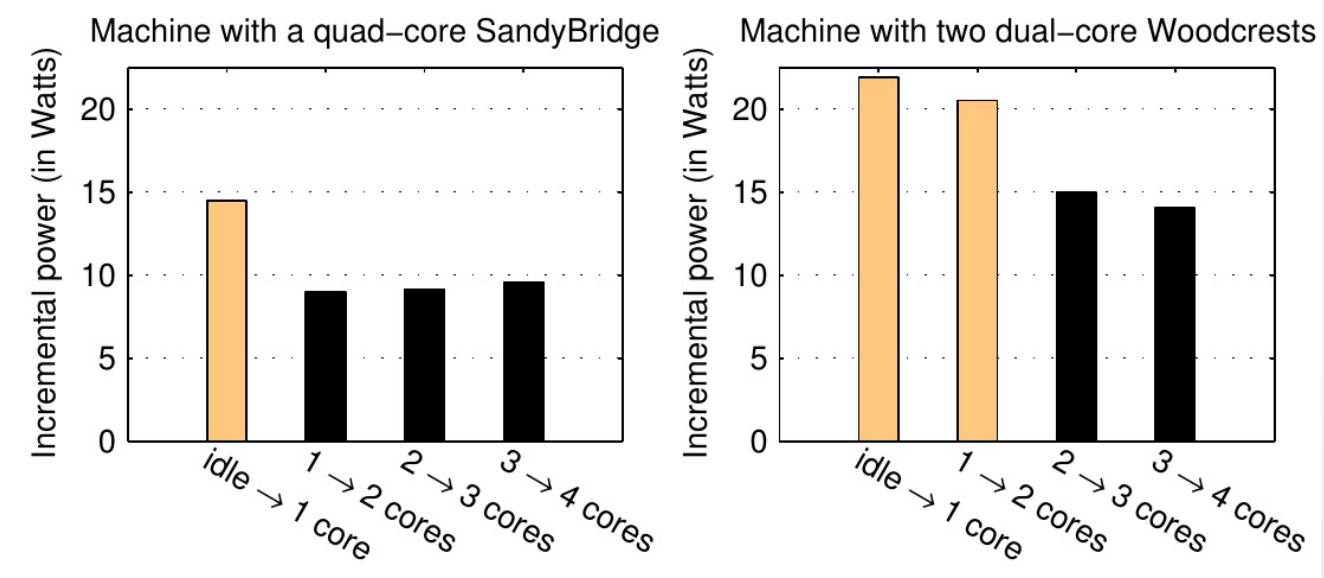
- Isolate power consumption up to individual request level
- Account power based on power containers (budgets)
- Prevent excessive power usage by “power viruses”
- Multicore servers

Related Work / Challenges

- Power modeling usually only for whole system
- High inaccuracies in performance counter based models
- Need expensive/complex calibration equipment
- Limiting individual applications or requests is difficult

