



Characterizing the Impact of Different Memory-Intensity Levels

Ramakrishna Kotla

The University of Texas at Austin

Anirudh Devgan, Soraya Ghiasi,

Freeman Rawson, Tom Keller

IBM Austin Research Laboratory

Power Problems in Modern Processors

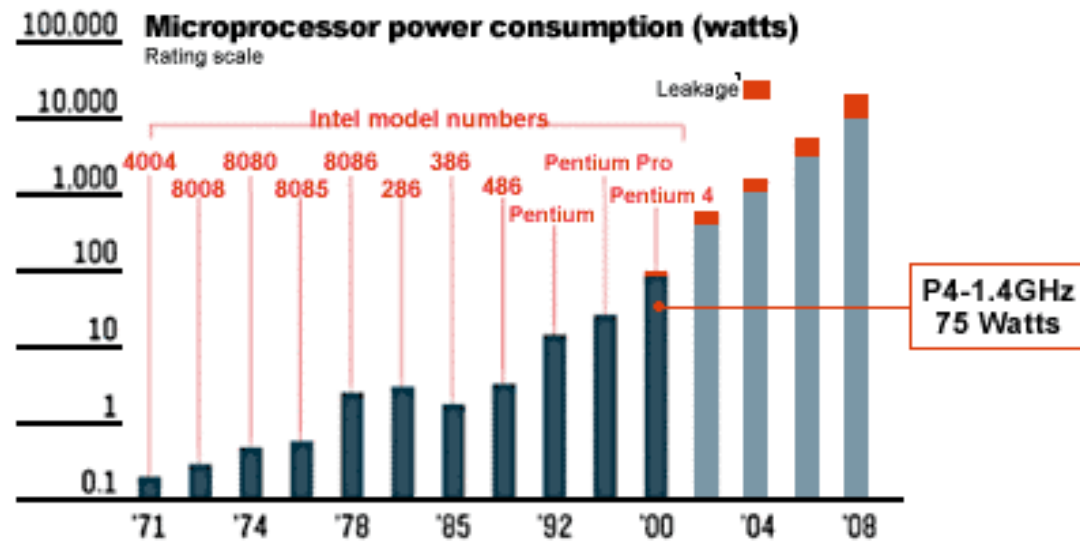
Forbes Magazine Review “Too Hot to Handle”

Feeding the Pentium Beast

04.02.01

from [Too Hot to Handle](#)

Intel's astonishing march toward ever denser chips comes with a cost: skyrocketing energy demands. The prospect of 100-kilowatt chips has designers scrambling for solutions.



Projection figures assume no advances in energy efficiency techniques. ¹Leakage is the dissipation of energy as a result of imperfect transistor function. Source: Intel.

Technology Trends

- Increase in transistor densities and operating frequencies
 - Power densities↑
 - Memory Wall
- Problems
 - Cooling
 - Power supply and distribution
 - Energy

Application Characteristics

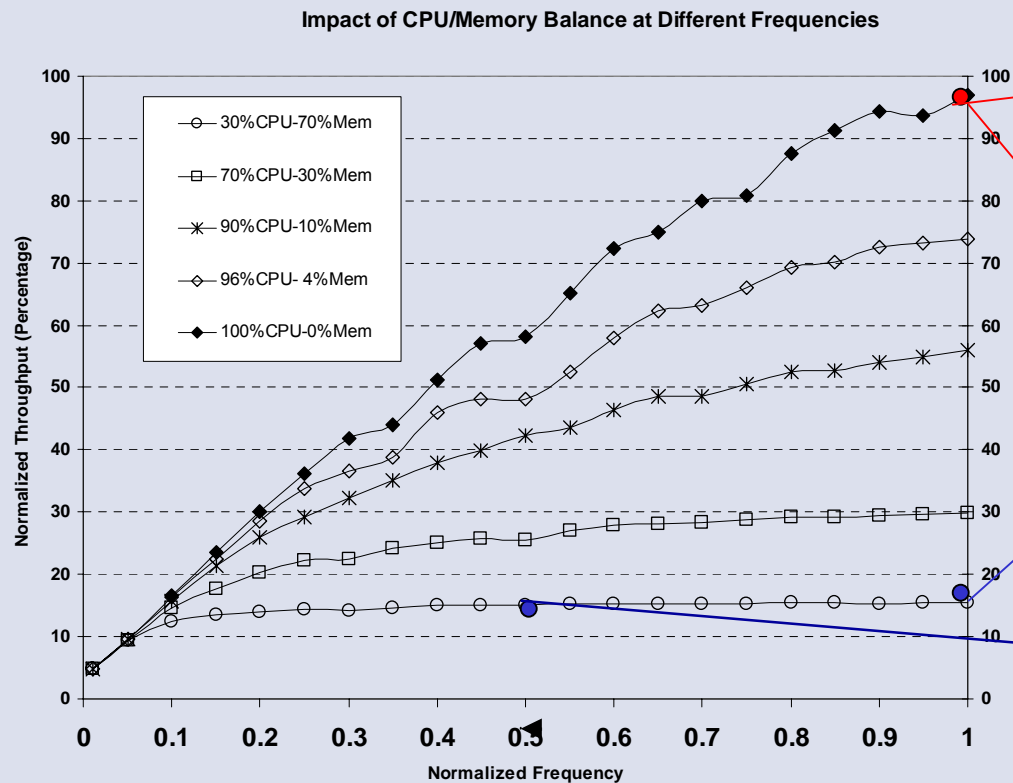
- Inter-application variability
 - CPU intensive - gzip, tpcw
 - Memory intensive - mcf, specJbb
- Intra-application variability
 - Phases of execution - gzip
 - Variation in load – Web servers
- Exploit application heterogeneity to reduce power consumption

