Introducing the Adaptive Energy Management Features of the Power7 Chip

POWER7 IMPLEMENTS SEVERAL NEW ADAPTIVE POWER MANAGEMENT TECHNIQUES WHICH, IN CONCERT WITH THE ENERGYSCALE FIRMWARE, LET IT PROACTIVELY EXPLOIT VARIATIONS IN WORKLOAD, ENVIRONMENTAL CONDITIONS, AND OVERALL SYSTEM USE TO MEET CUSTOMER-DIRECTED POWER AND PERFORMANCE GOALS. THESE INNOVATIVE FEATURES INCLUDE PER-CORE FREQUENCY SCALING WITH AVAILABLE AUTONOMIC FREQUENCY CONTROL, PER-CHIP AUTOMATED VOLTAGE SLEWING, POWER CONSUMPTION ESTIMATION, AND HARDWARE INSTRUMENTATION ASSIST.

Michael Floyd
Malcolm Allen-Ware
Karthick Rajamani
Bishop Brock
Charles Lefurgy
Alan J. Drake
Lorena Pesantez
Tilman Gloekler
Jose A. Tierno
Pradip Bose
Alper Buyuktosunoglu
IBM

Managing the power and performance trade-off of a running computer system is complex. Power7 has many low-level knobs for power management, but these also affect performance, depending on the type and combination of workloads being processed at a given time. Because there's no "one size fits all" policy, IBM's EnergyScale approach[1] employs an adaptive solution that encompasses hardware, firmware, and systems software.[2] A dedicated off-chip microcontroller, coupled with policy guidance from the customer and feedback from the Power Hypervisor and operating systems, determines operation modes and the best power and performance trade-off to implement during runtime to meet customer goals. Power7, like its predecessor Power6, provides the more traditional dynamic energy savings techniques such as clock-gating circuits when they're not needed, runtime scaling of frequency and voltage to adjust to varying use, and sensors to measure the environment and workloads under which the chip is operating.[3] This article describes several adaptive energy management features added to Power7 to augment these capabilities, presents empirically measured results of using these features, and discusses autonomic frequency-control capabilities that will provide further improved energy efficiency in the future.

Functional overview

The Power7 chip features eight processor *core chiplets*, each consisting of the processor core with its associated Level 2 (L2) and Level 3 (L3) caches, which are all fed by a common external power source that is controlled separately from the power source feeding the rest of the chip. For modularity and design reuse, each core chiplet runs asynchronously to the symmetric multiprocessor (SMP) interconnect fabric. For optimal

Published by the IEEE Computer Society

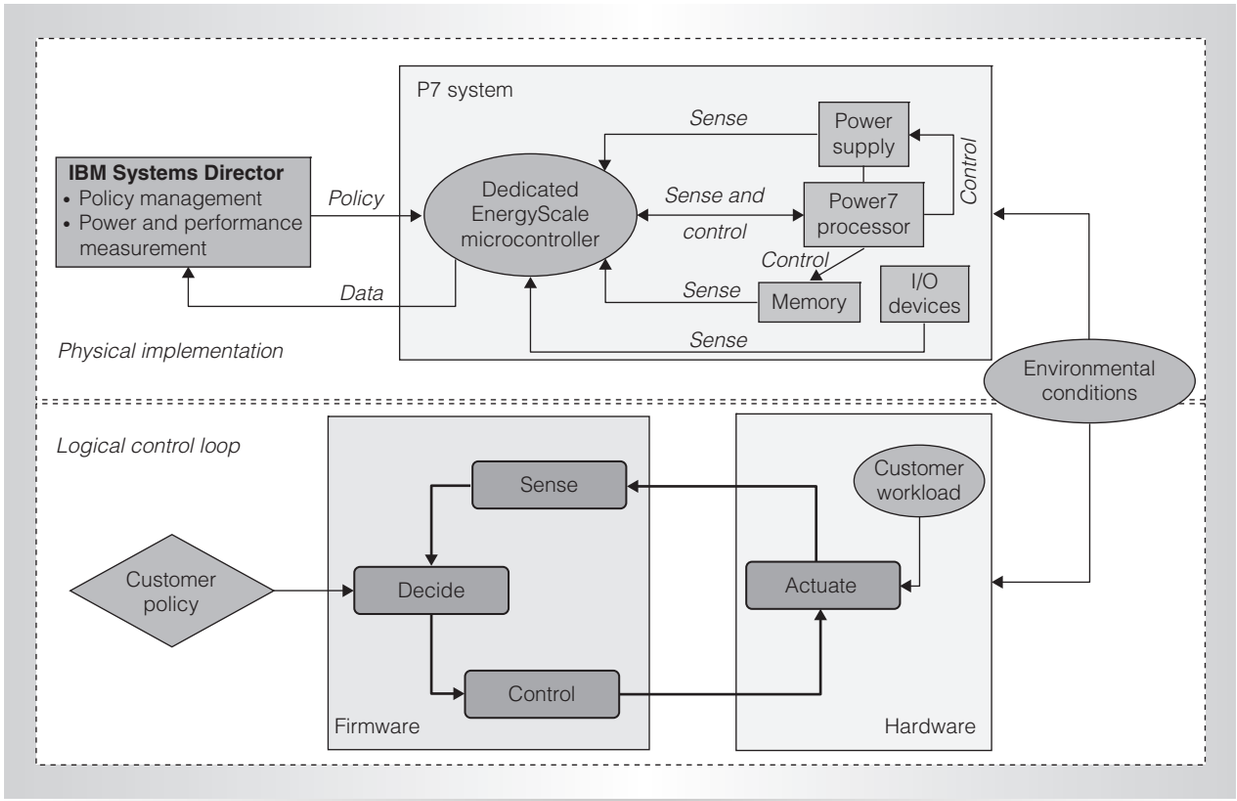

Figure 1. Physical and logical view of the Power7 energy management structure. The IBM Systems Director sends policy and management information from the customer to the EnergyScale microcontroller. It also controls the Power7 system operating parameters during runtime.

power, thermal, and yield concerns, we power the static RAM (SRAM) and embedded DRAM (eDRAM) array circuits with a separate, slightly higher voltage (Varray) than the voltage (Vlogic) used by the rest of the logic circuits in the core chiplet. This means the system must manage two voltages independently of the voltage supplied to the rest of the chip. At the system level, a dedicated external EnergyScale microcontroller communicates with the Power7 chip through a dedicated link that allows direct access to power management controls and sensors.

The dedicated EnergyScale microcontroller's firmware works with the system's other firmware elements to provide a total system energy management solution. It receives policy and management direction from the customer via the IBM Systems Director interface, while sensing and controlling the Power7 system's elements directly during runtime and reacting to changes in environmental conditions such as temperature and workload. Figure 1 shows a physical and logical view of the energy management structure.

Operating system software and hypervisor firmware together dynamically manage the workload running on the machine. The physical hardware is affected by the operating environment in which it is running in addition to the workload executing on it. Logically, the EnergyScale firmware runs a control loop with three steps:

1. Sense the state of the hardware, including workload and environmental conditions.
2. Decide which trade-offs to make on the basis of policy direction from the customer.
3. Control the hardware's behavior to direct these changes.

