Architecting for Power Management: The IBM® POWER7™ Approach

Malcolm Ware*, Karthick Rajamani*, Michael Floyd[§], Bishop Brock[§],
Juan C Rubio*, Freeman Rawson*, John B Carter*
[§]*IBM Systems and Technology Group, *IBM Research Austin*
{mware,karthick,mfloyd,bcbrock,rubioj,frawson,retrac}@us.ibm.com

Abstract

The POWER7 processor is the newest member of the IBM POWER® family of server processors. With greater than 4X the peak performance and the same power budget as the previous generation POWER6®, POWER7 will deliver impressive energy-efficiency boosts. The improved peak energy-efficiency is accompanied by a wide array of new features in the processor and system designs that advance IBM's EnergyScale™ dynamic power management methodology. This paper provides an overview of these new features, which include better sensing, more advanced power controls, improved scalability for power management, and features to address the diverse needs of the full range of POWER servers from blades to supercomputers. We also highlight three challenges that need attention from a range of systems design and research teams: (i) power management in highly virtualized environments, (ii) power (in)efficiency of systems software and applications, and (iii) memory power costs, especially for servers with large memory footprints.

1. Introduction

The POWER7 processor is the next generation server processor in the IBM POWER family. Each POWER7 chip is 567mm^2, contains 1.2 billion transistors, and is built in IBM's 45 nm (12s CMOS) Cu SOI technology. It is designed to offer better performance, more opportunities for parallelism, and higher levels of power efficiency than its predecessors. It is an 8-core design with each core having up to 4 SMT threads, where a core can run in single-threaded, SMT2 or full SMT4 mode. Cores are out-of-order, allowing them to maximize instruction-level parallelism.

Each core region, known as a "chiplet," contains a 32KB 4-way set associative L1 I-cache and a 32KB 8-way set associative L1 D-cache, a private per-core 256KB L2 cache and a 4MB portion of the shared 32MB L3 cache. The L2 is fully inclusive of both the local D/I L1 caches. The L3 cache exploits embedded DRAM technology [8] to maximize area and power efficiency. The clock frequency of each core chiplet may be independently (and asynchronously to the fabric) controlled via an innovative new digital PLL circuit.

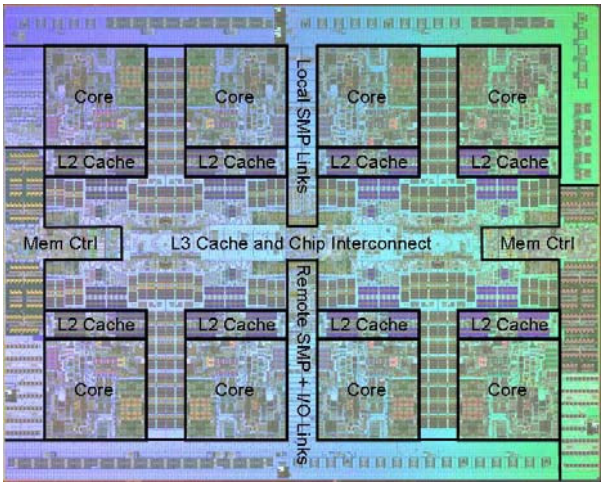

Figure 1: The IBM POWER7 Processor

Like POWER6, POWER7 integrates memory controllers on chip. It has full support for cache coherence for large SMP configurations and is designed to be used in a wide range of servers, ranging from blades to very large commercial configurations and supercomputers.

Starting with POWER6 [1], IBM's POWER family machines offer an array of power management capabilities collectively known as EnergyScale [2]. These capabilities include collection of power and performance measurements and a selection of power management modes, including: (i) *static power save*,

987-1-4244-5659-8/09/$25.00 ©2009 IEEE

which reduces power at a fixed performance cost; (ii) *dynamic power save*, which constantly adjusts core frequencies to exploit opportunities to save power with a minimal impact on performance; and (iii) a *maximum performance* variant, which exploits available power and thermal headroom to boost performance. EnergyScale also supports power capping, which ensures the safe and reliable operation of the servers within user-set power limits regardless of workload transients.