Heterogeneous Cores

- Cores with different power/performance characteristics
 - Technology Trends – creating an opportunity to build
 - Application characteristics – create an opportunity to use
- Requirements
 - Applications run on all cores
 - Scheduling mechanisms
 - Match application characteristics to heterogeneous processors

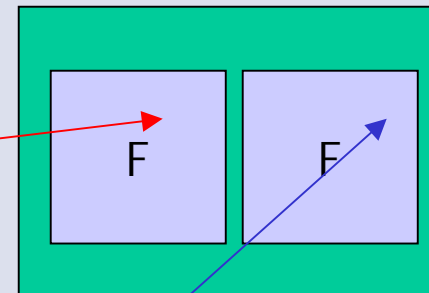
Making the case for Heterogeneous cores :

About 65% reduction in power without any performance degradation

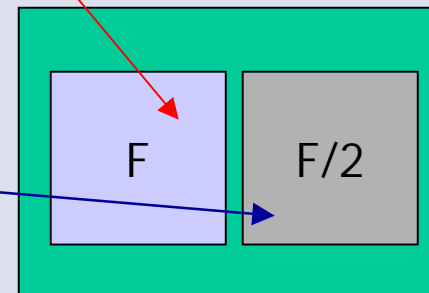


Synthetic benchmark on Power 4 machine

Homogeneous Core



Heterogeneous Core

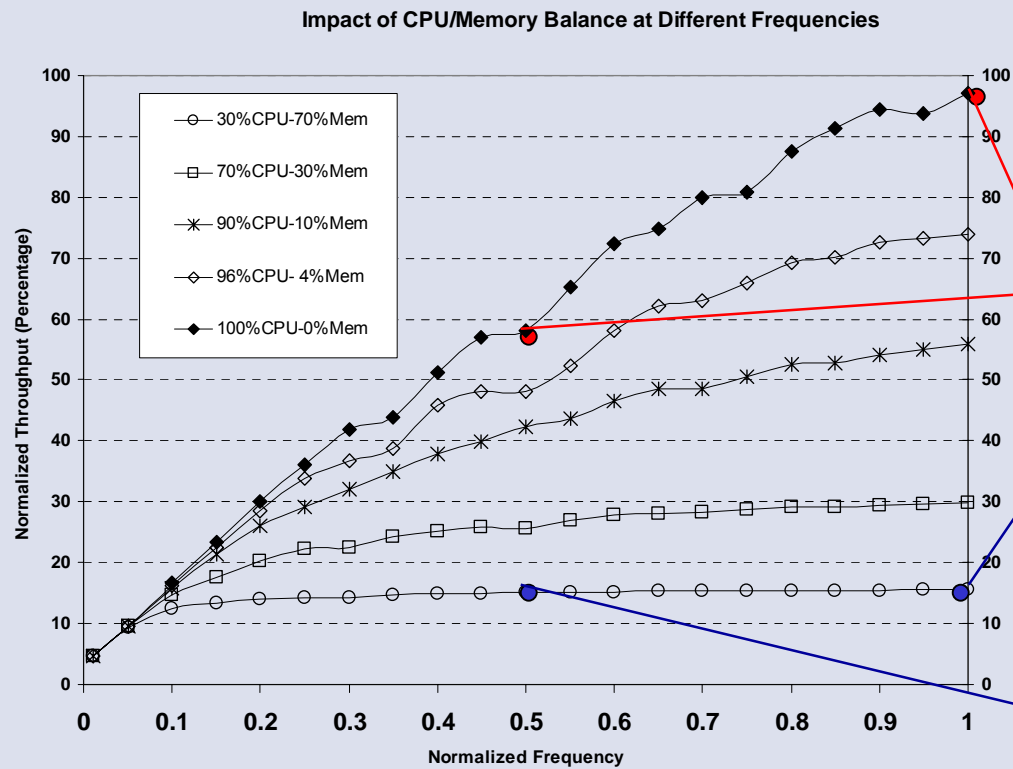


No loss in performance

Single-ISA Heterogeneous Cores

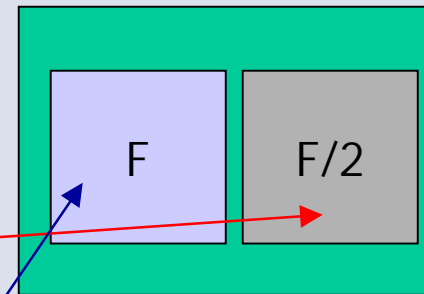
- Applications naturally run on any core
- Previous work: Micro-architectural heterogeneity [ISCA04, Micro03]
 - Cores with different micro-architectures
 - Run at same frequency
 - Ad-hoc sampling based scheduling mechanisms
- Our approach : Heterogeneity due to frequency/voltage variability
 - Cores with same micro-architecture but run at different frequencies/voltage
 - Accurate prediction based scheduling mechanism

Importance of Scheduling Heterogeneous Cores



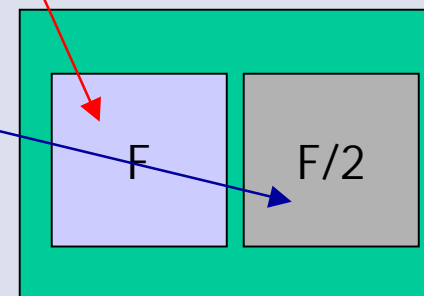
Synthetic benchmark on Power 4 machine

Traditional Linux scheduler



40% Performance loss

Heterogeneous Scheduler



No loss in performance

Heterogeneous Processor Scheduler

- Schedule Jobs onto processors running at different frequencies with minimal performance degradation
- Requirements :
 - Characterize applications : CPU/Memory intensive behavior
 - Adapt to phase changes of a program
 - Estimate performance at various frequencies

Performance Model

$$IPC = \frac{1}{\frac{1}{\alpha} + \frac{1}{Instr} (N_{L2} T_{L2_stalls} + N_{L3} T_{L3_stalls} + N_{mem} T_{mem_stalls}) f + \frac{C_{other_stalls}}{Instr}}$$

N_x : Number of memory references to various memory levels

f : Processor frequency

C_{other_stalls} : Cycles due to branch, pipeline stalls,...