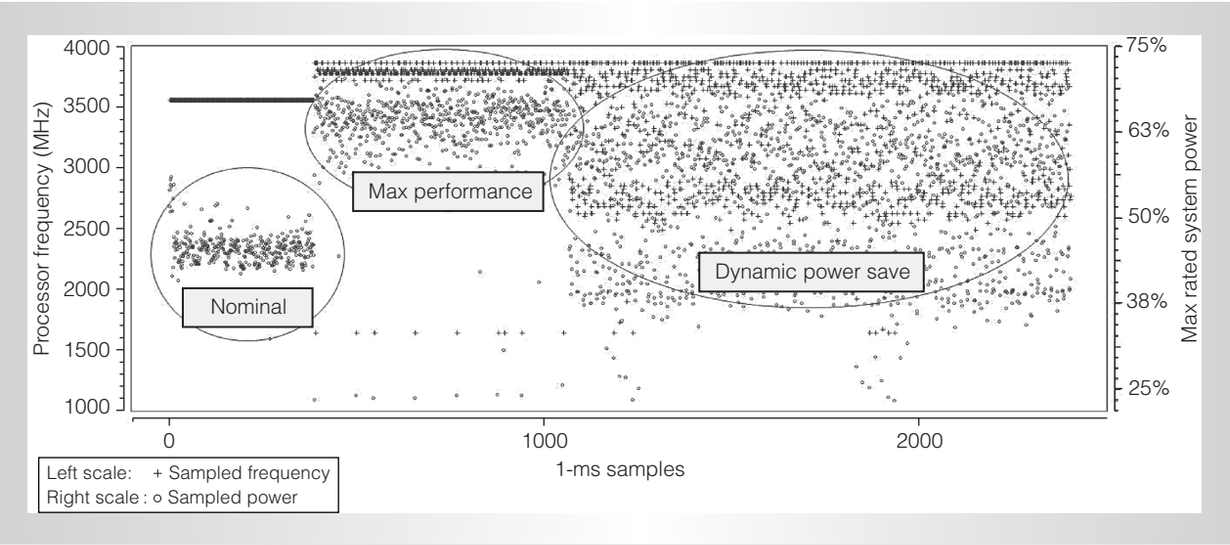

Figure 2. Empirical data illustrating the effect of EnergyScale policies on an IBM Power 750 Express Server. The frequency and voltage remain constant under nominal operation, are increased when maximum performance is required, and vary with workload under the Dynamic Power Save policy.

The hardware's response to the controls (labeled Actuate in Figure 1), along with any variations in workload and environment, changes the state of the system that can be sensed yet again.

Figure 2 illustrates the empirical results of a system under different EnergyScale policies. In this example, an IBM Power 750 Express Server is running representative customer code, which is a highly utilized scientific application with a varying workload profile.

Under nominal operation, the frequency and voltage remain constant so that power varies only on the basis of the workload. Under a maximum performance policy, the processor's average frequency—and therefore performance—is increased at the expense of higher system power (also called *turbo mode*). When the customer selects the Dynamic Power Save policy, the system's frequency and voltage—and therefore power—follow the workload's exact needs, resulting in increased energy efficiency (performance per watt consumed).

Instrumentation

Power7 provides several on-chip instrumentation enhancements to reduce the bandwidth and compute power required by the external EnergyScale microcontroller, allowing it faster response times and therefore more efficient energy management.

Temperature calculation

Each Power7 chip contains 44 digital thermal sensors (DTSs)—five per each processor core chiplet, and four in the SMP link and memory controller regions. They're located near predicted hot spots on the chip, as Figure 3 shows.

In the DTS design, a temperature-sensing circuit uses a band gap diode voltage comparator similar to previous designs.[4-6] What's new in Power7 is that the voltage reading is converted by the chip's logic to a temperature value in degrees Celsius via a polynomial curve fit ($px^2 + mx + b$, where x is the output of the comparator after analog-to-digital conversion). The coefficients are derived during manufacturing tests and calibrated per DTS using a traditional off-chip thermal diode as a reference, centered on the chip's expected operating temperature range from 65 degrees Celsius to 80 degrees Celsius. The firmware can then use the temperature values from the hardware directly without expending valuable real-time computing resources. The firmware uses the temperature measurements to guide EnergyScale decisions, tune the power proxy, and prevent the chip from possible overheating during

the dynamic (that is, turbo) modes of operation.

Critical path monitors

A critical path monitor circuit (CPM) is co-located with each DTS to provide real-time feedback on the chip's current timing margins. The CPM is a critical path synthesis circuit based on a design first included experimentally in the Power6 processor.[7] Figure 4 shows a high-level block diagram of the circuit.

The CPM uses four delay paths with different mixes of field-effect transistor (FET) and wire delay to approximate the different critical paths that will dominate in the microprocessor over its operating range. At clock cycle n, a pulse is launched down the delay paths and then captured in a 12-bit edge detector on the following clock cycle, $n + 1$. The penetration of the edge into the 12 bits indicates the circuit's timing margin at the given operating point. The pulse's delay through the synthesis paths is a function of a number of varying processes (such as voltage, temperature, workload, and age) that affect the operation of the circuit. In other words, the CPM provides a direct measurement of the chip's operating environment on a given clock cycle and how variation in that environment is affecting the current timing margin of that region of the chip.

Sensor consolidation

EnergyScale firmware accesses sensor information and sets controls on the processor via an on-chip serial communication infrastructure that provides 64-bit registers in a memory-mapped, 32-bit address space. An industry-standard Inter-Integrated Circuit (I^2C) interface provides connectivity from the EnergyScale microcontroller to this communication infrastructure on each Power7 chip. To reduce the number of transactions required to obtain power-management-related data, we packed multiple sensors into single read operations. In addition, by implementing a multicast read function, we further compact data from multiple chiplets into a single read operation, and certain controls can be written to multiple chiplets in a single write command.

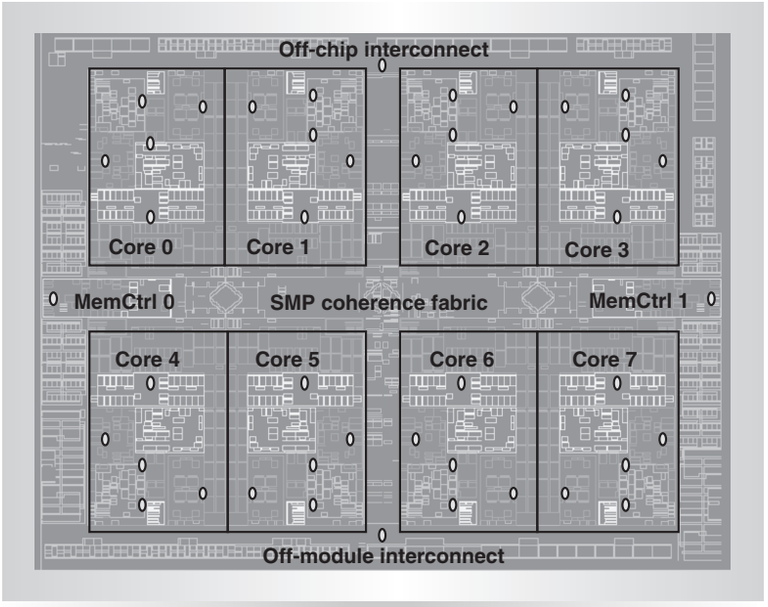

Figure 3. Digital thermal sensors marked with their physical location on the Power7 chip floorplan. Each Power7 chip contains 44 DTSs.