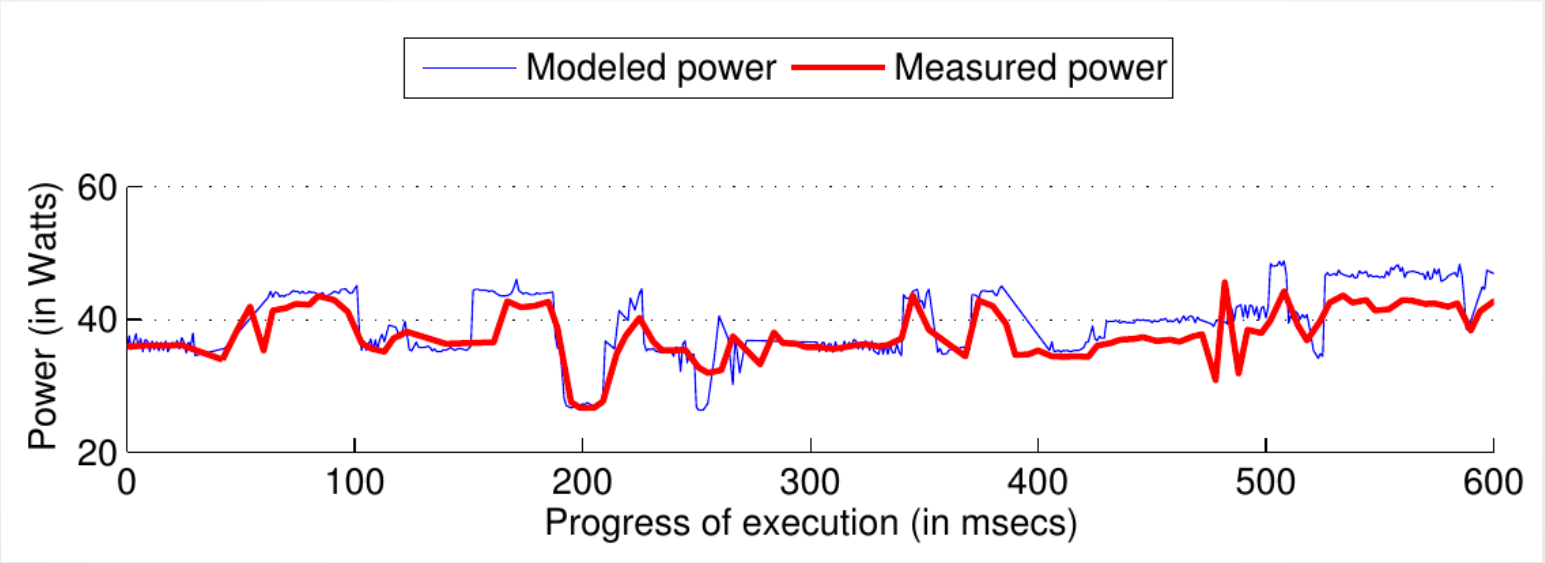
Design: Power Attribution

- Use performance counters for individual applications
- Introduce “chip maintenance power”