$Instr$: Number of instructions executed

T_{X_stalls} : Latencies to various levels in cache hierarchy

α : IPC of a perfect machine with infinite L1 caches and no stalls

Performance Model : Intuition

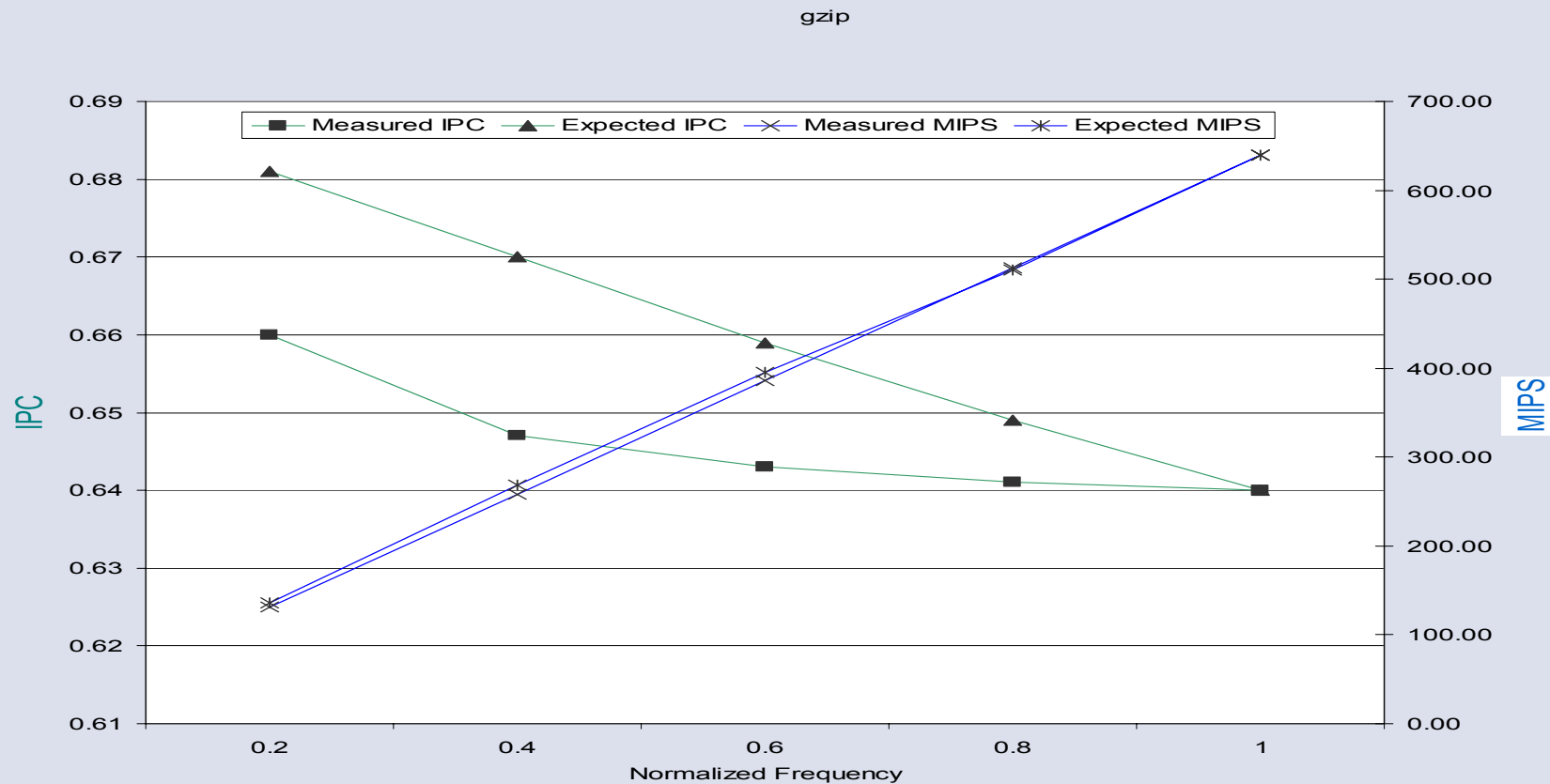
- CPU intensive applications
 - $IPC = \alpha$
 - Throughput (in IPS) = $\alpha * f$
- Memory intensive applications
 - $IPC = Instr/T * f$
 - Throughput (in IPS) = $Instr/T$
- Schedule CPU intensive application onto a processor running at high frequency
- Schedule memory intensive applications onto a processor running at low frequency

Implementation of Performance Model

- Using Power4 performance counters
- Assumptions :
 - TLB stalls are ignored
 - Memory/Cache Latencies are constant and based on random accesses
- Advantages :
 - Low overhead : Reading 8 performance registers
 - Simple model to characterize and estimate performance at various frequencies
 - General model – Not application specific