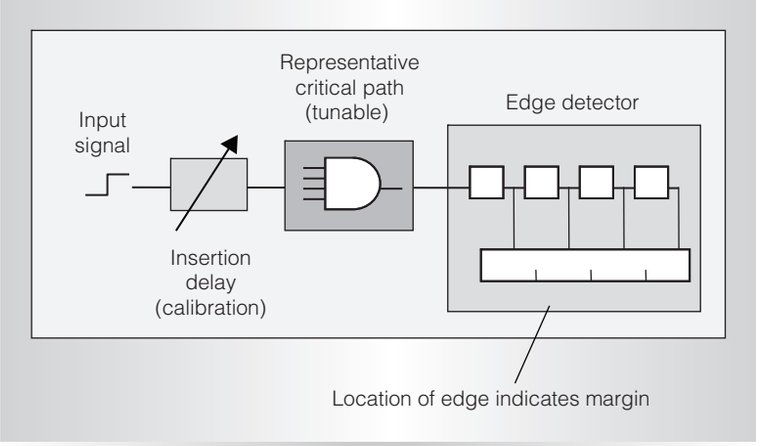

Figure 4. Block diagram of the critical path monitor circuit. The signal's penetration depth into the edge detector indicates how much timing margin exists for the chip's circuitry to operate.

Given the limited bandwidth of the I^2C bus and the growth in sensor traffic from the Power6 generation—both in the type and number of items tracked—we also needed an efficient access method for the data. Power7 implements an automated on-chip communications assist mechanism in the form of a transaction table that periodically collects performance and thermal sensors from around the chip. Once the latest

set of sensor data is consolidated in a central location, the EnergyScale microcontroller can stream it out as a single I^2C transaction, eliminating almost all of the I^2C protocol overhead that individual I^2C transactions would entail. The on-chip communications assist macro built into each Power7 chip allows for highly synchronized access to all the core chiplets such that the processor utilization measurements are done with precision, with at most 0.1 percent error.

Per-core asynchronous frequency scaling capability

Because previous Power systems' cores ran synchronously to the SMP interconnect, EnergyScale firmware had to change the reference clock's frequency for the entire system in order to change the processor cores' frequency. This system-level process was relatively slow and affected every core and cache in the system, making it suboptimal for adaptive power management algorithms. In addition, this method often constrained the EnergyScale firmware because the attached support chips (such as I/O controllers) also ran synchronously to this SMP interconnect but could only operate within certain frequency ranges, that differed according to the type and version of support chips present in the given system. In addition, multiple processor cores on a chip traditionally run off a shared analog phase-locked loop (APLL), a design that's not amenable to frequency scaling. To work around this problem, some systems have incorporated dynamically selectable, fractional-frequency operation (such as a programmable M out of N cycles) to downstream circuits either internal or external to the APLL.[8] With the scale-up to eight cores on a chip, running all cores at the same frequency or even from a subset of independently configurable fractional frequencies off the same reference would make the EnergyScale firmware less able to match power efficiency to the demand (utilization and workload type) running on each core. Power7 leverages the asynchronous core chiplet design by supporting a 50 percent to 110 percent change from nominal frequency, while minimizing latency by using a synchronous-write, asynchronous-read array queuing structure across the chiplet interface. To provide this flexibility, we designed employed a new circuit to let EnergyScale control frequency independently and dynamically during runtime on a per-core chiplet basis.

This new circuit in Power7, a per-core chiplet variable frequency generator, is built around a fractional-N digital phase-locked loop (DPLL).[9] The DPLL contains a digitally controlled oscillator (DCO), where the output frequency is selected as a multiplier off a reference clock base. An internal DPLL filter ensures that no overshoot or undershoot in cycle time is present in the transient during requested frequency changes. The combination of the controlled-frequency change rate and the absence of short cycles allows Power7 to use this DPLL-based variable frequency generator to dynamically adjust the core chiplet's frequency while the core is fully operational, executing code. The DPLL can change frequency in excess of 50 Mhz per microsecond (μs) across the full range of supported frequencies. Consequently, the Power7 sees dramatic improvement in both slew rate and frequency range over the Power6, in addition to providing the capability on a per-core basis.

Figure 5 shows two scenarios where the per-core frequency scaling capability with DPLL provides increased energy efficiency. Scenario1 is when a single core is busy at 100-percent load while the other seven are also running, but with their workloads' demands met by running at the minimum allowed frequency. Scenario2 is when a single core is busy in a chip and seven others are napping. While both scenarios show energy reductions with the per-core scaling, the higher benefits for Scenario 1 show the DPLL's unique value for energy-efficiency improvements when all cores are active.

Processor idle states

In the Power architecture, the hypervisor controls the entry to and exit from processor idle states through state-specific privileged instructions. Operating systems guide the hypervisor on which idle state to use based on expected idleness for the processor thread (and ultimately the core) through specific

hypervisor calls. The Power7 processor supports two distinct idle states: Nap and Sleep.

The Nap idle state quiesces instruction fetch and execution and turns off all clocks to the execution engines in the core. This addresses power in latches not already turned off naturally by local clock gating when the execution pipelines go idle. Nap on Power7 keeps the caches (L1, L2, and L3) coherent and doesn't purge them.

Sleep is a more aggressive idle mode than Nap, in that it clocks off the entire core chiplet including the caches and even its own clock source (the DPLL). Coherency is maintained by purging L1, L2, and L3 caches to memory and invalidating all translation lookaside buffers before entering this state. Upon waking from Sleep, the hardware logic restarts the DPLL, powers up the L3 cache's eDRAM, and sequences the core chiplet back to functional operation. Power7 allows additional power reduction for Nap and Sleep by supporting automated coupling of the on-chip frequency and voltage-slewing capabilities with the processor idle states. For Nap, hardware logic selectively supports scaling down frequency to a preprogrammed lower value on Nap entry and ramping up to the operational frequency value on Nap exit. Since this lowers power for still-clocked components on the chiplet, such as the caches, it can potentially also lower the performance of data accesses into these caches from other still active cores. When all the cores on a package the chip are in Sleep, the hardware logic can be configured to automatically lower the external voltage to *retention voltage*, a low-voltage, reduced leakage level at which latches and arrays can continue to retain data.

Figure 6 shows some typical latencies associated with exiting the processor low-power modes and range of measured chip-level power reduction relative to idling at the nominal operating point with the operating system polling for work. Nap latency is typically around 2 μs; Sleep latency as observed by an application is typically less than 1 ms, where its minimum duration is dominated by the eDRAM re-enablement process upon exit and its maximum duration is dominated by the L3 cache purge on entry. When voltage for the chiplets is dropped to

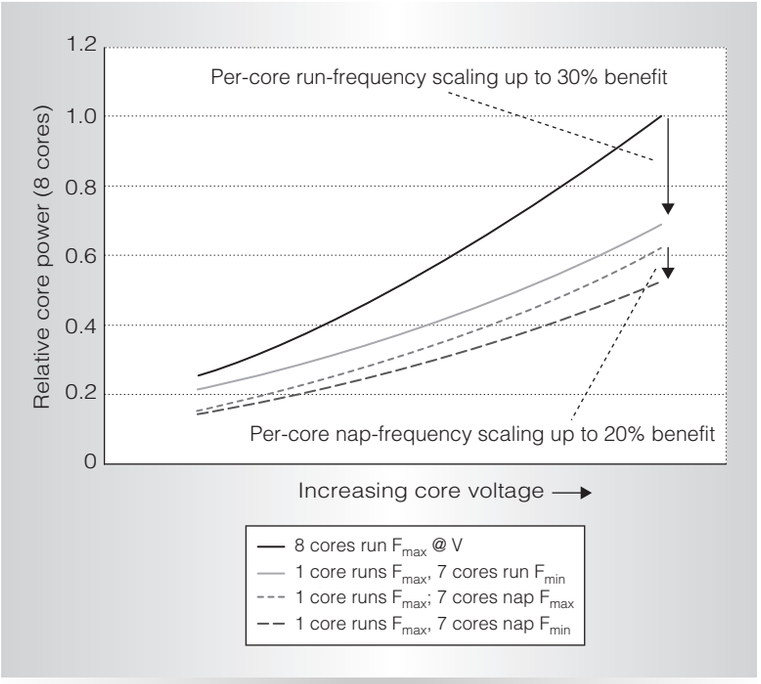

Figure 5. Benefit of per-core frequency scaling with the digital phase-locked loop (DPLL). In both scenarios, scaling with the DPLL provides increased energy efficiency. The benefit is highest when all cores are running but only one core is executing an intensive workload.