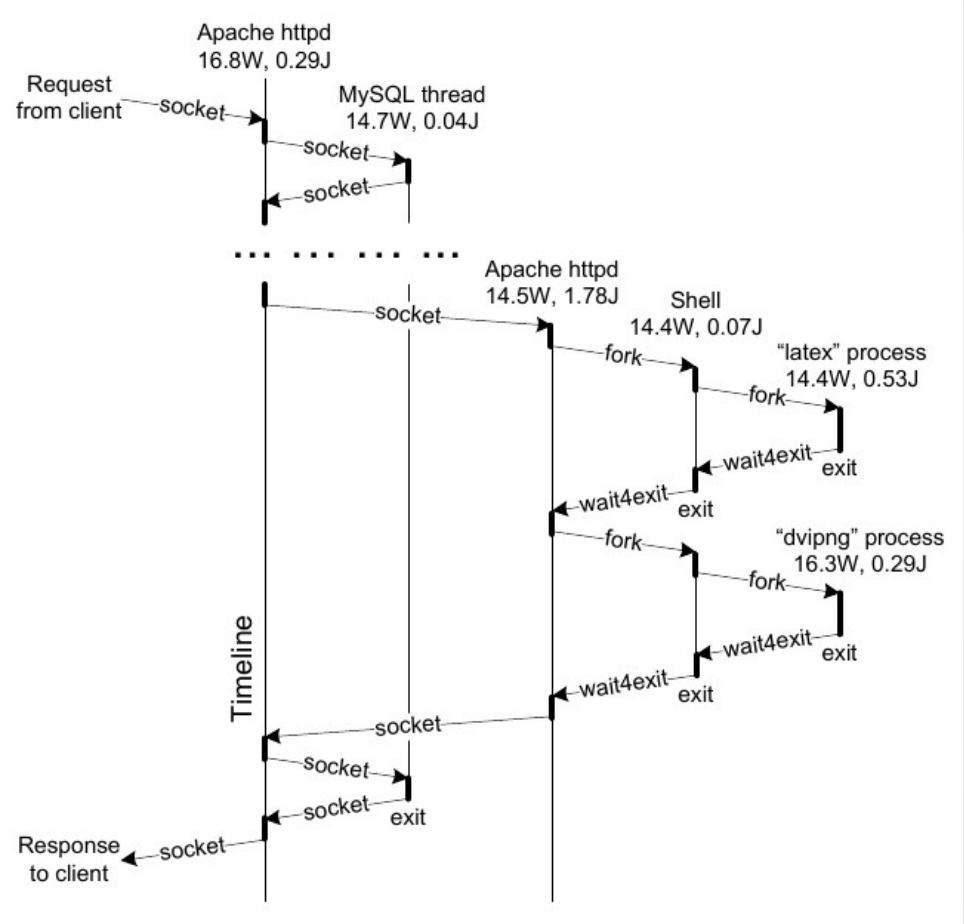
Design: Recalibration

- Modeling inaccuracies can introduce quite some error
- Fix it by on-line recalibration



Design: Request Power Accounting

- Track IPC, forking, socket communication
- Tag socket messages with sender request context for persistent socket connections



Design: Management Possibilities

- Limiting “Power Viruses” by selective duty-cycle modulation
  - Power consumption target per request
- Distribute requests to the most efficient server for this request type in heterogeneous environments
  - enabled by fine grained profiling of requests

