Multiple Clock and Voltage Domains for Chip Multi Processors

Efraim Rotem- Intel Corporation Israel
Avi Mendelson- Microsoft R&D Israel
Ran Ginosar- Technion Israel institute of Technology
Uri Weiser- Technion Israel Institute of Technology

Presented by:
Michael Moeng- University of Pittsburgh

Outline

- **Multiple Voltage Domains**
- **Power Model**
- **Performance Model**
- **Power Management Policies**
- **Results**

Multiple Voltage Domains

- Multiprocessors can distribute power in several ways:
  - Single clock domain (also implies single voltage domain)
    - All cores operate at same frequency and voltage

Multiple Voltage Domains

- Multiprocessors can distribute power in several ways:
    - Single clock domain (also implies single voltage domain)
        - All cores operate at same frequency and voltage
    - Multiple clock domains -- communicate through FIFO buffers with minor overhead
        - Multiple Voltage Domains:
            - Cores independently scale frequency and voltage

Multiple Voltage Domains

- Multiprocessors can distribute power in several ways:
    - Single clock domain (also implies single voltage domain)
        - All cores operate at same frequency and voltage
    - Multiple clock domains -- communicate through FIFO buffers with minor overhead
        - Multiple Voltage Domains:
            - Cores independently scale frequency and voltage
        - Single voltage domain
            - Individual cores use only frequency scaling
            - Single voltage for all cores determined by highest frequency

Multiple Voltage Domains

- Multiprocessors can distribute power in several ways:
    - Single clock domain (also implies single voltage domain)
        - All cores operate at same frequency and voltage
    - Multiple clock domains -- communicate through FIFO buffers with minor overhead
        - Multiple Voltage Domains:
            - Cores independently scale frequency and voltage
        - Single voltage domain
            - Individual cores use only frequency scaling
            - Single voltage for all cores determined by highest frequency
    - Clustered topologies:
        - Hybrid approach between two extremes

Multiple Voltage Domains - Power Delivery

- Previous works assume no overhead for extra voltage regulators.
- A voltage regulator must be designed for a nominal current.
- Additional voltage regulators have consequences for:

Multiple Voltage Domains - Power Delivery

- Previous works assume no overhead for extra voltage regulators.
- A voltage regulator must be designed for a nominal current.
- Additional voltage regulators have consequences for:
    - Current Sharing
    - Power Delivery Network Resistance

Current Sharing

- A regulator will realistically be designed for a maximum current of 130% to 250% of its nominal current.

- Compare chip power delivery systems:
    - single voltage regulator, X~2.5X amps
    - two voltage regulators, .5X~1.25X amps each
    - N voltage regulators, X/N~2.5X/N amps each

Current Sharing

- A regulator will realistically be designed for a maximum current of 130% to 250% of its nominal current.

- Compare chip power delivery systems:
  - single voltage regulator, X~2.5X amps
  - two voltage regulators, .5X~1.25X amps each
  - N voltage regulators, X/N~2.5X/N amps each

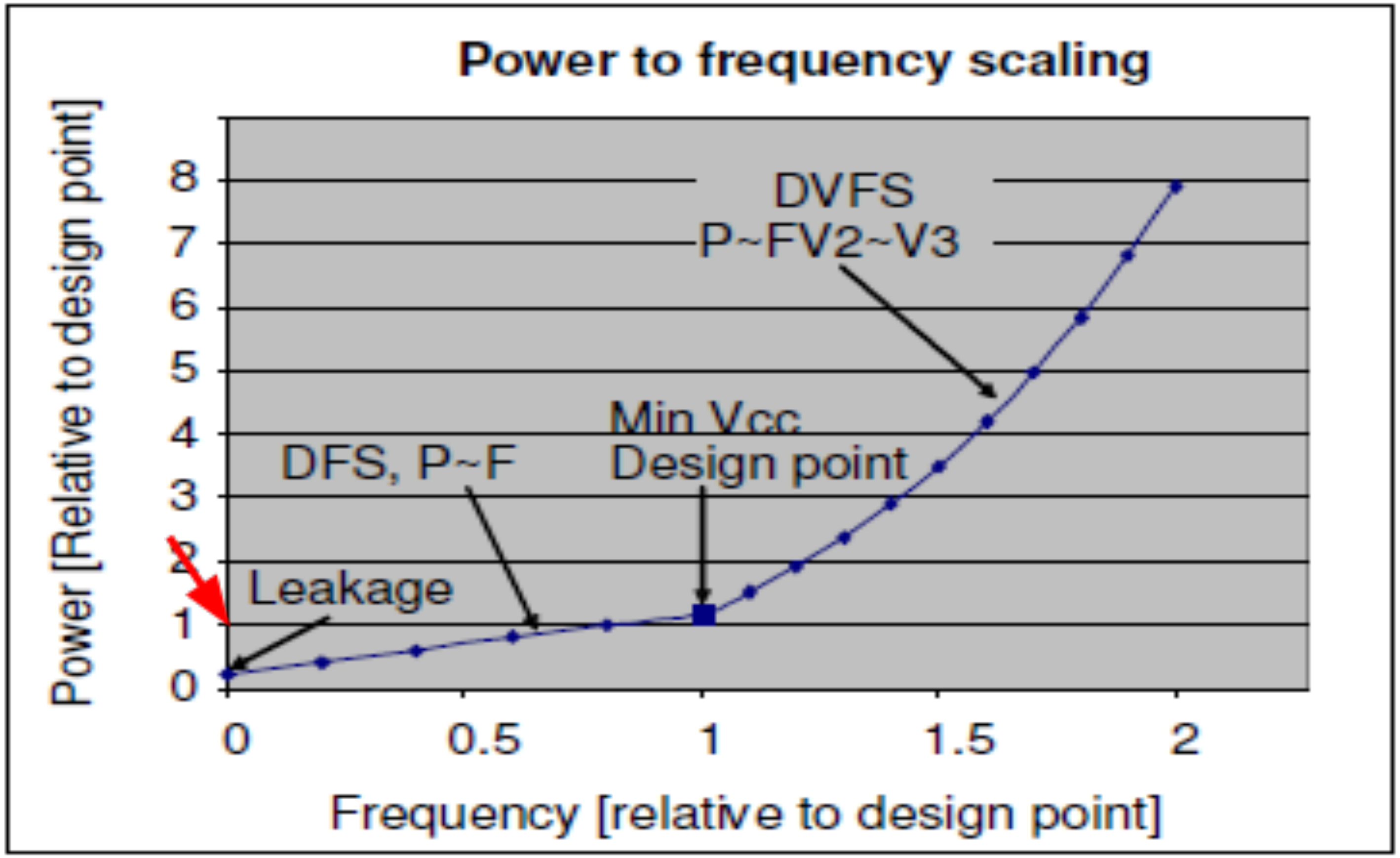
- Maximum power to a single core can be much higher with fewer regulators.