EnergyScale relies on a combination of features provided by the components of the POWER7-based system. It is primarily controlled by a dedicated microcontroller, the *Thermal and Power Management Device* (TPMD), which operates under the control of the *Flexible Support Processor* (FSP) present on all POWER family servers. The TPMD implements the power management policies for the system. In larger, multi-node POWER7 machines, where each node is processor-memory-power delivery complex in its own right, there is one TPMD for every node in the system. Hard real-time firmware running on the TPMD implements the selected power management policies and collects data for reporting purposes.

The POWER7 offers a variety of new features in support of EnergyScale that represent a dramatic improvement over those found on the POWER6. These new features allow POWER7 machines to offer more energy savings, better energy proportionality, and finer control over the various components of the system. This paper describes the features and their uses.

2. Architected Processor Idle Modes

The POWER architecture specifications [6] supported by POWER7 defines four power-saving modes that can provide a continuum of power savings versus latency and software impact. POWER7 implements two of these modes, *Nap* and *Sleep*. Both Nap and Sleep are hypervisor-privileged modes that maintain few of the architected processor resources; exit from a power saving instruction is similar to a thread-level reset. Power-saving instructions that trigger the entry to these modes also cause dynamic SMT mode switching. The core-level power-saving modes described below are only activated when *every* thread in the core has executed a Nap or Sleep instruction.

2.1. Nap

Nap is a processor low-power state designed for short processor idle periods. Nap in POWER7 is designed to have lower latency than in POWER6, where it was the only idle power mode supported. The Nap state is entered whenever the hypervisor has executed a *power-save* instruction on all threads, with at least one thread executing a Nap instruction. In the Nap state, all of the execution units in a core and the L1 cache are clocked off; however the higher level caches and certain timing facilities remain functional, allowing low-latency workload resumption in the event of timer or external interrupts. Core RAS and configuration registers remain accessible to firmware during Nap.

The Nap state by itself provides modest power reduction over a software idle loop. Further, the hardware supports the option of automatically lowering cache frequencies while in the Nap state. This feature provides a significant reduction in power for napping core chiplets, at the potential expense of increased access latency for shared data requested by a non-napping core from a napping cache (L2/L3). The latency from the presentation of an interrupt to a napping core to the first instruction completion after Nap exit is typically less than 5µs. Instruction execution begins immediately upon wakeup regardless of whether the frequency was dropped while in Nap. If the frequency was dropped for Nap, it is slewed back up to the operating point set by the TPMD firmware while instruction execution resumes at Nap exit.

2.2. Sleep

Sleep is a new architectural feature introduced in POWER7. It is a lower-power, higher-latency standby state intended for cores that the hypervisor/OS predicts will be unused for an extended period of time. The Sleep state is entered when every thread on a core executes a Sleep instruction. Upon entering Sleep, hardware state machines purge all data from the core and caches before completely clocking off the entire chiplet. A small logic macro associated with the core chiplet remains awake to handle external interrupts that wake the core out of sleep.

When all cores in a POWER7 processor enter the Sleep state, the voltage supplied to the core chiplets can be automatically lowered down up to a *retention* level, a non-operational voltage sufficient only to maintain static configuration data in the latches and arrays. This mode provides the lowest standby power for a POWER7 processor. Note that firmware running on the FSP or TPMD can temporarily restore operational clocks and voltages to a Sleeping chiplet at any time for maintenance operations, e.g., to access RAS or configuration registers, without restarting instructions on that core.

The latency for entering Sleep varies based on the system design and workload configuration. Sleep exit latency for a single core is typically less than 1ms, and is dominated by the time required to re-initialize the L3 cache eDRAM. Chip-level Processor Sleep exit is dominated by voltage change latency from the retention voltage, which varies by system depending on the voltage control scheme used in the system design.