Measured active Power

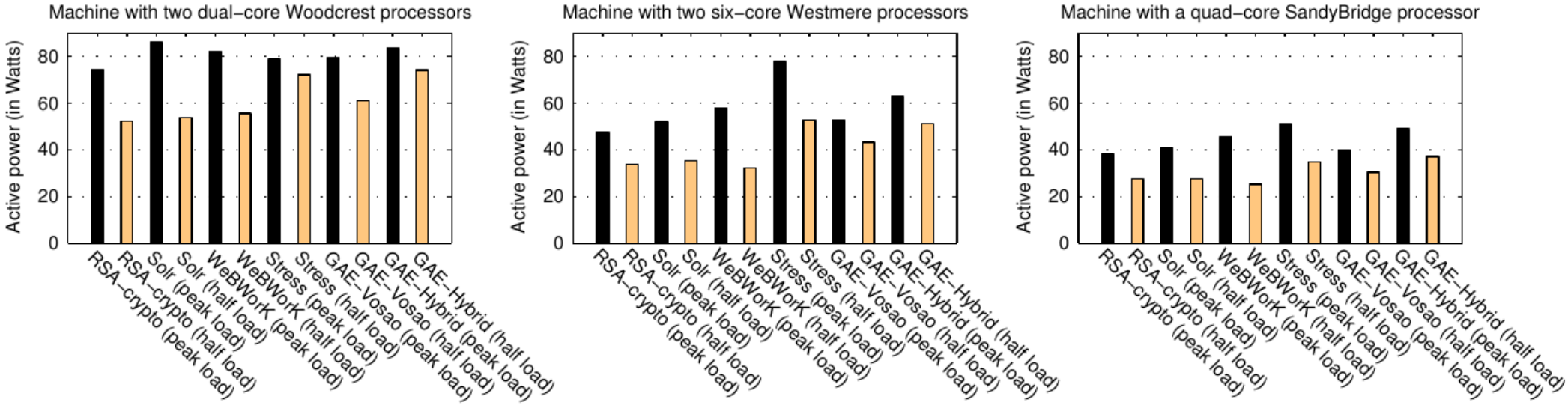
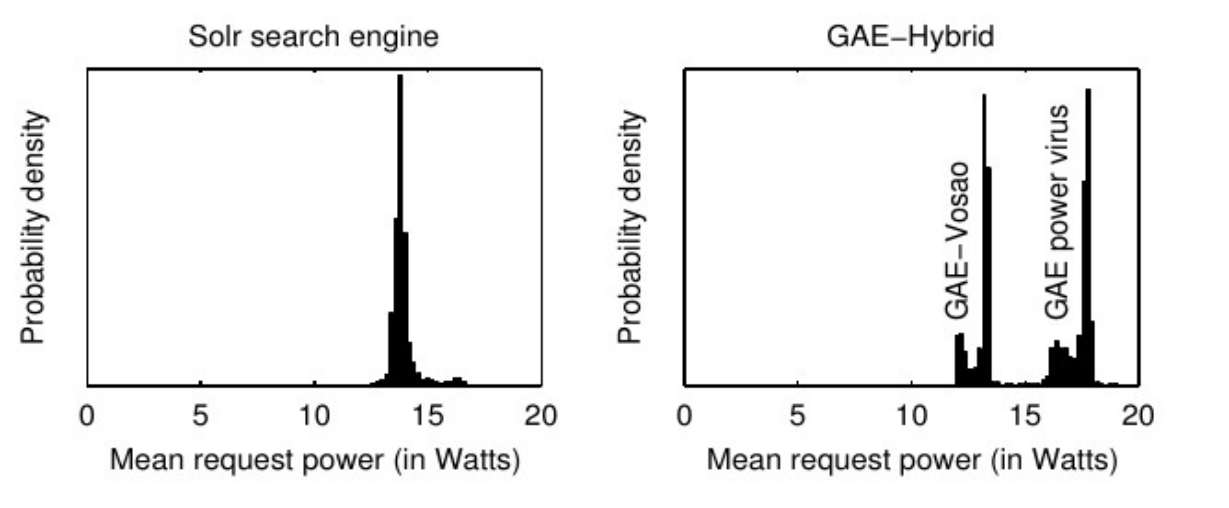
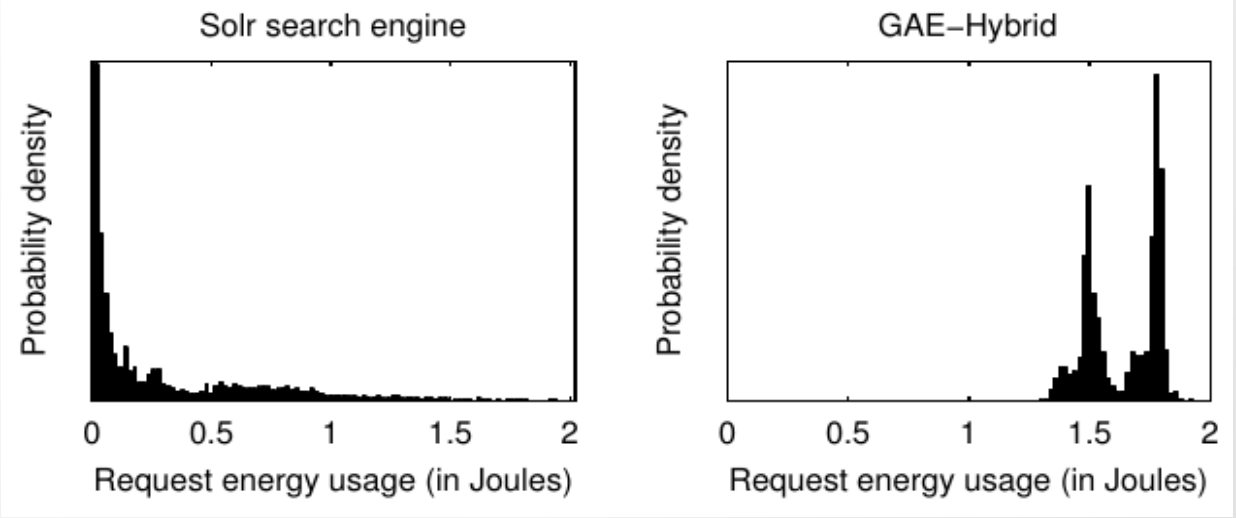


Figure 5. Measured active power of application workloads on three machines and two load levels.