Resistance in Power Delivery Network

- Splitting Power Delivery Network N ways results in N times higher resistance
- For symmetric workloads, each regulator also supplies N times less current -- no penalty
- When assigning power asymmetrically, higher resistance results in a voltage drop -- wasted power

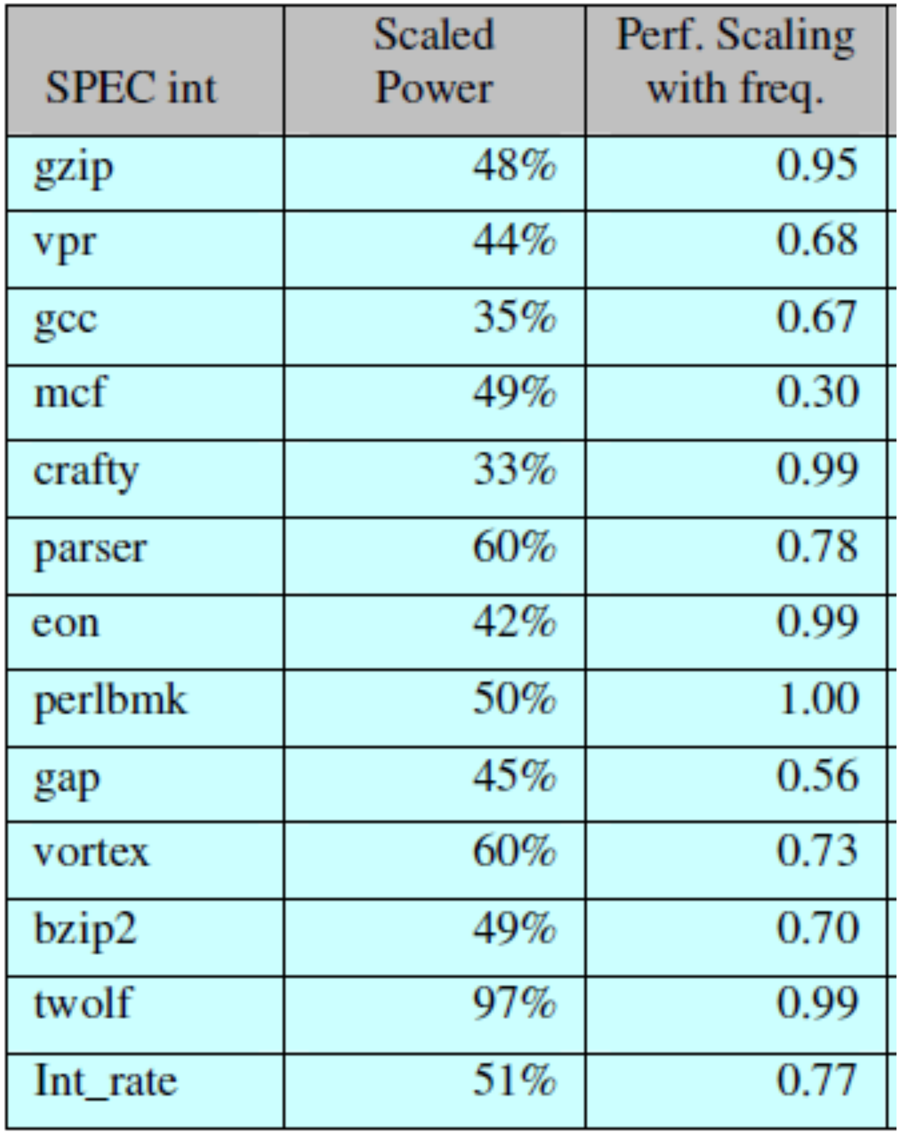
Power Model

Power Model

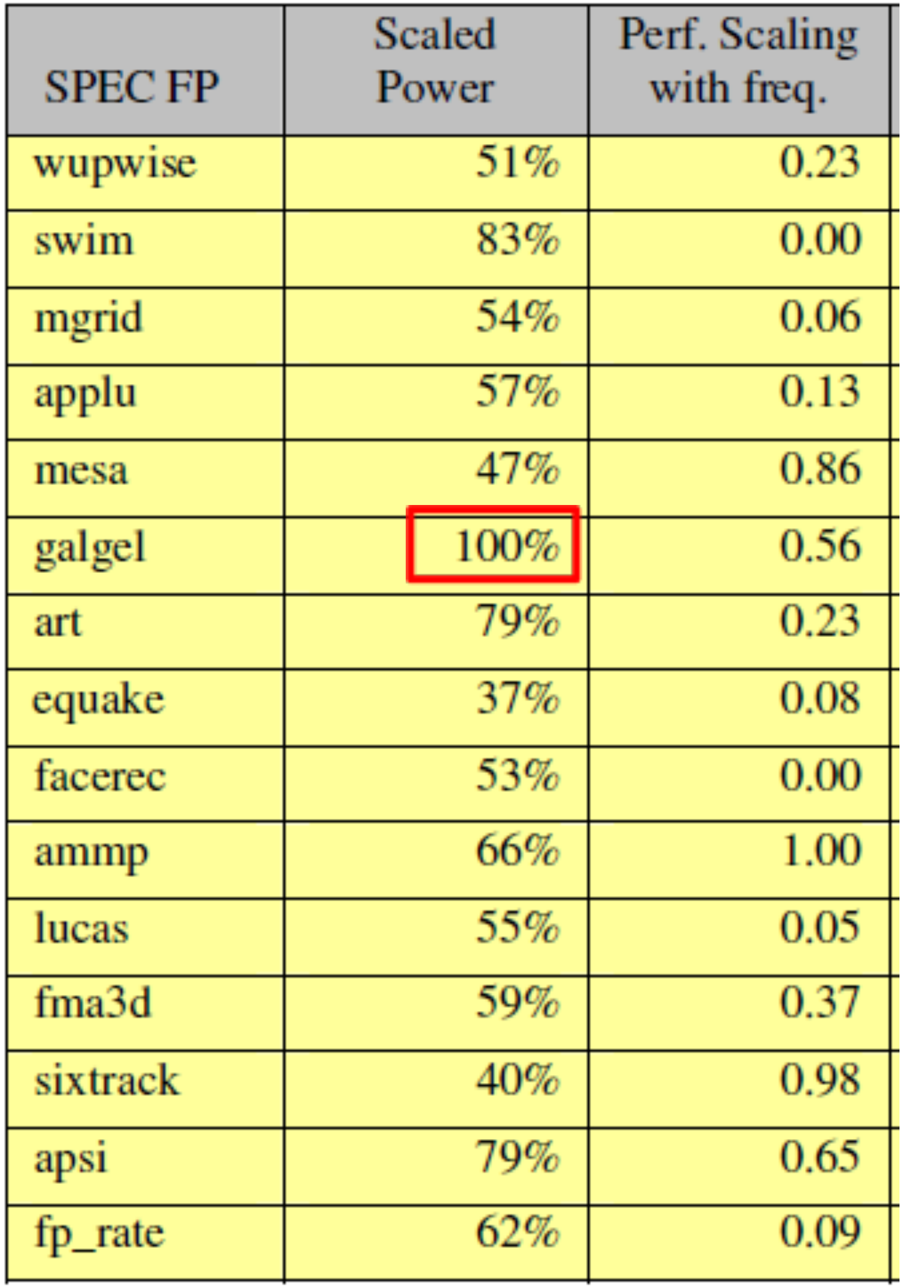


- Assumption: Future high-oower CMPs will be designed with nominal frequency and power at the minimum operating voltage allowed by a process.