Figure 2 shows the relative power reductions for Nap and Sleep as measured on a small, early sample set of POWER7 processors. For these measurements the minimum frequency used, fmin, was about 46% of the maximum frequency used, fmax. Vmax and Vmin refer to the set of voltages corresponding to those frequency levels and Vret to the set of voltages corresponding to the retention level.

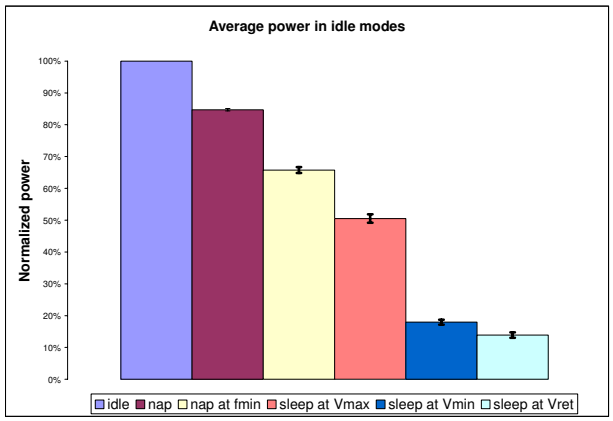

Figure 2: Comparison of processor idle modes

3. Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling has been advocated as an effective tool for processor energy management with many proposals based on its exploitation [7]. The POWER7 processor makes significant advances in processor voltage and frequency controls. POWER7 offers per-core-chiplet frequency scaling, which means that the frequency of each processor core can be adjusted independently. Unlike POWER6, when POWER7 changes the voltage for the core chiplets it can do so without affecting the operating voltage (and therefore frequency) of the SMP interconnect fabric and buses or the connection to the memory controller or IO devices. To co-optimize cache and logic performance and power, processor cores are provided with two independent voltage sources, V_{dd} and V_{cs}, which can be slewed to match frequencies used. The higher V_{cs} voltage allows for reliable, fast operation of the cache array (SRAM and eDRAM) cells, while a slightly lower corresponding V_{dd} voltage significantly reduces leakage for the majority of logic circuits in the core chiplet that perform computation and maintain data coherence.

Table 1: Major POWER7 voltage domains

Rail	Type	Use
V_{dd}	Dynamic	CPU core, cache logic
V_{cs}	Dynamic	Cache arrays, other SRAM
V_{io}	Fixed	Interconnect logic and I/O
V_{mem}	Fixed	Memory controller I/O

3.1. DPLL (Digital Phase-Locked Loop)

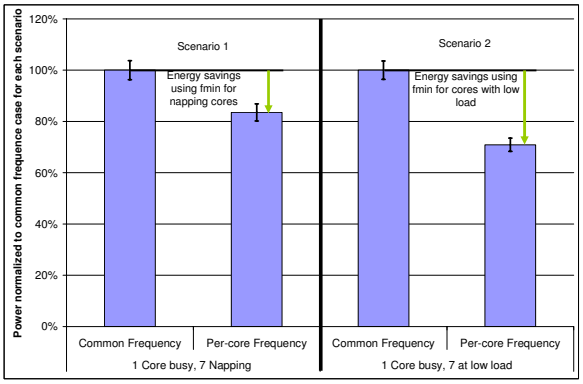

Figure 3: Better energy reduction from DPLL with per-core frequency scaling.

POWER7 for the first time allows cores in a POWER system to run asynchronously with respect to the SMP interconnect fabric using digital PLL technology [9]. EnergyScale firmware can exploit this capability to set an optimal frequency for each core chiplet based on sensors collected during runtime. This is a critical feature in improving the energy-efficiency of the processor when in active use but with different loads on the various cores. Figure 3 shows two scenarios where the per-core frequency scaling capability with DPLL provides increased energy-efficiency. Scenario1 is when a single core is busy in a chip and seven others are napping. Scenario2 is when a single core is busy at 100% load while the other seven are also running but with their workloads' demands met by running at fmin. While both scenarios exhibit power and energy reductions with the per-core scaling, the higher benefits for scenario 2 show the DPLLs unique value for energy-efficiency improvements even when all cores are active.