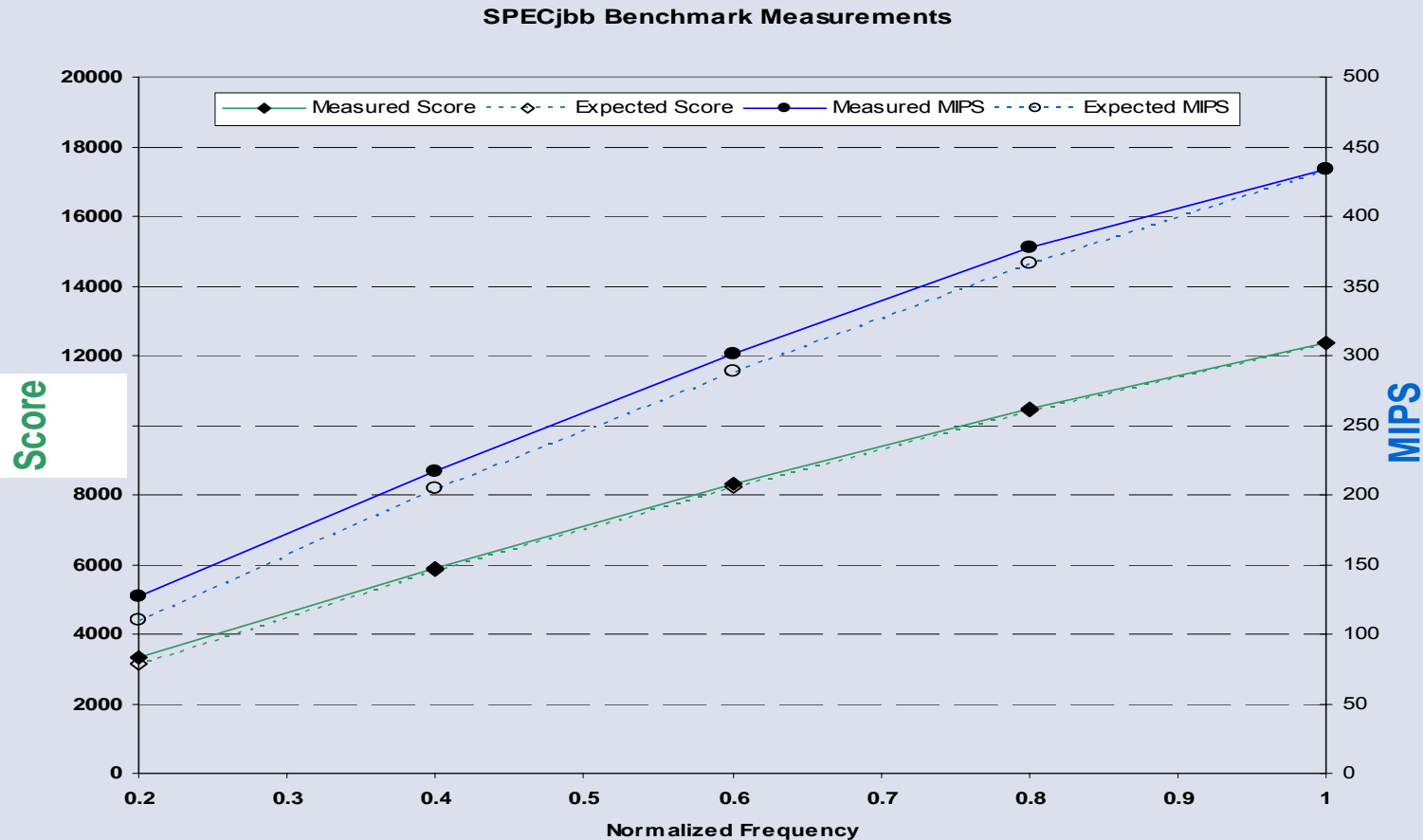
Performance Model Accuracy: Gzip

Run on Power4 processor with f (Nominal) = 1GHz with avg. error = 2.4%