Benchmarks



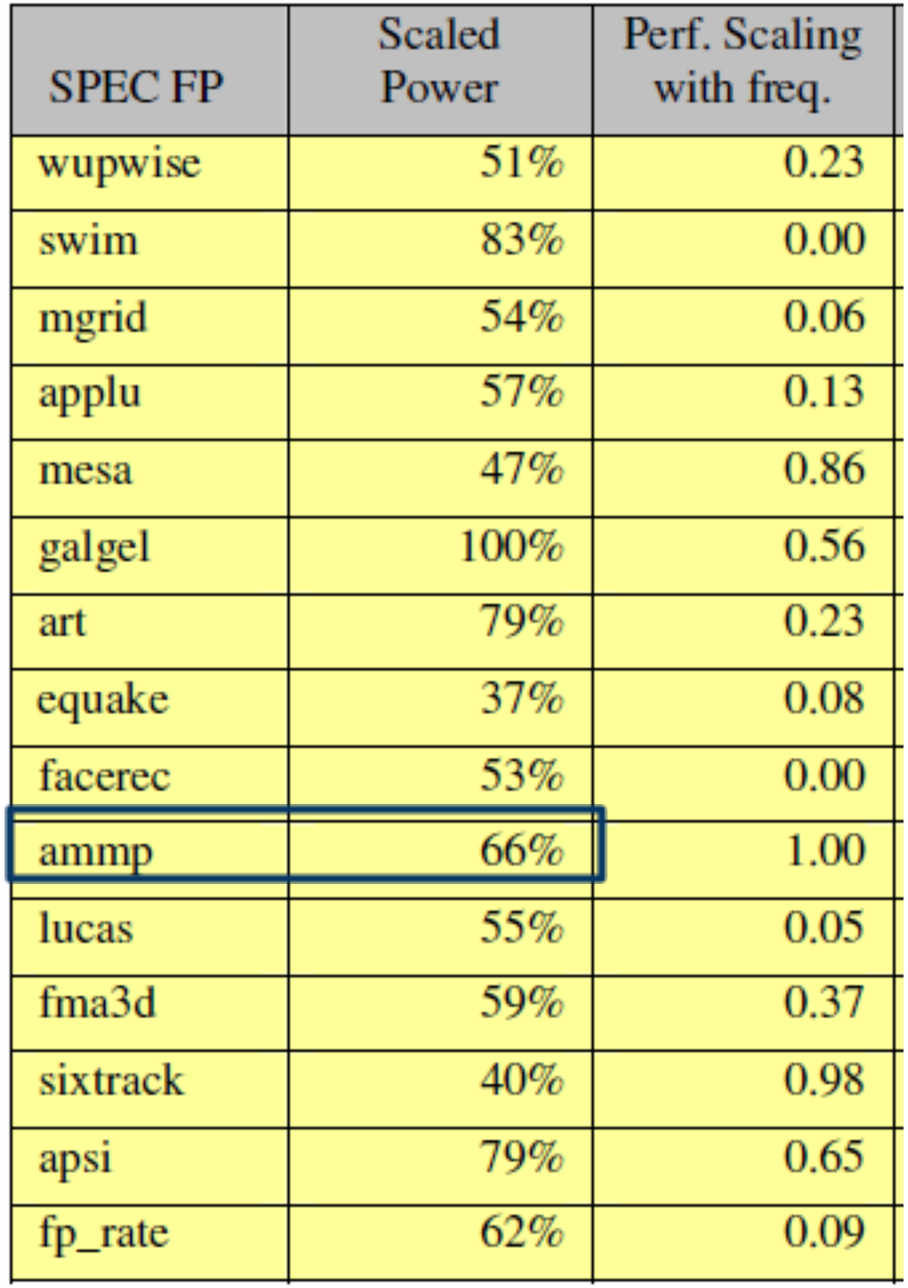
SPEC int	Scaled Power	Perf. Scaling with freq.
gzip	48%	0.95
vpr	44%	0.68
gcc	35%	0.67
mcf	49%	0.30
crafty	33%	0.99
parser	60%	0.78
eon	42%	0.99
perlbmk	50%	1.00
gap	45%	0.56
vortex	60%	0.73
bzip2	49%	0.70
twolf	97%	0.99
Int_rate	51%	0.77



SPEC FP	Scaled Power	Perf. Scaling with freq.
wupwise	51%	0.23
swim	83%	0.00
mgrid	54%	0.06
applu	57%	0.13
mesa	47%	0.86
galgel	100%	0.56
art	79%	0.23
equake	37%	0.08
facerec	53%	0.00
ammp	66%	1.00
lucas	55%	0.05
fma3d	59%	0.37
sixtrack	40%	0.98
apsi	79%	0.65
fp_rate	62%	0.09

Quick Check

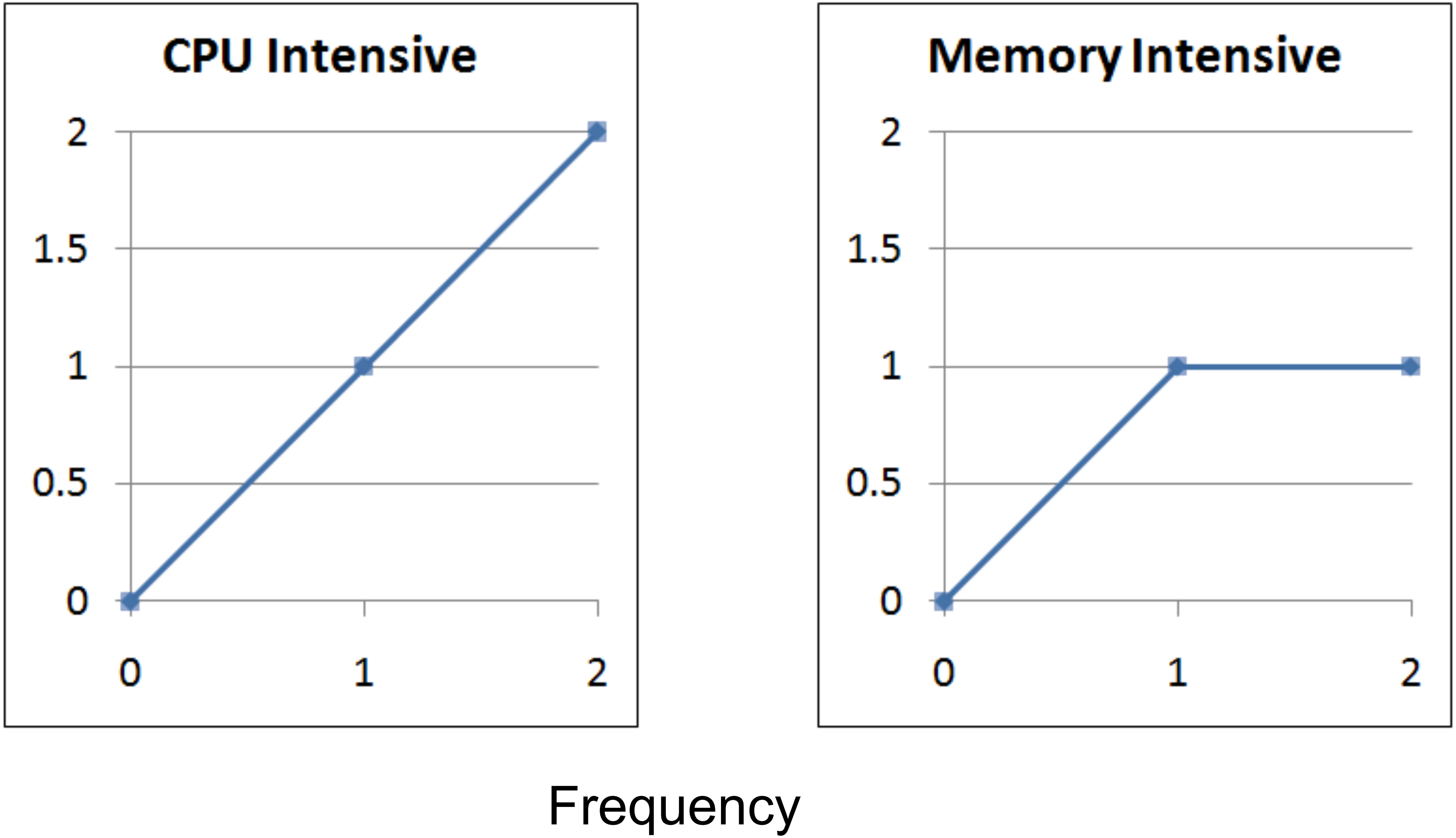
If we run 16 copies of **ammp** at nominal frequency, how much power do we have left?