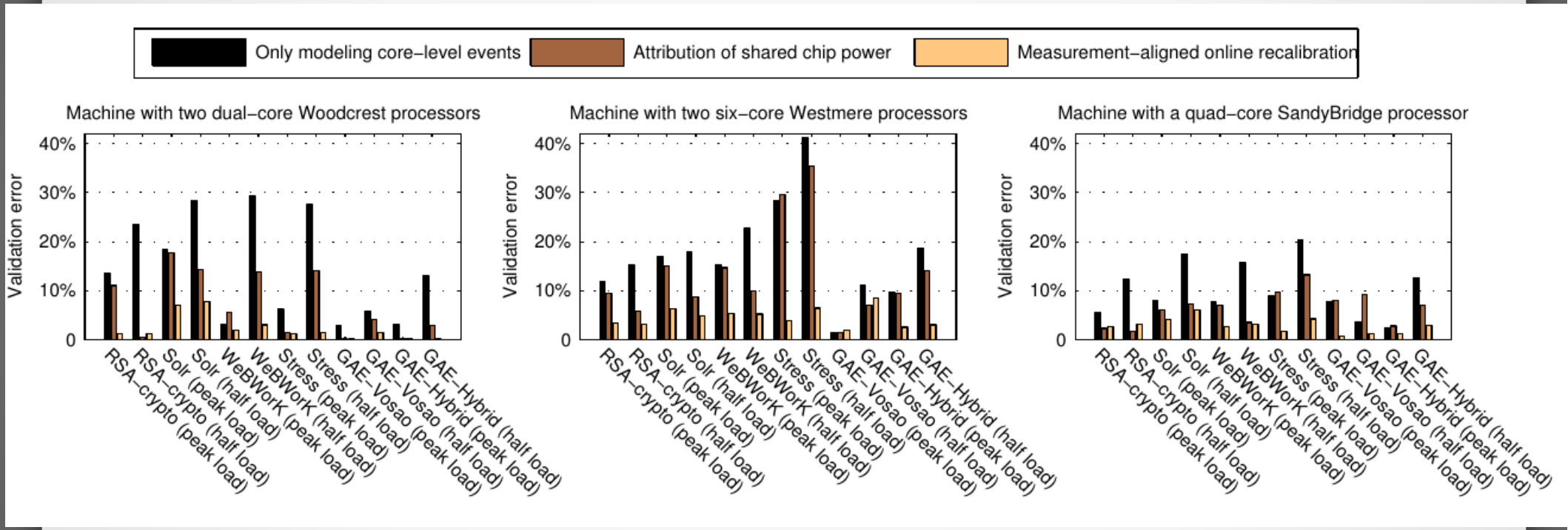
$$
\begin{aligned}
C_{\text{idle}} &= 26.1 \text{ Watts;} \\
C_{\text{core}} \cdot M_{\text{core}}^{\max} &= 33.1 \text{ Watts;} \\
C_{\text{ins}} \cdot M_{\text{ins}}^{\max} &= 12.4 \text{ Watts;} \\
C_{\text{cache}} \cdot M_{\text{cache}}^{\max} &= 13.9 \text{ Watts;} \\
C_{\text{mem}} \cdot M_{\text{mem}}^{\max} &= 8.2 \text{ Watts;} \\
C_{\text{chipshare}} \cdot M_{\text{chipshare}}^{\max} &= 5.6 \text{ Watts;} \\
C_{\text{disk}} \cdot M_{\text{disk}}^{\max} &= 1.7 \text{ Watts;} \\
C_{\text{net}} \cdot M_{\text{net}}^{\max} &= 5.8 \text{ Watts.}
\end{aligned}
$$