Performance Model Accuracy : SPECjbb

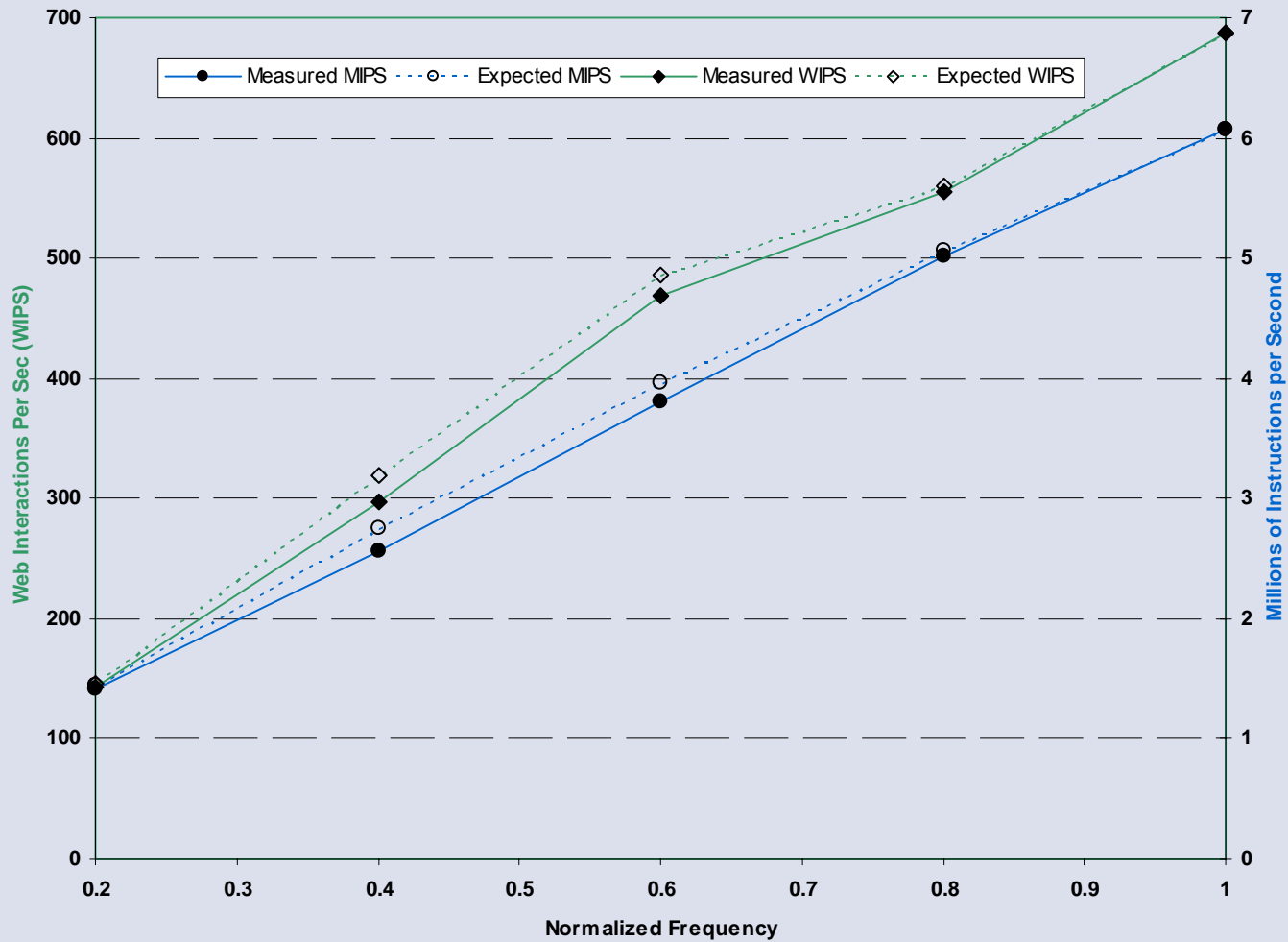
Run on Power4 processor with f (Nominal) = 1GHz with avg. error < 7%