SPEC FP	Scaled Power	Perf. Scaling with freq.
wupwise	51%	0.23
swim	83%	0.00
mgrid	54%	0.06
applu	57%	0.13
mesa	47%	0.86
galgel	100%	0.56
art	79%	0.23
equake	37%	0.08
facerec	53%	0.00
ammp	66%	1.00
lucas	55%	0.05
fma3d	59%	0.37
sixtrack	40%	0.98
apsi	79%	0.65
fp_rate	62%	0.09

Performance Model

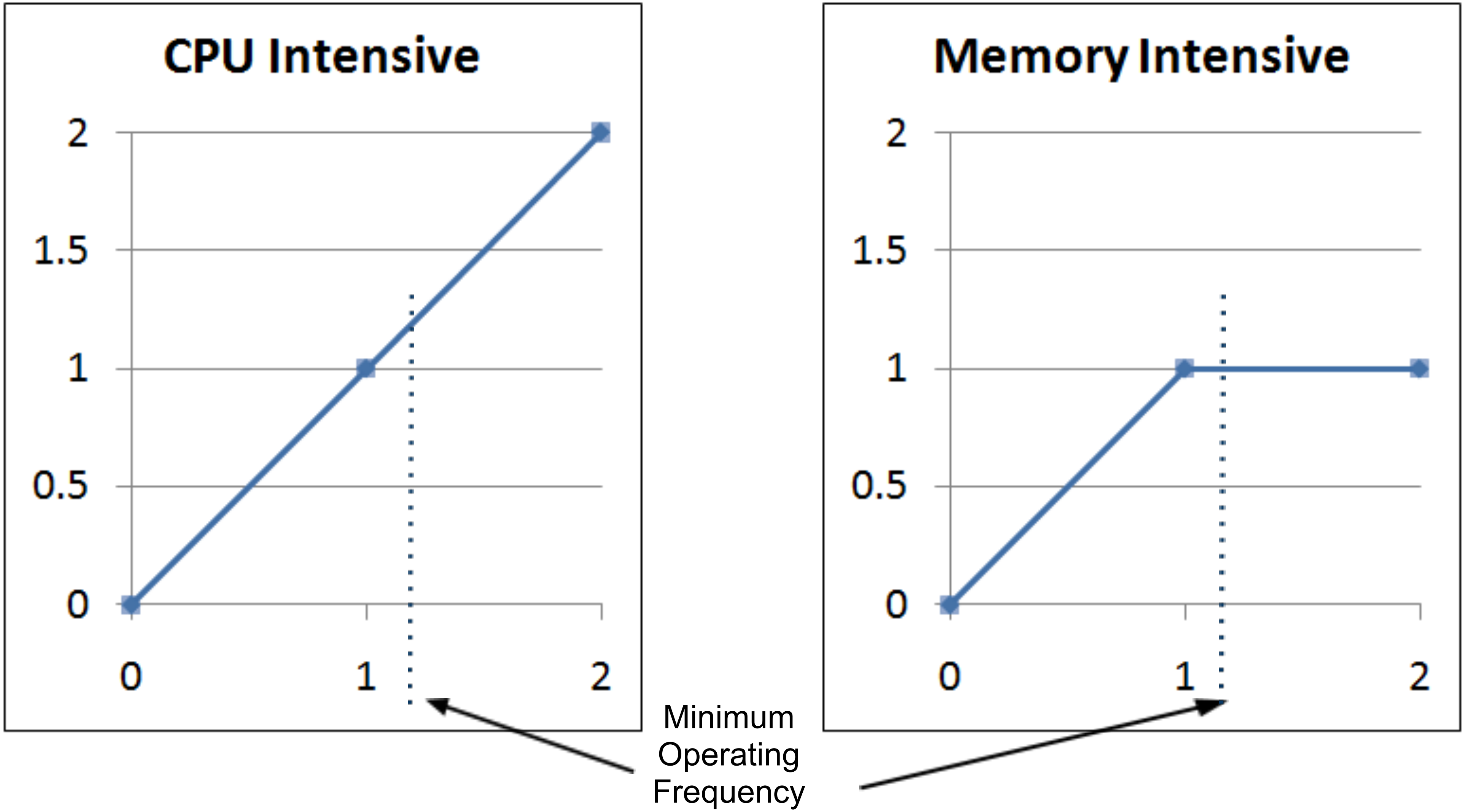
Performance Model



CPU Intensive

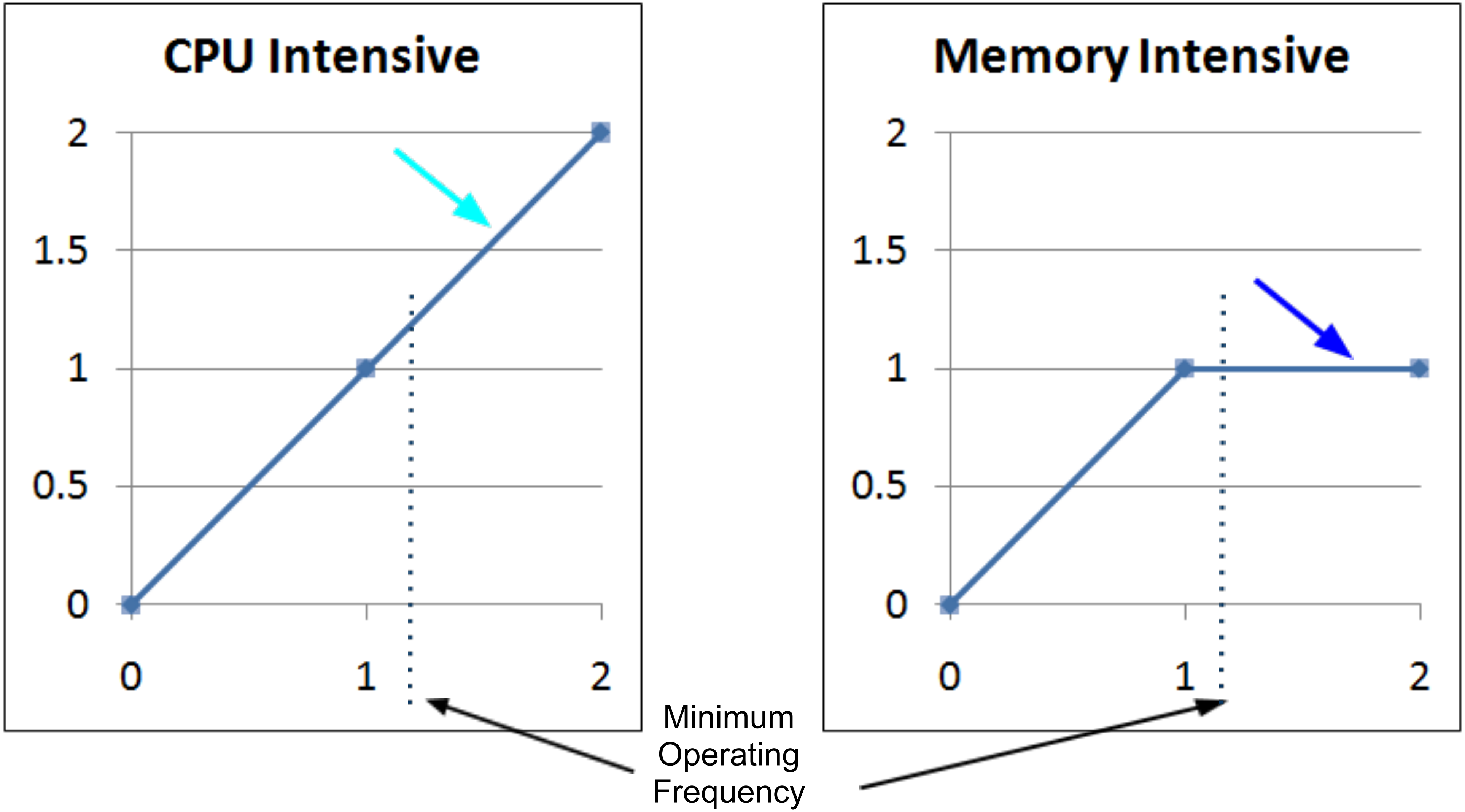
Memory Intensive

Performance Model



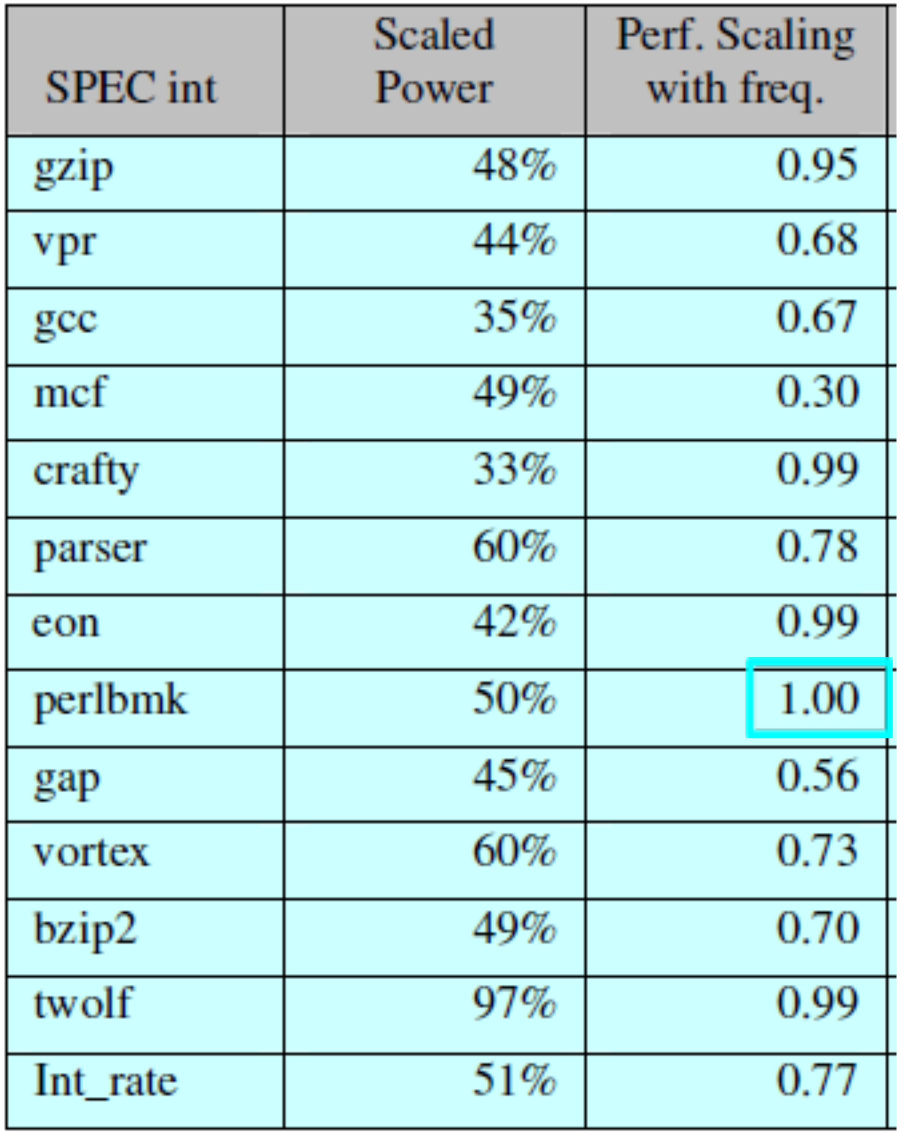
Minimum Operating Frequency

Performance Model

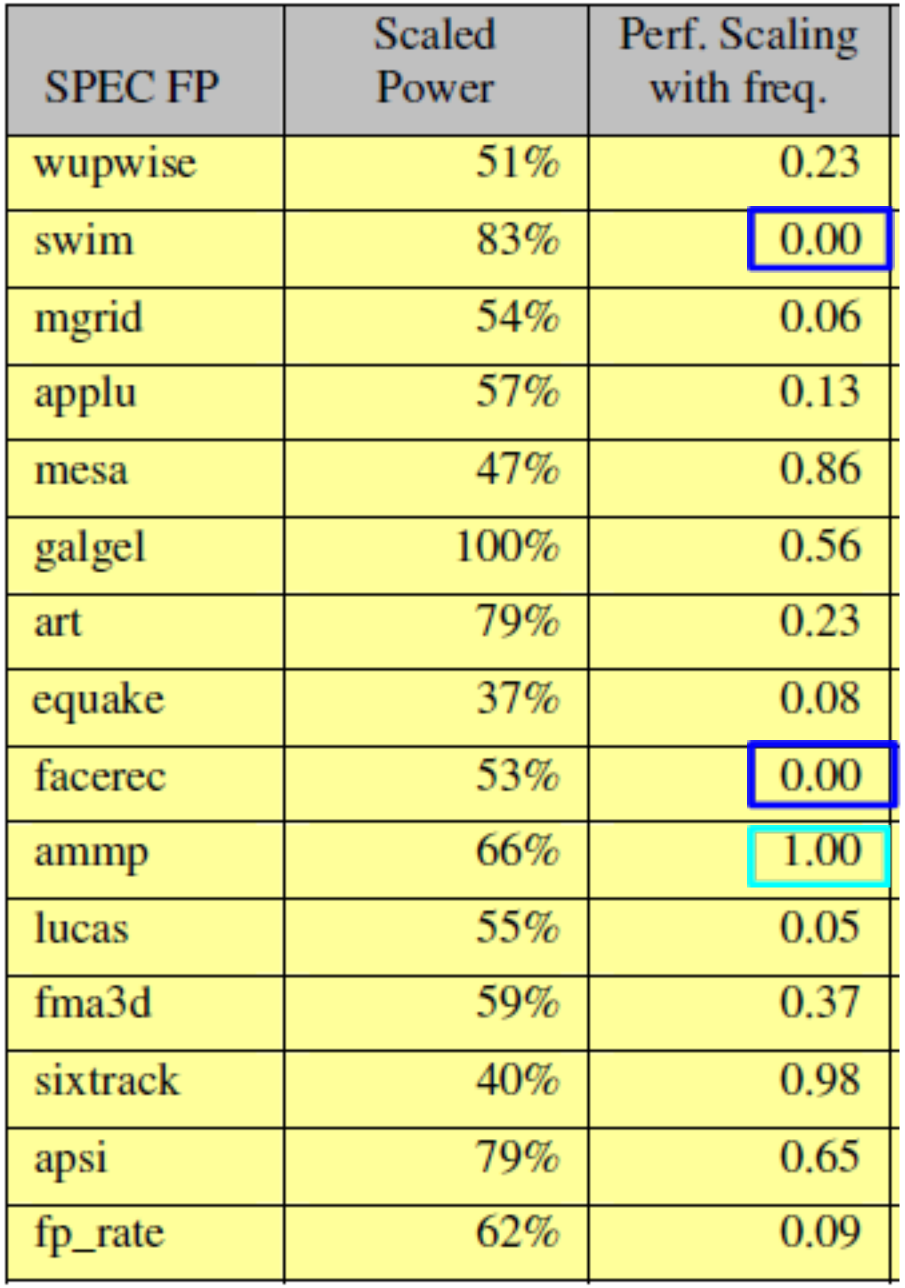


Minimum
Operating
Frequency

Benchmarks



SPEC int	Scaled Power	Perf. Scaling with freq.
gzip	48%	0.95
vpr	44%	0.68
gcc	35%	0.67
mcf	49%	0.30
crafty	33%	0.99
parser	60%	0.78
eon	42%	0.99
perlbmk	50%	1.00
gap	45%	0.56
vortex	60%	0.73
bzip2	49%	0.70
twolf	97%	0.99
Int_rate	51%	0.77