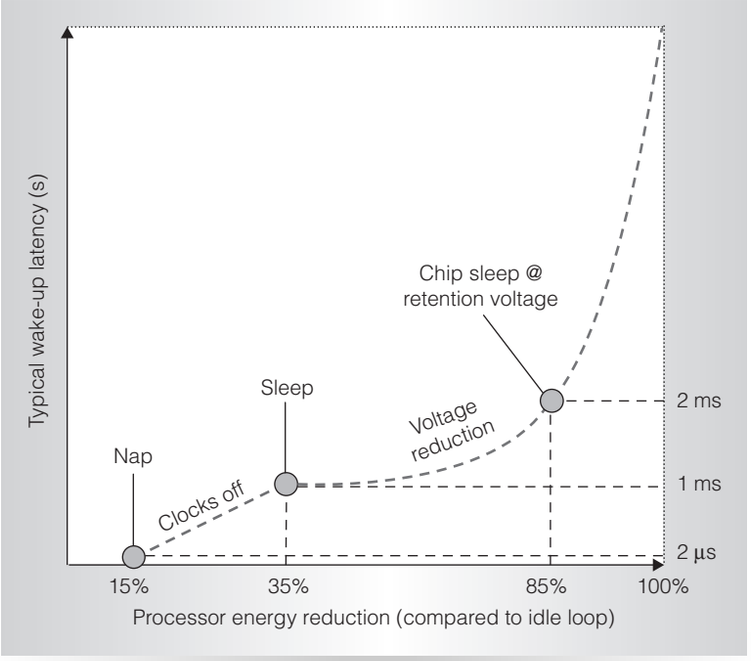

Figure 6. Processor idle states. Here we show some typical latencies associated with exiting the processor low-power modes and their corresponding amount of power reduction relative to idling at the nominal operating point with the operating system polling for work. (Vret: retention voltage.)

retention level, typical resumption latency increases to 2 ms due to the voltage slew back to the desired operating point.

Automated voltage-slewing controls

To support reducing the voltage to retention for an all-core Sleep state in the required sub-10-ms timescale, Power7 includes integrated voltage control for both the array voltage (Varray) and logic voltage (Vlogic) feeding the core chiplets. The on-chip controller automatically selects between the idle and operational target voltage. By communicating to the Power7 chip, the EnergyScale microcontroller can request the on-chip controller to adjust the operational voltage in response to changes in environment, workload, and customer policy.

Industry-standard voltage regulator modules (VRMs) can safely process only small, instantaneous voltage change requests. Power6 EnergyScale firmware spent hundreds of milliseconds sequencing large voltage change requests via multiple steps to elicit a smooth, stable VRM response and avoid voltage transients during the transition. Power7 chip hardware automates this function by providing a voltage sequencing engine, freeing up compute power on the microcontroller for more advanced power management algorithms. A change in the target voltage triggers the automated sequencer to step toward the new voltage level at a programmable rate, while maintaining the desired relationship between the Varray and Vlogic voltages.

Power7 low-end and mid-range systems contain VRMs external to the chip directly sourcing the two voltage rails to the core chiplets. They're controlled by an industry-standard 8-bit parallel interface (PVID), which expresses voltage in 6.25-mV (millivolt) increments. This technique is fast because it's limited only by the response time of the VRMs, which sample the PVID input in the megahertz range. High-end systems requiring greater power delivery and reliability use redundant power supplies with embedded controllers that support more sophisticated techniques such as current sharing and failover. In such systems, the sheer number of processors and resulting scarcity of available connector and board wiring make PVID interfaces impractical at the system level. Therefore, Power7 chips provide a serial control interface over I^2C to communicate to the power supplies' embedded controllers, which then perform the stepping across the full operational voltage range in a few milliseconds.

Power proxy

Effective power management in a microprocessor requires a runtime measurement of power. However, the measurement of real, calibrated power consumption in hardware is difficult. For example, physically measuring power might require periodically stalling the processor for proper calibration. Isolating power consumption on a per-core chiplet basis is simply not possible when multiple chiplets share the same power grid. Additionally, reliable internal power measurement circuits weren't available for the Power7 chip. In the absence of actual measurement, we developed a technique to provide an estimate, or proxy, for the amount of power each core chiplet was consuming.

Researchers have examined using existing hardware performance counter data, collected by software, to estimate core or chip power consumption.[10] However, previous research hasn't addressed choosing the right (and minimal) set of activity monitors and corresponding weights to estimate power at high accuracy. AMD has referred to an on-chip power monitoring and control facility,[11] and Intel has also reported a mechanism to choose the chip's global settings (such as voltage or frequency) by monitoring activity levels across various regions within the chip.[12] However, a comparison with our technique is not possible because these other designers haven't reported specific implementation concepts and detailed measurement-based performance (that is, accuracy analysis) data for their designs.

Power7 implements a hardware mechanism in the form of a power proxy by using a specially architected, programmably weighted counter-based architecture that monitors activities and forms an aggregate value. Activities are carefully selected with an understanding of the processor's

microarchitecture, such that they correlate maximally with active power consumption. Figure 7 shows a diagram of power proxy activity event collection in the Power7 processor core chiplet.

The activity events (A_i) in the processor core, L2 cache, and L3 cache are each multiplied by a programmable weight factor (W_i) and then added together, with a constant offset for empirical curve fit, into a single value representing average active power over a sampling interval, where $P_{act} = \sum (W_i \times A_i) + C$. The estimation occurs across a programmable timescale interval as small as a few microseconds. The event power P_{act} then adjusts for active power not associated with the available activity events (such as clock grid power and other frequency-dependent events) by multiplying the average measured frequency (F_{avg}) over the interval by a final constant to generate a chiplet *power proxy value*, where $P_{proxy} = P_{act} + (K \times F_{avg})$. By selectively weighting the different events and constants relative to each other on the basis of empirical post-silicon correlation work, we can estimate the amount of active power the chiplet consumes within an accuracy of a few percent. Power management firmware then adjusts the hardware activity-based power proxy for the effects of leakage, temperature, and voltage to estimate the chiplet's total power.

To perform a measurement-based weight calibration, we collect sample points for a series of targeted benchmarks such that all the power proxy events are represented. Each sample point contains a power measurement (for the core and L2 and L3 caches) and a count for each activity event. For the workloads' duration, sample points are taken periodically at granular intervals, such as 1 ms. Once we gather and post-process the data, we run genetic algorithms to determine the power proxy settings to provide an optimal curve fit.