Performance Model Accuracy : TPCW

Run on Power4 processor with f (Nominal) = 1GHz with avg. error < 7%

TPC-W Benchmark Measurements

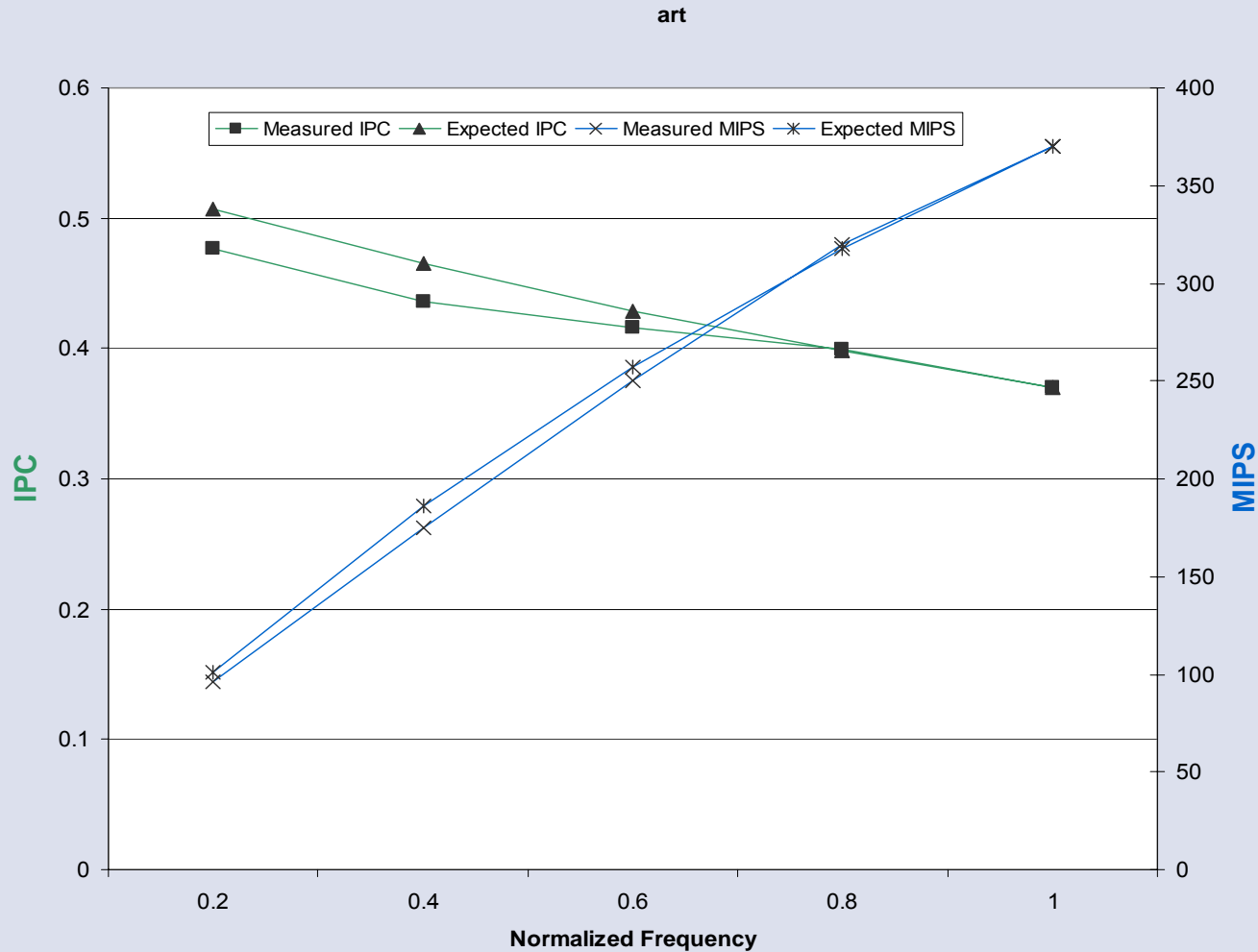


Handling Prefetching

- Prefetching hides latencies
 - Violates our assumption of using constant random access latencies
 - SPEC CPU Benchmarks : Estimation error for ART/MCF > 20 %
- Adopt Empirical Model
 - $IPC = 1/(af+b)$, a, b are constants and f is frequency
 - Disadvantage : At least two samples are required
- We use combination of original and empirical model
 - Use empirical model when error exceeds certain threshold with original model