Power/Energy distribution

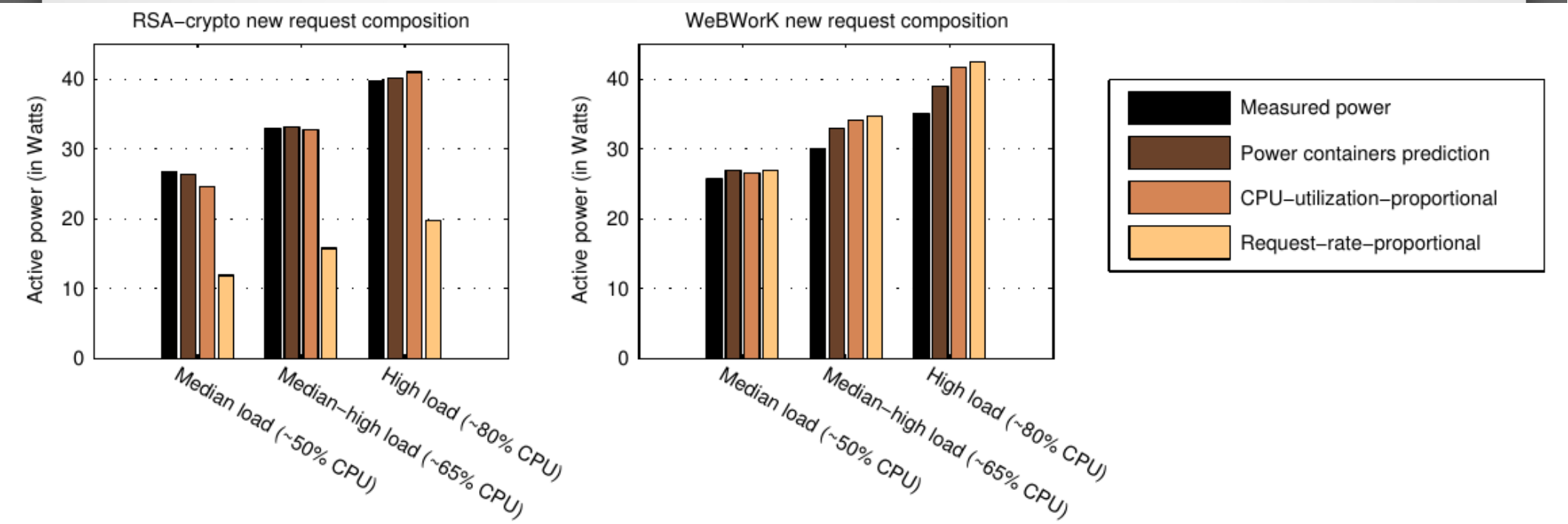




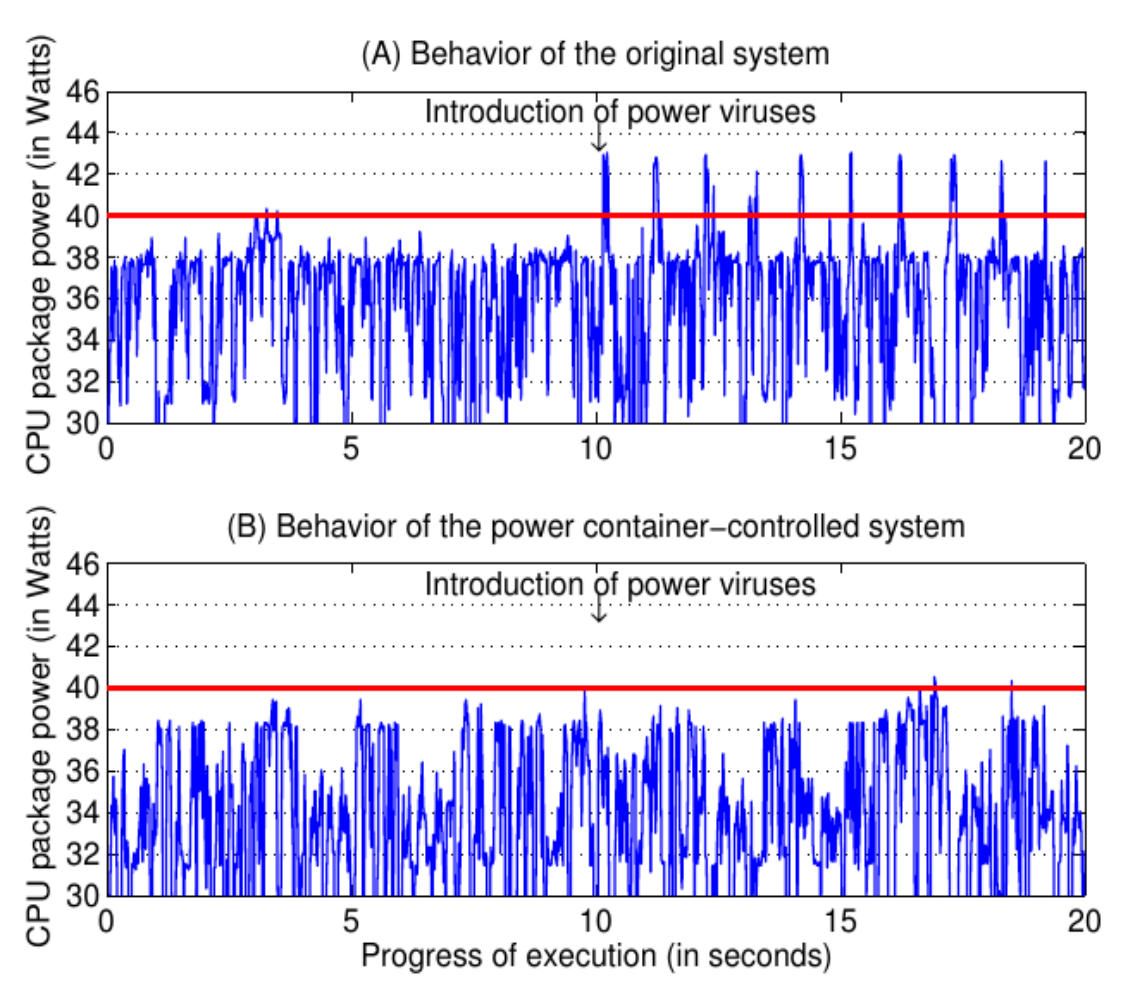
Modeling approaches

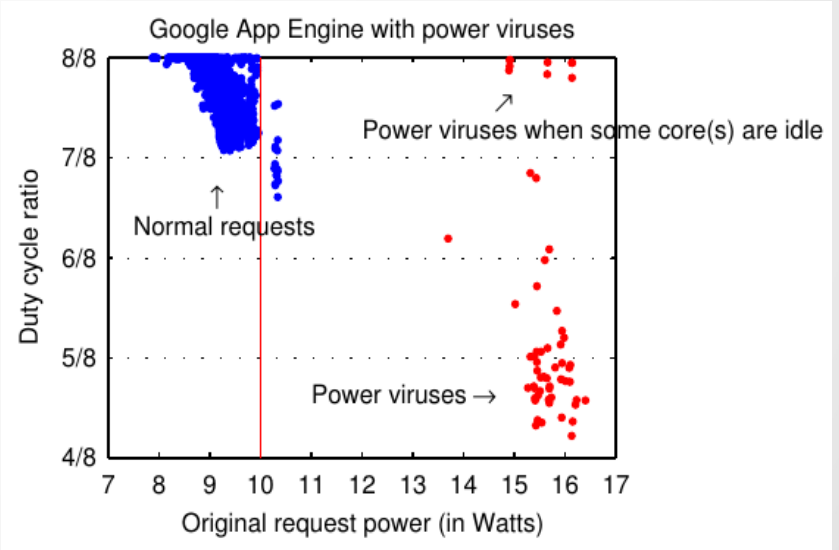


Evaluation

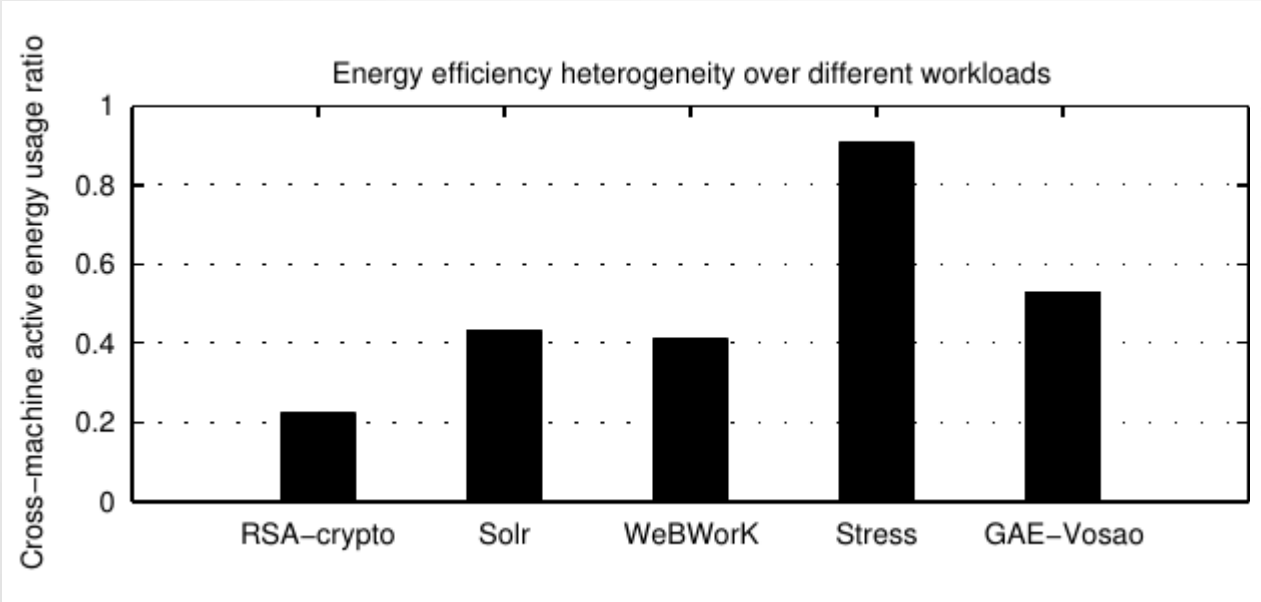

RSA–crypto new request composition
WeBWorK new request composition
Active power (in Watts)
0
10
20
30
40
Median load (~50% CPU)