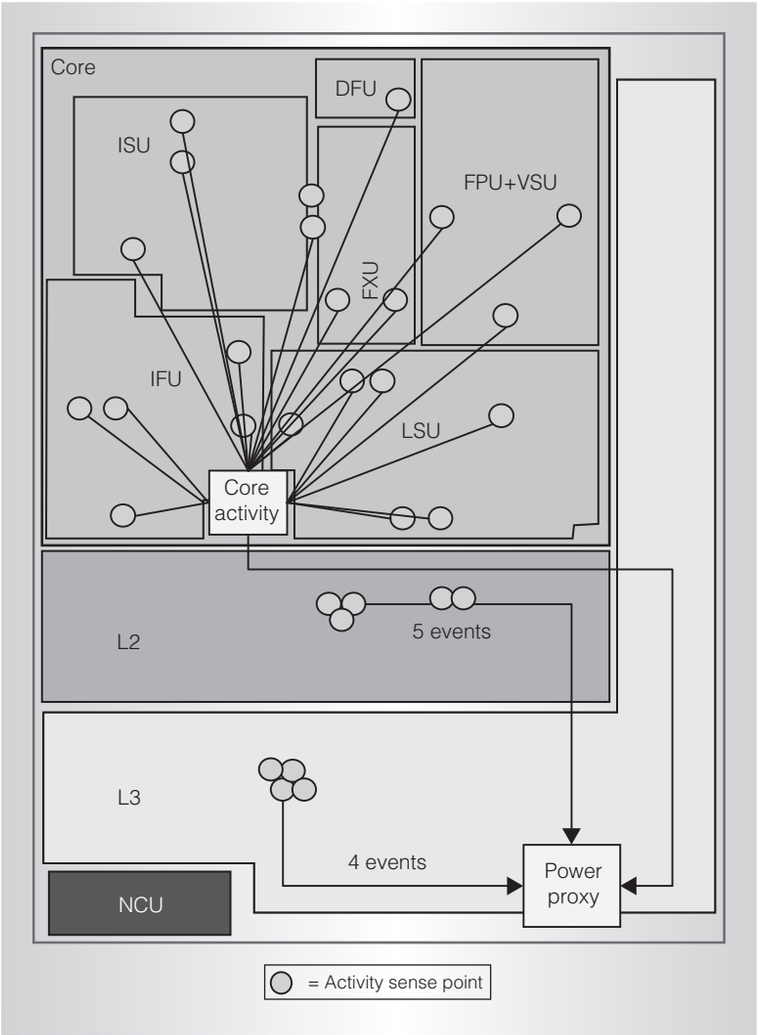

Figure 7. Diagram of power proxy activity event collection in the Power7 processor core chiplet. The various units are depicted on this representation of the physical floorplan, where circles indicate the sense points provided by those units. Units are named based on the function they provide. The Core section of the chiplet includes multiple Units: instruction fetch (IFU), instruction sequencing (ISU), fixed point (FXU), decimal float (DFU), load-store (LSU), and a combined floating point with vector-scalar extensions (FPU + VSU). The remainder of the chiplet consists of Level 2 (L2) and Level 3 (L3) cache units as well as the non-cacheable unit (NCU).

Empirical data in Figure 8 shows the results of the power proxy run against a range of sample workloads and microbenchmarks collected using real hardware. The *y*-axis shows power data normalized to the workload representing peak power, and the *x*-axis contains a set of disjoint samples from many independently run benchmarks. Empirical results demonstrate that the power proxy estimates are close to actual power consumption in most cases. In our initial attempts to calibrate against hardware, 91 percent of samples fell within plus or minus 10 percent relative error, and 73 percent of samples fell within plus or minus 5 percent error.

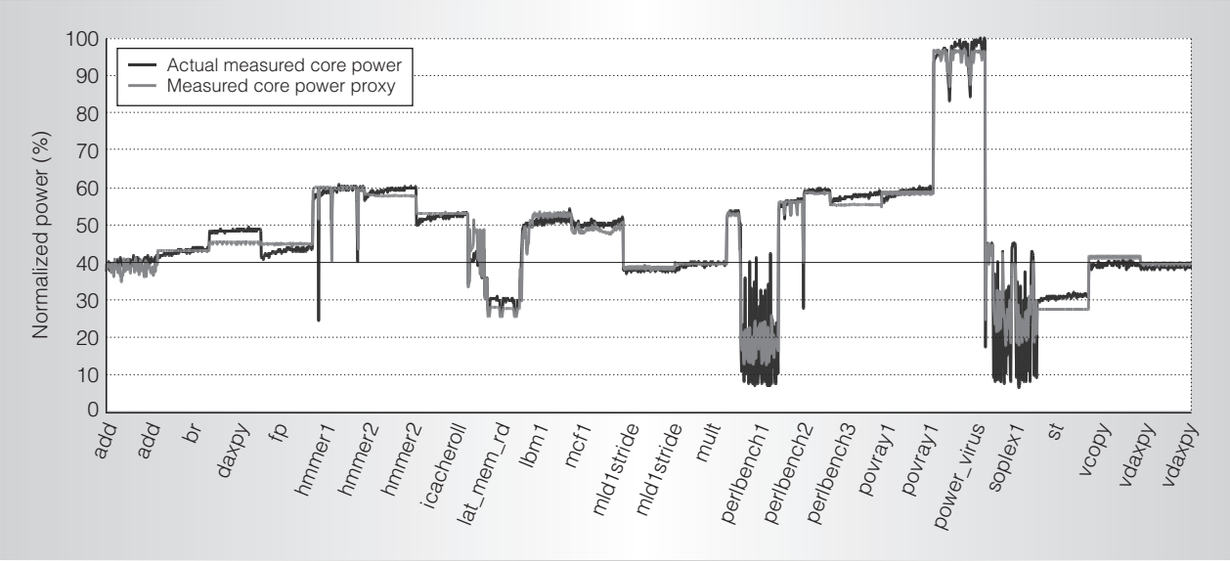

Figure 8. Empirically measured power versus estimated power on Power7 chip hardware using the power proxy mechanism. The *y*-axis shows power data normalized to the workload representing peak power, and the *x*-axis shows samples from independently run benchmarks. Empirical results demonstrate that the power proxy estimates closely match the actual power consumption in most cases.

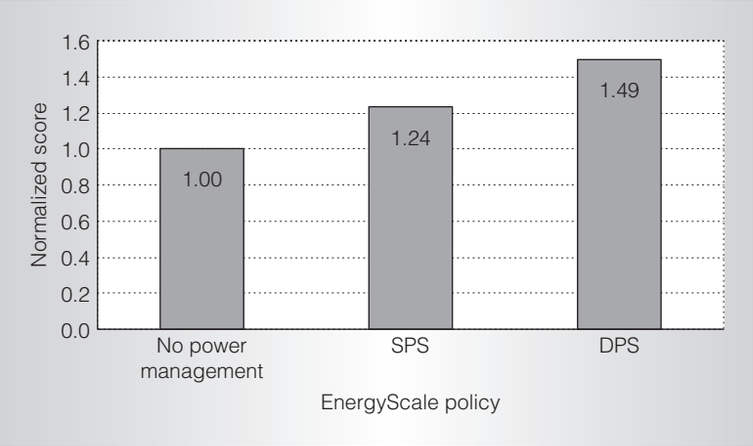

Figure 9. SPECpower_ssj2008 score by EnergyScale Policy. The score is normalized to the system running without power management enabled for both the Static Power Save (SPS) and Dynamic Power Save (DPS) policies. (Higher is better.)

Power efficiency algorithms and results

As with Power6, the Power7 EnergyScale microcontroller monitors several on-chip counters to measure system use and varies both the chip's frequency and voltage. Power7 servers add dynamic fan management, which reduces the fan's power consumption by regulating fan speeds to maintain a safe thermal setpoint as measured by the DTS located in the core chiplets and by other system components such as memory.