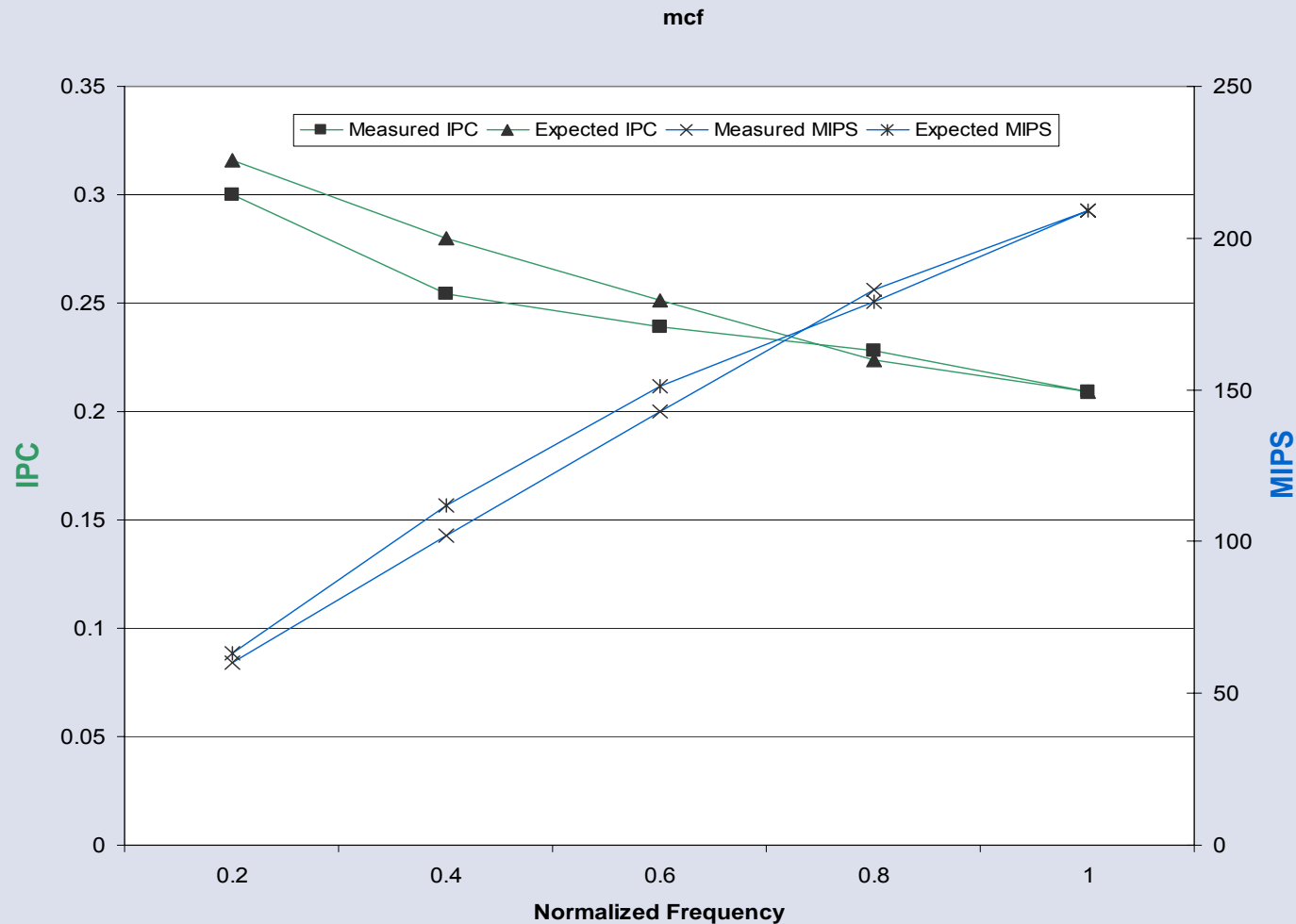
Empirical Model Accuracy : Art

Run on Power4 processor with f (Nominal) = 1GHz with avg. error < 6 %



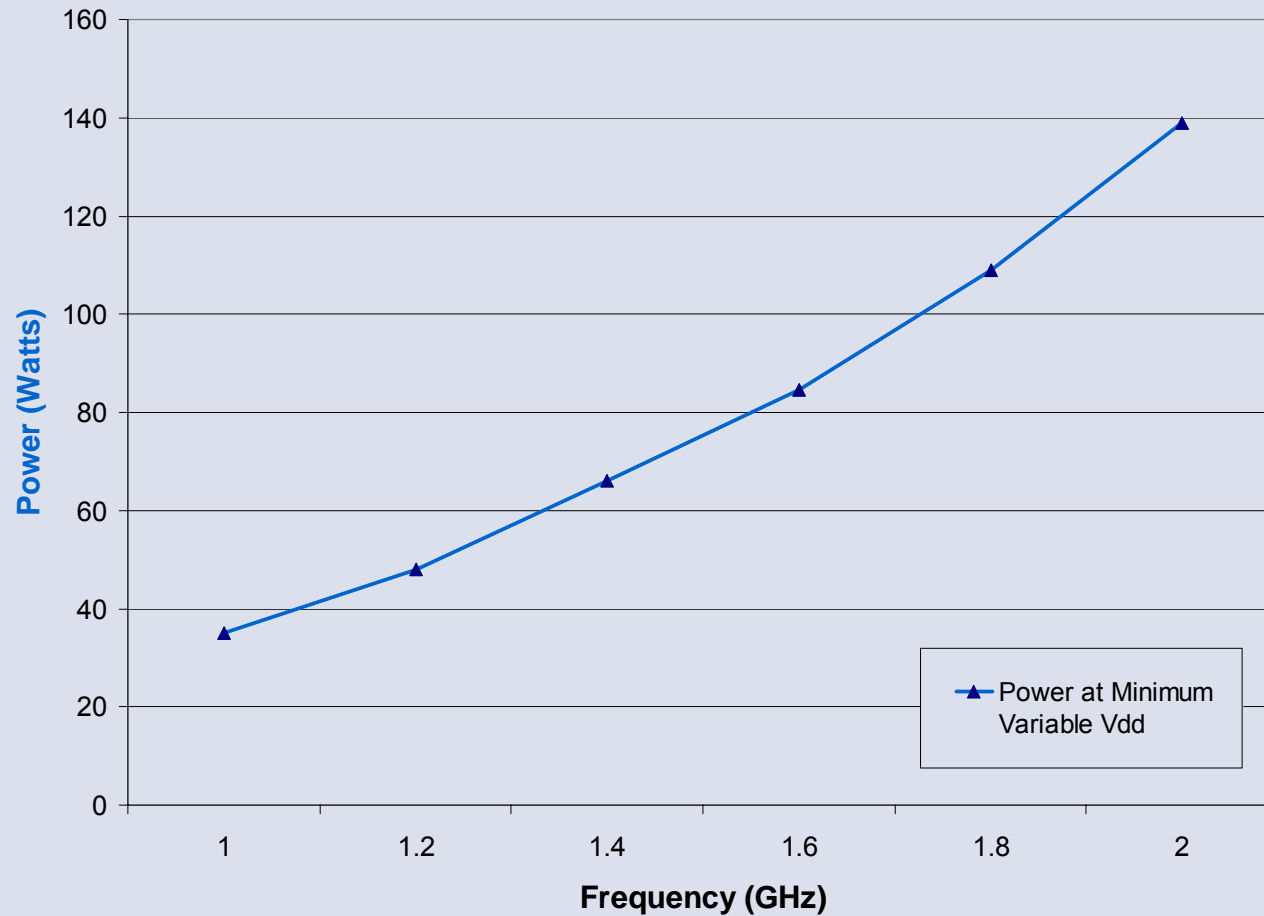
Empirical Model Accuracy : MCF

Run on Power4 processor with f (Nominal) = 1GHz with avg. error = 6.2%



Power4: Power at different frequencies/voltages

Using IBM's Lava power estimation tool



Power/Performance Characteristics

When Power 4 processor is run at 80% of nominal frequency ,
58% reduction in power can be achieved

Benchmark	Performance Degradation
ART	13.5%
MCF	12.5%
GZIP	20%
SPECJBB	14%
TPCW	20%

Conclusions

- Exploit application characteristics to reduce power consumption
- Accurate performance model using basic performance counters
 - Predict performance at different frequencies
 - Help scheduler to decide at which frequency to run