Analog PLLs traditionally used as clock sources, while providing accurate high frequency clocks, are

limited in their capability to dynamically change frequency output during runtime and are limited by M:N output ratioing compared to an input reference clock. For POWER7 a digital PLL design was employed that offers frequency steps with a resolution of around 1% of the full range. The DPLL slews frequency without pausing instruction execution. It can slew a frequency range of more than 2GHz in under 50µs. Consequently, the POWER7 sees dramatic improvement in both slew rate and frequency range over the POWER6.

3.2. PVIC and SVIC

Voltage regulation in today's systems is done by an external device known as a Voltage Regulation Module (VRM). The VRM receives DC power at a higher distribution voltage (say 12V) and lowers it to a voltage level usable by the microprocessor. The VRM also has a feedback loop to ensure that its output stays within the specified voltage as the microprocessor's load changes. A separate VRM is employed for each voltage domain for the POEWR7.

To exploit fine-grained opportunities for power management, POWER7 has on-chip circuitry to manage the VRMs directly instead of relying on an external controller. Instantaneous change from low to high voltage operation is not possible, especially at the current loads required by today's high-performance processors. Therefore, DVS (Dynamic Voltage Scaling/Slewing) is supported directly by the POWER7 hardware, which transitions to the target voltage following a programmable slope (step size and delay) to ensure clean voltage transitions.

Two communication channels are provided by the POWER7 chip to support requirements of different systems. The Parallel Voltage Interface Control (PVIC) outputs a parallel 8-bit value that encodes the voltage for systems whose controls connect directly to the VRM. The VRM continually monitors this interface, so that incremental voltage changes can be done within microseconds and slewing across the full-range from maximum voltage down to Vretention can be done under 1ms. Each of the V_{dd} and V_{cs} domains has its own PVIC interface allowing each domain to be independently tuned for maximum efficiency.

For our higher-end systems utilizing multiple, redundant VRM masters for additional RAS, dual Serial Voltage Interface Control (SVIC) paths have been introduced in the POWER7. Each SVIC uses an independent I2C serial bus to send an 8- or 16-bit encoded voltage to a redundant master. To protect against noise in the line, a CRC checksum byte is appended to each packet. A hardware voltage sequencer on the POWER7 processor guarantees that the redundant VRM masters for a given voltage domain stay in sync. Additional coordination for the redundant voltage regulation controls causes voltage changes to take on the order of 5 to 10 ms to traverse the full range. The slewing capability in the presence of redundant VRM masters is a first in the industry to our knowledge and meets both high RAS requirements as well as aggressive power management goals.

Figure 4 shows the power reduction characteristics of DVFS on a small sample set of early POWER7 processors running a cache-resident DAXPY loop on each of the 32 SMT threads on each processor. The expected sharper power reductions at higher performance points and the shallower reductions at lower levels depict the well-tuned V-F characteristics for the processor and the benefit of having independent dynamically scalable voltage rails for the logic and array structures on chip.

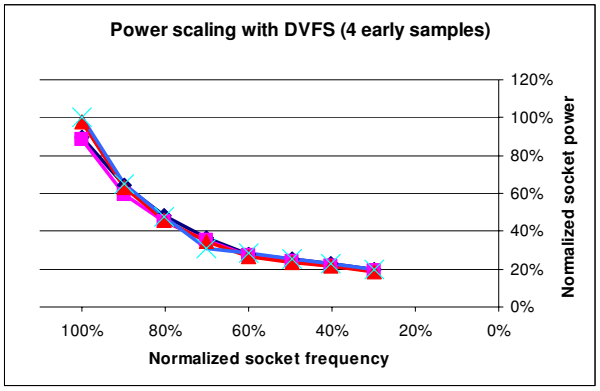

Figure 4: Power reduction with DVFS (Vmin was reached at roughly 60% of fmax for these measurements)