Power7's power management features significantly improve server energy efficiency, as shown by running the industry standard SPECpower_ssj2008 benchmark[13] on an IBM Power 750 Express server[14] with four Power7 processors with a nominal ship frequency of 3.55 GHz and 64 Gbytes of memory.

The SPECpower_ssj2008 benchmark is representative of server-side Java business applications. It measures system energy efficiency while injecting transactions into the server at load levels from 10 percent to 100 percent of calibrated maximum throughput. The final benchmark score is the ratio of the sum of system throughput to the sum of system power consumption, each sum being across all load levels. Figure 9 shows the normalized benchmark scores across three EnergyScale policies. The ''no power management'' case represents how the server would run without any EnergyScale power management firmware, at a static nominal operation and with dynamic fan control turned off.

The system AC power consumption for each benchmark load level is shown in Figure 10 and is normalized to the maximum power measured across all experiments.

The first dramatic improvements are found in the Static Power Save (SPS) EnergyScale policy. SPS uses a fixed processor frequency which is lower than nominal and incorporates dynamic fan management. This approach minimizes the power consumed by the processors and fans and leads to a significant jump of 24 percent in the score over the nominal system with no dynamic fan management. The score improvement comes from running the processors at 70 percent of nominal frequency, but realizing a system power consumption that is roughly 60 percent of the nominal system across all load levels (see Figure 10).

Using the Dynamic Power Save (DPS) EnergyScale policy boosts the score to 49 percent over no power management, doubling the improvement SPS made. DPS continues to use dynamic fan management but lets the processor frequency vary from 50 percent to 109 percent of nominal use in response to processor use. The effect of changing voltage and frequency under DPS mode is visible at loads greater than 50 percent (had maximum performance ''turbo mode'' been disabled, the DPS and ''no power management'' point at 100-percent load would coincide). The DPLL provides a minimum frequency stepsize of 28 MHz (one-eighth of the 224-MHz reference clock input), and decisions are made approximately every 100 ms, which enables a fine degree of control. This lets us minimize the global frequency selection to match the needs of the SPECpower_ssj2008 load level, which further results in the minimum possible chip voltage to complete the required work.

DPS runs at 109 percent of nominal frequency under heavy workloads, which allows the benchmark throughput overall to be increased by about 7 percent over nominal frequency owing to a higher work injection rate being selected during the benchmark's calibration phase. The throughput for DPS was 41 to 45 percent higher across all load levels when compared to SPS. At the 50 percent load level and below, the system power consumption is nearly identical for DPS and SPS, even though DPS yields a higher throughput. Although DPS consumes more power than SPS above the 50 percent load level owing to higher frequency selection, this is more than offset by the higher work injection rate across all load levels, and lets DPS achieve better throughput-per-watt efficiency.

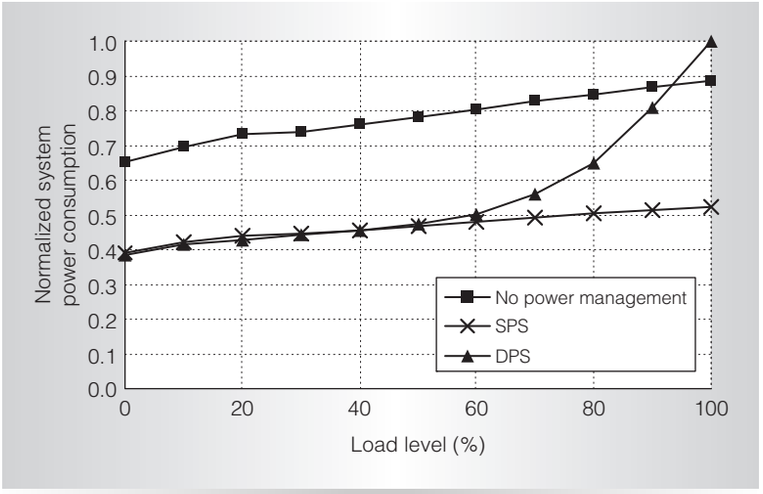

Figure 10. System AC power consumption per load level for SPECpower_ssj2008. The power consumption at each point is normalized to the maximum power consumed by the system for this benchmark, which occurs with the DPS policy under full load (in turbo mode). The system running without power management is noticeably less energy efficient than either the SPS or DPS policies.

Autonomous frequency control

The per-core asynchronous frequency scaling capability also lets Power7 implement several new autonomous frequency control mechanisms beyond what was included in the results we described earlier. Autonomous frequency controls move processor core chiplet frequencies within a defined range in response to operating conditions, exploiting the fine-grained, low-latency frequency control provided by the per-core DPLLs. The upper end of the frequency range is typically the nominal operating frequency for the current voltage, and the lower end of the range varies depending on the power management policy and system-level timing constraints. Given the common voltage planes shared by all cores in a Power7 chip, per-core autonomous frequency scaling can be effective for managing power in a virtualized

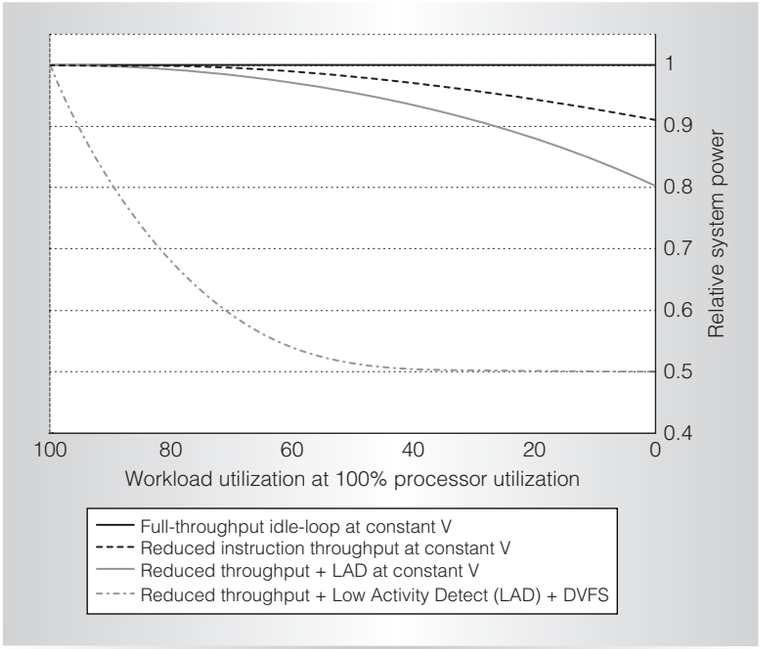

Figure 11. Benefit of frequency scaling available to the low-activity detection mechanism. (DVFS: Dynamic Voltage and Frequency Scaling.)

environment where interpartition constraints could preclude lowering common voltages.