SPEC FP	Scaled Power	Perf. Scaling with freq.
wupwise	51%	0.23
swim	83%	0.00
mgrid	54%	0.06
applu	57%	0.13
mesa	47%	0.86
galgel	100%	0.56
art	79%	0.23
equake	37%	0.08
facerec	53%	0.00
ammp	66%	1.00
lucas	55%	0.05
fma3d	59%	0.37
sixtrack	40%	0.98
apsi	79%	0.65
fp_rate	62%	0.09

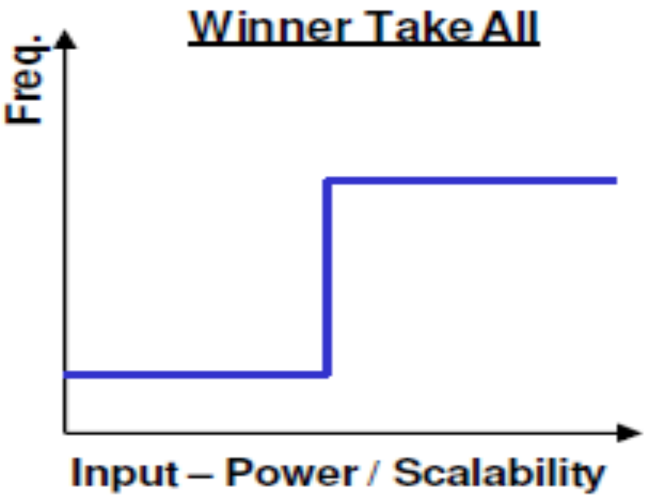
Power Management Policies

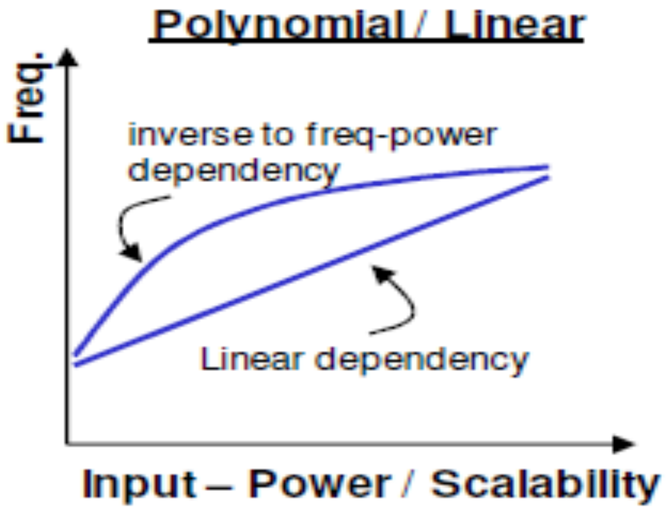
Goal: Maximize performance given a power constraint

Power Management Policies

Goal: Maximize performance given a power constraint

- Assume benchmarks have already been profiled (we know the frequency scaling)
- Policies assume its better to give core with better scalability a higher frequency, and provide a function of frequency given scalability.





Quick Check 2

The polynomial policy scales frequency inversely with the freq-power dependency.

What is this function?

Power Management:
following constraints

After each core's desired power level is determined:

- If desired current exceeds current capacity, scale frequency down to maximum allowed
- All values are normalized so total power meets power constraints

Evaluation

- Simulation and real machine execution used to determine parameters for each benchmark

Evaluation

- Simulation and real machine execution used to determine parameters for each benchmark

- "Oracle" simulated using a gradient descent algorithm

Evaluation

- Simulation and real machine execution used to determine parameters for each benchmark

- "Oracle" simulated using a gradient descent algorithm

- Monte Carlo modeling for workload generation
- Evaluates workloads with 2,4,8,12,14,16 threads to show performance with idle cores

Evaluation

- Simulation and real machine execution used to determine parameters for each benchmark