3.3. On-chip power management control units

Control over dynamic and autonomous power management is distributed in the POWER7 design. A local core chiplet control macro (OHA) manages and controls the per-core DPLL, per-core autonomous frequency control loops, the chiplet power proxy described later, and the sequencing of the entry into and exit from architected idle modes. This localized control and sense implementation provides support for independent power management at the granularity of a single core.

Each POWER7 processor also supports a centralized Power Management Controller (PMC). The PMC is responsible for chip-level power management functions including voltage sequencing and chip-level

Sleep. For example, the PMC collects the idle states of all cores, and if it finds that every core is in the Sleep state it will initiate the reduction of core voltage to the retention level.

The PMC also coordinates the actions of the external TPMD controller with hypervisor-directed idle state requests, and serves as an on-chip proxy for the TPMD. Since the TPMD has no control over idle state changes and sleeping cores are not programmable, the TPMD directs its frequency and voltage change requests to the PMC. If a core is in Sleep state when a frequency change request arrives, the request is held by the PMC and forwarded to the core OHA when it reawakens.

4. Memory Power Control

The memory sub-system takes up an increasing fraction of server power in balanced server designs as next-generation processor designs opt for increasing the number of cores within their allotted power budget as an alternative to boosting frequencies. For large-scale consolidation platforms like the high-end IBM POWER servers, memory system power can potentially exceed that of the processors.

The POWER7 memory system uses DDR3 memory with the controller adding support for *self-refresh* idle modes while continuing support for the *power-down* idle modes first introduced in POWER6 for DDR2. Dynamic memory request throttling is also available to cap memory power consumption. Along with request throttling, programmable power control state machines in the memory controller can also limit the number of ranks powered on at any given time providing additional dynamic power-performance trade-off capabilities for the memory sub-system. Finally multiple boot-time selectable memory interleaving schemes are supported to enable a wider choice of power-sensitive performance tuning options for the memory sub-system. For example one interleave scheme may favor finer granularity power control for lower average power in exchange for lower peak single stream performance, while a second scheme may offer maximum single stream performance at the cost of larger power control granularity and higher average power.

5. Sensors

EnergyScale dynamic power management uses real-time thermal, power and performance feedback to realize different power management policies.

5.1. Temperature sensors

POWER7 adopts a Digital Thermal Sensor (DTS) design, a temperature sensing circuit that utilizes a bandgap diode voltage comparator. The voltage reading is converted by the logic on the chip to a temperature in degrees Celsius via a polynomial curve fit ($px^2 + mx + b$). The coefficients are derived during manufacturing test and calibrated per DTS using a traditional off-chip thermal diode as a reference. Circuitry in the chip automatically detects and notifies the service element if a thermal "trip" occurs when a DTS temperature exceeds a predefined threshold. The DTSs are placed near areas of high power density and are spaced to allow both a thermal map and localized hotspots of the chip to be obtained in real-time. These hotspots change based on both workload type and multi-core utilization patterns and can be measured and managed by the EnergyScale system. Ambient temperature sensors, thermal sensors on the Memory Buffer Chip and DIMMs, and VRM thermal-trip circuits help complete the real-time view of the thermal conditions visible to the system-level power and cooling management solutions.