Low-activity detection

The Power7 low-activity detection (LAD) mechanism autonomously manages core chiplet frequencies in response to workload changes observed on intervals as short as a few microseconds. LAD is triggered by changes in per-core utilization or performance, typically derived from instructions-per-cycle (IPC) measurements when the processor is executing a workload. LAD hardware lowers core chiplet frequencies as long as the triggering metric is below a programmable threshold. For example, if this mechanism observes an average IPC of less than 0.25 over an interval of 32 μs the EnergyScale firmware may program this mechanism to drop core frequency to 50 percent of nominal. The LAD mechanism permits rapid frequency scaling to exploit low activity periods that are too short for firmware or system software to detect. Prior proposals in this context have been primarily software driven and consequently unlikely to take advantage of low activity at very fine timescales.[15-17]

An example of LAD's potential arises in scalable server systems using message passing or user-level remote direct memory access protocols that employ busy-wait polling strategies for service requests. This leads to cases where a system that's 100 percent busy by traditional metrics might actually be 100 percent idle in terms of the useful work being accomplished, defeating traditional use-based dynamic voltage and frequency scaling (DVFS) energy-savings algorithms. Polling servers that know they're idle can immediately reduce their energy consumption by executing strategies to purposefully reduce instruction throughput while still maintaining an acceptable polling rate. The LAD mechanism will then automatically trigger frequency scaling in response to the reduced throughput, further reducing energy consumption. Finally, software control loops executing on the EnergyScale microcontroller can also lower the maximum frequency and associated voltages suitably after having observed that the LAD logic has reduced the effective average frequency. This "green polling" methodology allows for significant energy reductions in lightly used servers with only small increases in transaction latencies.

Figure 11 illustrates the energy savings possible with LAD and a Green Polling technique for a Power7 system. Application-directed reduced instruction throughput coupled with LAD can provide up to a 20-percent reduction in system power as workload usage drops. Aggressive DVFS policies can produce an additional 30-percent reduction.

Reducing wasteful guard band

We choose a microprocessor's voltage levels during the manufacturing test and characterization process on the basis of its intended operating frequency range. We set the voltages conservatively so that the microprocessor will operate at a target frequency in high-temperature environments over all expected workloads with a sufficient circuit-timing margin. Typically, we include some additional guard band voltage to cover for unknown variables and test inaccuracy.

In a Power7 system, we can sense the available timing margin at runtime using

the CPM instrumentation and adjust the guard band dynamically to meet changing environmental conditions and workloads. Each cycle the CPMs determine the minimum sensed timing margin across the core chiplet. The DPLL implements a proportional feedback loop based on the combined CPM outputs to control the frequency during runtime, slowing it down if the timing margin is too small and speeding it up if the timing margin is larger than necessary to achieve a comfortable guard band. This forms the CPM-DPLL feedback loop, a circuit that dynamically compensates circuit timing in response to runtime variation.

We can use this reduced guard band either to boost performance by overclocking the microprocessor or to reduce power consumption by undervolting the microprocessor. Figure 12 shows overclocking results for running the SPECpower_ssj2008 (100-percent load level only) on a typical Power7 chip. The dynamic guard band results are normalized to the static case's performance throughput as measured by the workload's transaction rate, and both cases use the same turbo voltage level. The static guard band case shows the server running the DPS policy (frequency is 109 percent of nominal) using the traditional guard band. The dynamic guard band case shows 7.3 percent more throughput because of the CPM-DPLL feedback loop selecting to run at an even higher frequency (at a constant voltage). A hand-tuned frequency, which is empirically determined by when this particular chip actually fails, is only slightly higher. This small remaining guard band is necessary because the CPMs' calibration, tracking to the real critical circuits, and location in the floorplan are not perfect.

Figure 13 shows an example of undervolting using an experimental version of EnergyScale firmware that lowers voltage while meeting the frequency target set by the DPS policy. The entire system's power consumption (normalized to the static case) drops by 15.8 percent without affecting performance because frequency holds constant. Using hand-tuned voltage reduces power consumption by 26.4 percent before the chip fails. This result was not surprising because unlike the remaining frequency guard band, which has a largely linear affect on performance, the unclaimed voltage guard band has a more than quadratic relationship with power. Regardless, in both the overclocking and undervolting cases, large amounts of the traditional guard band are reclaimed.

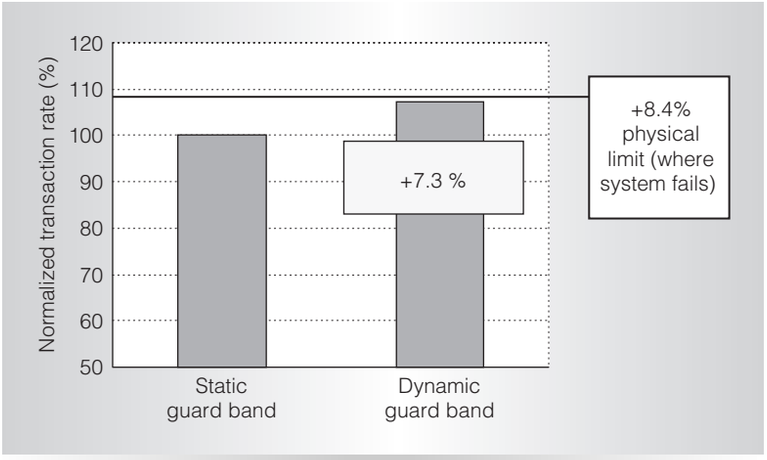

Figure 12. Overclocking results for the SPECpower_ssj2008 on a Power7 chip. The dynamic guard band case shows 7.3 percent more throughput because of the critical path method (CPM-DPLL feedback loop) selecting to run at a higher frequency (at a constant voltage). A hand-tuned frequency representing zero guard band only allows for 8.4 percent improvement before this particular chip fails.

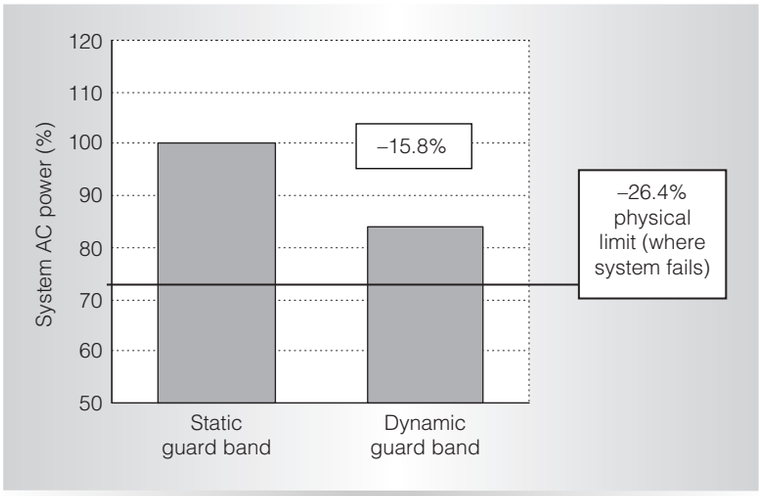

Figure 13. Undervolting results for an experimental version of EnergyScale. Frequency stays constant, allowing the system's power consumption to drop without affecting performance. The critical path method reclaimed almost 16 percent of the normal guard-banded system power.

As we've just illustrated, energy efficiency can best be achieved by either

reducing power consumption while maintaining the same amount of performance or by increasing performance at the same power level. The direction for future EnergyScale hardware and firmware development will be to build upon the capabilities and results achieved on Power7 to do just that. The effort is centered on increasing performance to stay just inside the system's power delivery envelope combined with dramatic power reduction for inactive portions of the computing system. This focus enables higher performance from systems than is being achieved today by shifting power from idling portions of systems to those that are active and will benefit from running faster.