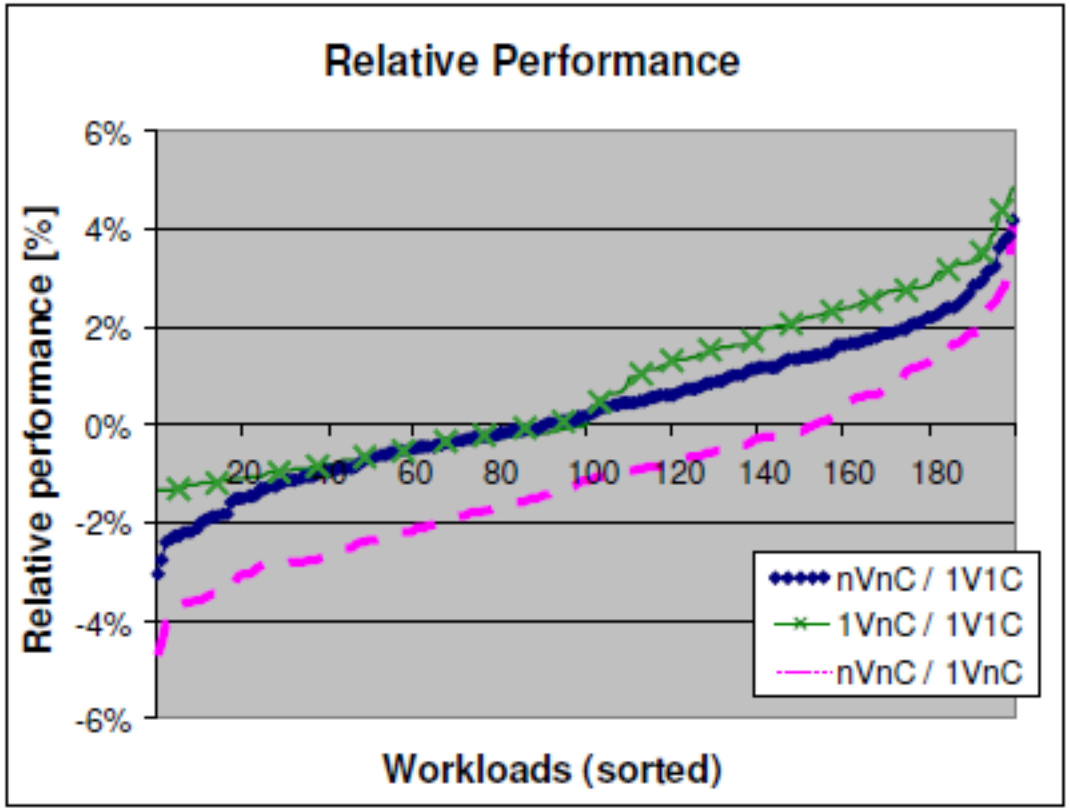
- "Oracle" simulated using a gradient descent algorithm

- Monte Carlo modeling for workload generation
- Evaluates workloads with 2,4,8,12,14,16 threads to show performance with idle cores

- Baseline is single-clock domain, single-voltage domain
  - 10-30% improvement over no-DVFS
  - Quick Check 3: How does this improve performance?

Oracle policy

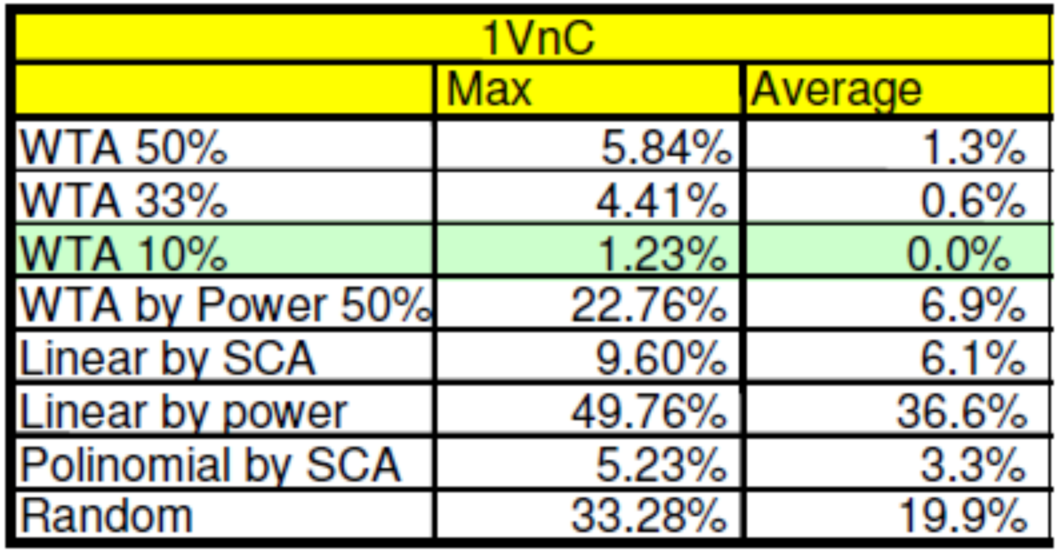
- For about half the workloads, it's best to use the same frequency for all cores



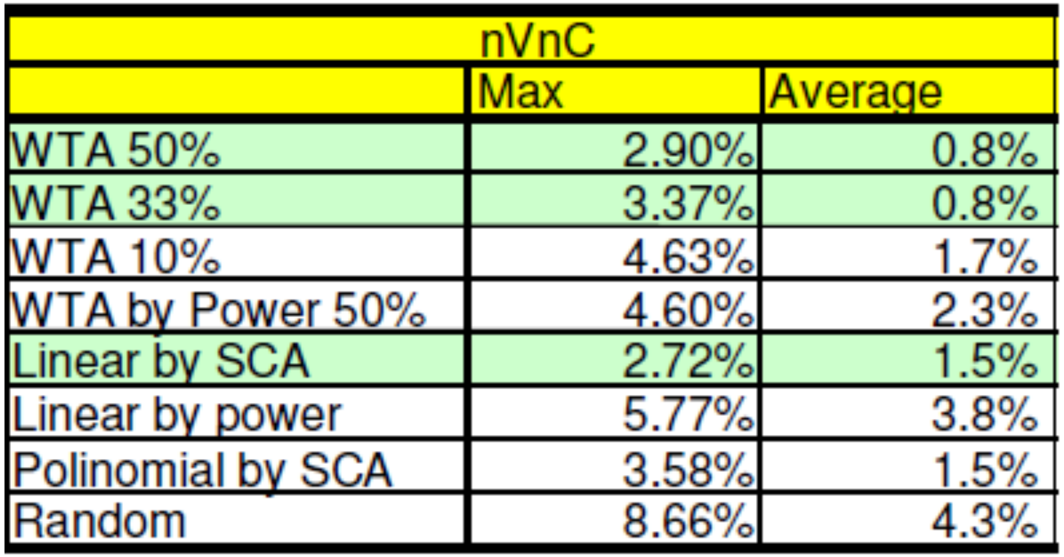
- Loss comes from asynchronous FIFO buffers

Best policies for each configuration

- Shows loss vs oracle
  - Lower is better
- Knowledge of frequency scalability is crucial



1VnC		
	Max	Average
WTA 50%	5.84%	1.3%
WTA 33%	4.41%	0.6%
WTA 10%	1.23%	0.0%
WTA by Power 50%	22.76%	6.9%
Linear by SCA	9.60%	6.1%
Linear by power	49.76%	36.6%
Polinomial by SCA	5.23%	3.3%
Random	33.28%	19.9%