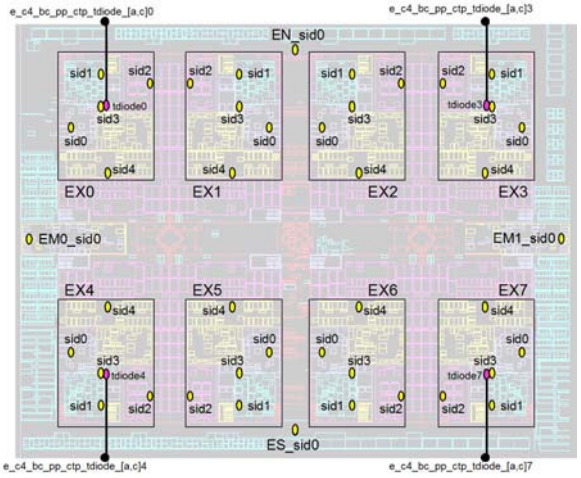

Figure 5: Digital thermal sensors (DTS) marked on the POWER7 chip, along with the 4 thermal diodes used for calibration – the sid*n*s mark the DTS locations. The EX*n*s mark the eight chiplet regions.

5.2. Power sensors

Power measurements in POWER7 systems are a third-generation IBM design. The TPMD has fourteen A/D channel inputs, each sampling at a minimum of 1kHz with 10 bits of conversion accuracy. Each channel has appropriate anti-aliasing filtering to improve measurement accuracy even under transient

conditions. This aids the quality of the power capping functions implemented by the firmware running on the TPMD. Periodic calibration on each channel ensures precision measurement even in the presence of significant thermal swings.

The channels are used to obtain precision measurements for the voltage and current senses of the bulk power supplies (AC/DC convertors), and separately the power consumption of the POWER7 processors, associated memory buffer chips, memory DIMMs, fans, and I/O sub-system components.

The measurement circuitry and calibration process ensure that the bulk power measurements are accurate to within 2% and subsystem measurements to within 3%. The TPMD can read the values every millisecond. By contrast, solutions that obtain current readings from VRM devices are typically 16X slower (16msec) and 2X less accurate (7-8% error). All power is measured at a common 12V level to include VRM efficiency losses, board distribution losses, and actual device power consumption within the corresponding voltage domain. All measurements are maintained in the firmware at 100mW resolution and are available as sensors at granularities ranging from 'raw' 1ms samples or as aggregated sensors for intervals up to 8 seconds.

It is difficult to isolate the power consumption of each processing element attached to a common power plane or to measure power consumption at the core level using only chip-level current measurements. Several researchers have described schemes for using hardware performance counters collected by software to estimate core or chip power consumption [3,4]. POWER7 implements such a scheme largely in hardware in the form of *power proxies*.

To compute a power proxy value, more than 50 different architectural events are programmably weighted and combined in hardware into a single number, on time scales as small as 32 µs. This number represents active core power related to instruction activity, plus clock-tree power dependent only on the current frequency. Power management software then adjusts the hardware proxy for the effects of leakage, temperature, and voltage. Events counted include instruction dispatches, register file and cache accesses, and execution unit activity. The weights for each event are derived empirically by measuring core-level power against the uncombined event counts for a variety of workloads. The weighting architecture anticipates a least-squares type curve fit during calibration and includes two levels of weighting and support for the negative coefficients required for optimally fitting systems of dependent variables. Early experiments suggest that power proxy estimates are very close to actual power consumption in most cases, although the accuracy of any such estimate varies with the workload. For example, we have seen POWER7 core power vary by as much as 6% for the same application depending on the data being manipulated by the core, a property very difficult to capture by any rate-based proxy.

The POWER7 memory controllers also support weight-programmable, event-based power proxies for the memory sub-system. With proper calibration these are designed to drive local power control state machines in the memory controller and augment traditional performance-centric memory access scheduling with power awareness.

5.3. Processor and memory activity counters

Dedicated processor activity sensors provide the power management firmware core-level resource usage information that can be used by the power management algorithms. The TPMD firmware uses this sensor data to manage the aggregate power-performance demand of each core. The demand may come from a single operating system or from multiple operating system images sharing the processor. Events tracked by the counters include cycle counts, instruction rates, and memory hierarchy event counts, rates, or stalls. The programmable nature of these counters allows them to be tuned to the wide-array of memory hierarchy characteristics available across the family of POWER7 systems. In addition to core-level activity, the memory controllers, which are system-wide shared resources, support real-time memory event counters dedicated to power management to track both demand and power-mode usage impact at multiple granularities within the memory sub-system. These counters are intended to be used to implement various performance-aware EnergyScale power management policies.