To reach this goal will require better instrumentation, more intelligent algorithms, finer-grained voltage controls, increased guard band reduction techniques, smaller idle latencies, and faster EnergyScale firmware response time. For example, implementing support for a deeper-architected idle mode (named Winkle in the Power ISA) lets us completely shut off the power to a processor core or a core chiplet when it is idle. Faster wakeup latencies will allow the OS and Hypervisor software stacks to invoke the idle instructions more often and under more scenarios. Incorporating more intelligence and algorithms natively into the host (OS and Hypervisor) code can enable greater energy efficiency by better synchronizing resource scheduling with resource state management. Better support for main memory usage monitoring and power control will help address a growing component of server power. The ability to scale voltage with frequency on a finer-grained basis and with lower response times will provide much better DVFS capability to match the power of the core with the workload it is running. More advanced algorithms coupled with faster real-time EnergyScale firmware (such as executing in the microsecond instead of millisecond range) will allow energy efficiency to track more closely with the exact needs of the workload. The challenge will be to match the best combination of these techniques to the policy selected by the user. MICRO

Acknowledgments

Some material in this article is based on work supported by DARPA under agreement HR0011-07-9-0002. Statements regarding chip and system features do not imply that IBM will introduce a system with this capability.

References

1. M. Broyles et al., ''IBM EnergyScale for Power7 Processor-Based Systems,'' white paper, IBM, Aug. 2010.
2. H.-Y. McCreary et al., ''EnergyScale for IBM Power6 Microprocessor-Based Systems,'' *IBM J. Research and Development,* vol. 51, no. 6, 2007, pp. 775-786.
3. M.S. Floyd et al., ''System Power Management Support in the IBM Power6 Microprocessor,'' *IBM J. Research and Development,* vol. 51, no. 6, 2007, pp. 733-746.
4. M. Yoshida and D.W. Boerstler, *Thermal Sensing Circuit Using Band Gap Voltage Reference Generators without Trimming Circuitry,* US patent 7,789,558, to Toshiba and IBM, Patent and Trademark Office, 2010.
5. T.H. Lee, M.G. Johnson, and M.P. Crowley, *Temperature Sensor Integral with Microprocessor and Methods of Using Same,* US patent 5,961,215, to Advanced Micro Devices, Patent and Trademark Office, 1999.
6. J.G. O'Dwyer, *Apparatus and Method for Temperature Measurement Using a Band Gap Voltage Reference,* US patent 7,225,099, to Xilinx, Patent and Trademark Office, 2007.
7. A. Drake et al., ''A Distributed Critical Path Timing Monitor for a 65-nm High-Performance Microprocessor,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE Press, 2007, doi:10.1109/ISSCC.2007.373462.
8. J. Dorsey et al., ''An Integrated Quad-Core Opteron Processor,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE Press, 2007, doi:10.1109/ISSCC.2007.373608.
9. J.A. Tierno, A.V. Rylyakov, and D.J. Friedman, ''A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI,'' *IEEE J. Solid-State Circuits,* vol. 43, no. 1, 2008, pp. 42-51.

10. C. Isci and M. Martonosi, ''Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data,'' *Proc. 36th Ann. IEEE/ACM Int'l Symp. Microarchitecture,* IEEE CS Press, 2003, doi:10.1109/MICRO.2003.1253186.
11. R. Jotwani et al., ''An x86-64 Core Implemented in 32nm SOI CMOS,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE Press, 2010, doi:10.1109/ISSCC.2010.5434076.
12. B. Stackhouse et al., ''A 65-nm 2-Billion Transistor Quad-Core Itanium Processor,'' *IEEE J. Solid-State Circuits,* vol. 44, no. 1, 2009, pp. 18-31.
13. *SPECpower_ssj2008,* version 1.10, Standard Performance Evaluation Corp., Warrenton, VA, 2010.
14. *IBM Power 750 Express Server,* IBM Systems and Technology Group, Somers, NY, 2010.
15. K. Choi, R. Soma, and M. Pedram, ''Dynamic Voltage and Frequency Scaling Based on Workload Decomposition,'' *Proc. 2004 Int'l Symp. Low Power Electronics and Design,* ACM Press, 2004, pp. 174-179.
16. G. Semeraro et al., ''Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling,'' *Proc. 8th Int'l Symp. High-Performance Computer Architecture,* IEEE CS Press, 2002, pp. 29-40.
17. N. Abou Ghazaleh et al., ''Integrated CPU and L2 Cache Frequency/Voltage Scaling Using Supervised Learning,'' *Proc. 2007 ACM SIGPLAN/SIGBED Conf. Languages, Compilers, and Tools for Embedded Systems,* ACM Press, 2007, p. 41-50.

Michael Floyd is the architect and lead for the Power7 EnergyScale design in the IBM Systems and Technology Group. His work in IBM server development has included hardware bring-up, test, debug, and reliability, in addition to design, lead, and microarchitect roles on the Power4, Power5, and Power6 processors and support chips. Floyd has an MS in electrical engineering from Stanford University.

Malcolm Allen-Ware is a member of the Power-Aware Systems Department at the IBM Austin Research Lab. His work has included researching dynamic power management for all classes of IBM servers, including Intel, Power6, Power7, and z196 processors. Ware has an MS in communications theory and computer architecture from North Carolina State University.

Karthick Rajamani is a research staff member and Manager of the Power-Aware Systems Department at the IBM Austin Research Lab. His work has driven dynamic power management architectures, including designs of the Power6 and Power7 processors and systems for EnergyScale research in energy-efficient memory subsystems and energy-aware systems software. Rajamani has a PhD in electrical and computer engineering from Rice University.

Bishop Brock is a senior engineer in the IBM Systems and Technology Group. His work on the Power7 project has included the development of hardware verification, microarchitectural validation, and power/performance prediction methodologies for power management hardware and firmware systems. Brock has an MS in computer science from the University of Texas at Austin.

Charles Lefurgy is a research staff member at the IBM Austin Research Lab. His work has included energy management features found in IBM System p and x servers. Lefurgy has a PhD in computer science and engineering from the University of Michigan. He's a member of IEEE and the ACM.

Alan J. Drake is a member of the exploratory VLSI group at the IBM Austin Research Lab. His work has included designing critical path monitors and other integrated sensors used for noise detection in high-performance microprocessors. Drake has a PhD in electrical engineering from the University of Michigan.

Lorena Pesantez is a member of the server hardware performance team in the IBM Systems and Technology Group, where she specializes in Power systems instrumentation. Her work has included the definition, hardware data collection, and tuning of the power proxy for the Power7 chip. Pesantez

has a BS in electrical engineering from the University of Texas at Austin.

Tilman Gloekler is the design lead for the Power7 chip power management implementation in the IBM Systems and Technology Group. His work has included design, verification, and bring-up for signal processing and high-performance microprocessor chips. Gloekler has a PhD in electrical from Aachen University of Technology, Germany.

Jose A. Tierno is a research staff member and manager of the mixed signal design group at IBM Research. His interests include the design of digital replacements for analog circuits, in particular all-digital phase-locked loops, and the design of self-timed digital circuits. Tierno has a PhD in computer science from the California Institute of Technology.

Pradip Bose is a research staff member and manager of the Reliability and Power-Aware Microarchitecture Department at the IBM T.J. Watson Research Center. His work has included pre-silicon modeling and definition of IBM Power-series microarchitectures, beginning with the research precursor of the first RS/6000 product. Bose has a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He's a fellow of IEEE and an advisory board member of *IEEE Micro*.

Alper Buyuktosunoglu is a research staff member in the Reliability and Power-Aware Microarchitecture Department at the IBM T.J. Watson Research Center. His work has included the design of IBM p-Series and z-Series microprocessors in the areas of power-aware computer architectures, dynamic power management, and high-level power-performance modeling. Buyuktosunoglu has a PhD in electrical and computer engineering from the University of Rochester. He's a senior member of IEEE and an editorial board member of *IEEE Micro*.

Direct questions and comments about this article to Michael Floyd, IBM Bldg 045, 11400 Burnet Rd Austin, TX 78758; mfloyd@us.ibm.com.