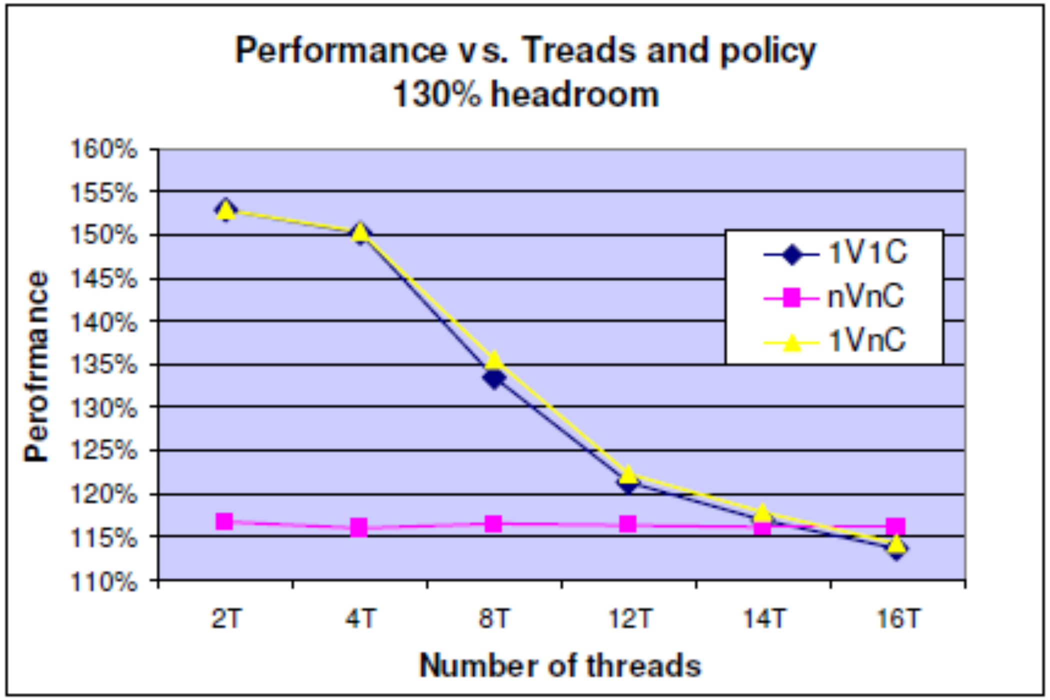


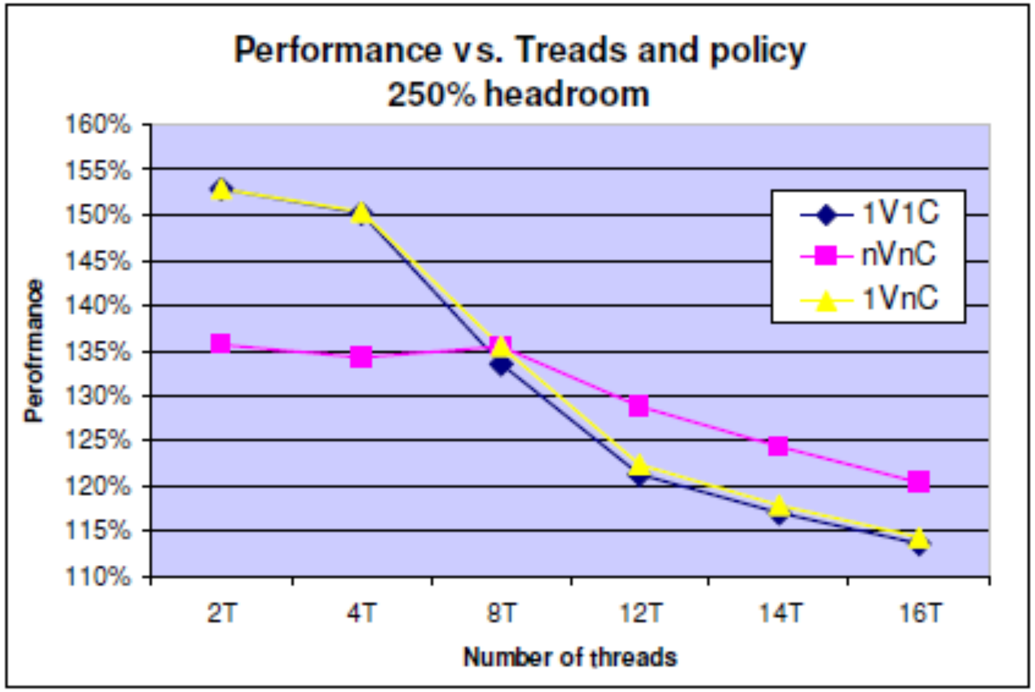
nVnC		
	Max	Average
WTA 50%	2.90%	0.8%
WTA 33%	3.37%	0.8%
WTA 10%	4.63%	1.7%
WTA by Power 50%	4.60%	2.3%
Linear by SCA	2.72%	1.5%
Linear by power	5.77%	3.8%
Polinomial by SCA	3.58%	1.5%
Random	8.66%	4.3%

WTA = Winner Take All, SCA = Scalability

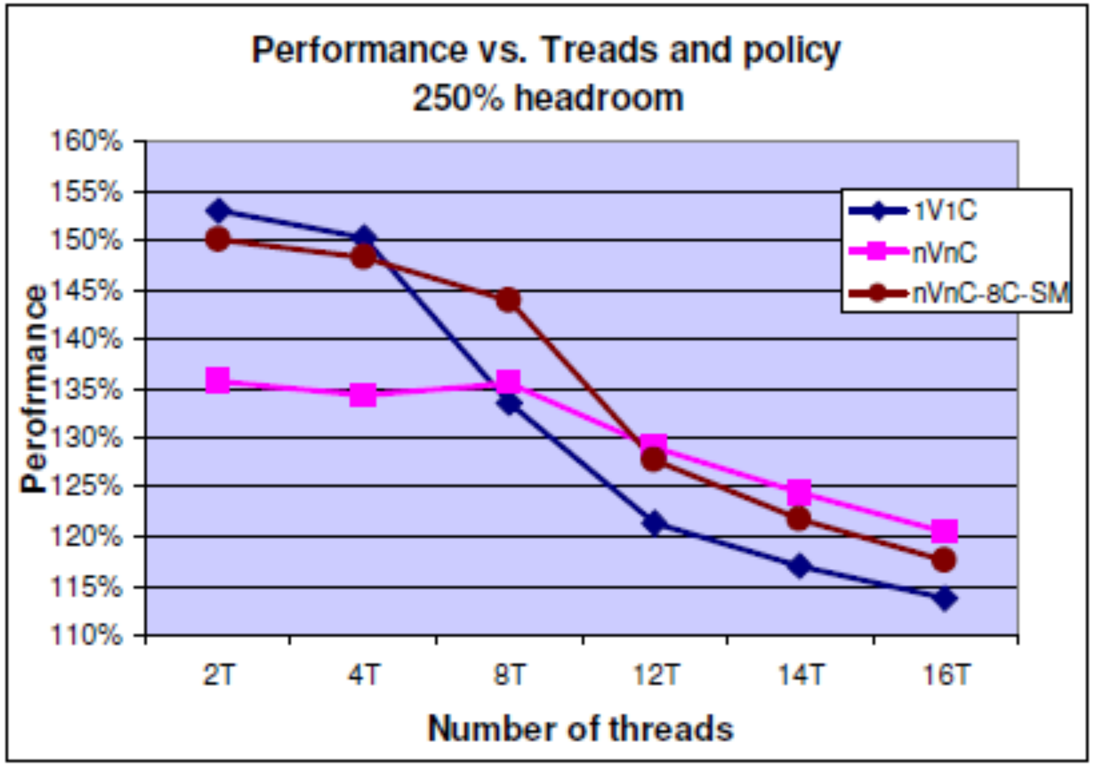
Limiting threads

- Multiple voltage domains are heavily dependent on high headroom for voltage regulators





Clustered Topologies



- Matches performance of single voltage domain with few threads
- Matches performance of multiple voltage domains with many threads