Median–high load (~65% CPU)
High load (~80% CPU)
Measured power
Power containers prediction
CPU–utilization–proportional
Request–rate–proportional

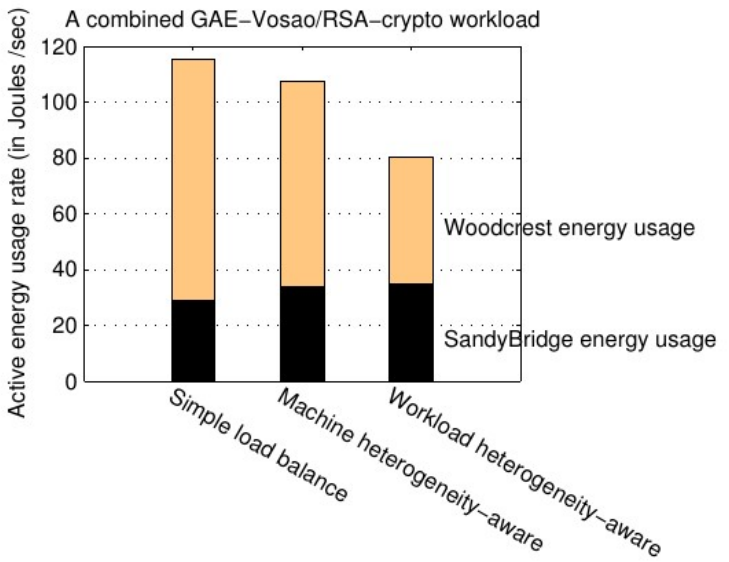
Power Virus limiting





Exploiting Heterogeneity





Discussion / Critique

- They don't make clear when they use RAPL / CPU Power for calibration/comparison and when they use full system power
- The details on how to track requests were a bit short
- … as were the details on I/O power attribution
- Why are Viruses sometimes not limited? Per request power limit is exceeded! Also … What about energy?
- What about the overhead in OS primitives?
- “... sample the SandyBridge power once every 10 ms ...”