6. Communication infrastructure for Power Management

Access to the diverse array of POWER7 sensors and controls is available to the power management logic via a special on-chip network.

6.1. Pervasive Infrastructure

The power management functions of POWER7 are part of the *pervasive infrastructure*, an on-chip communication and maintenance network that provides access to control and status registers during runtime. The pervasive infrastructure is accessible to the FSP

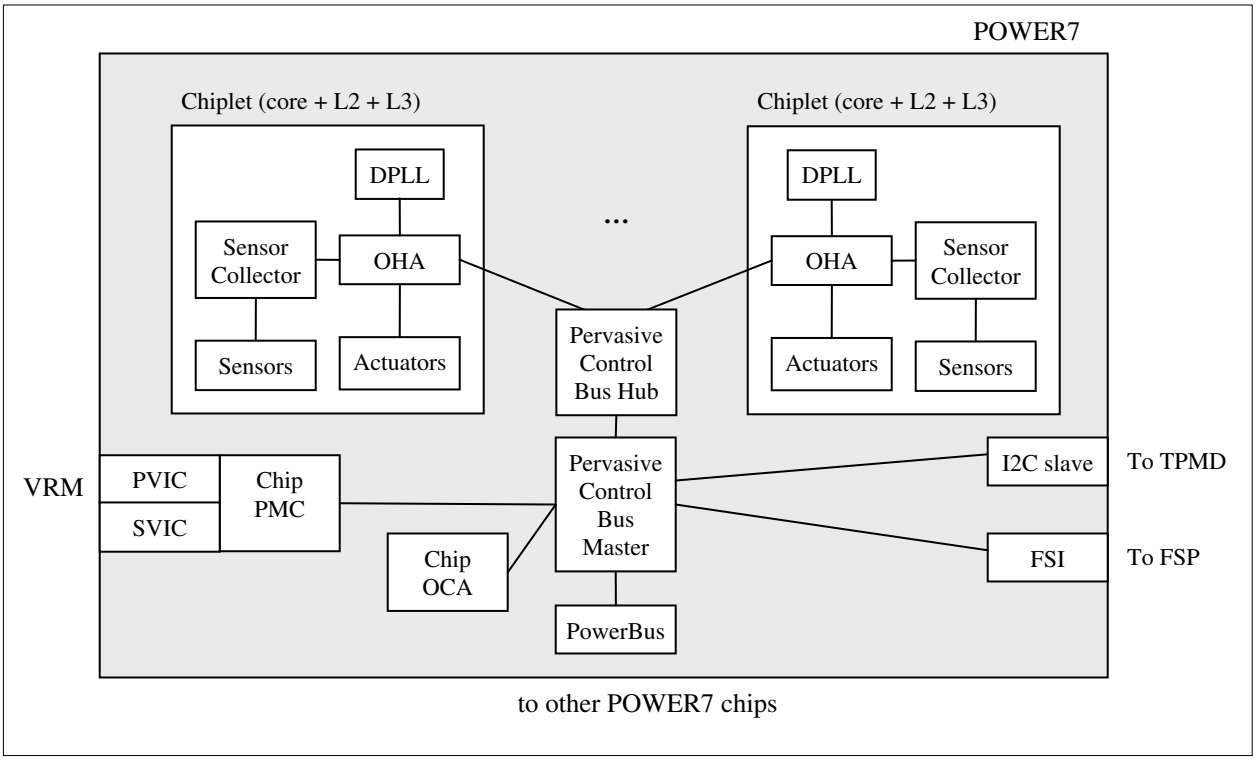

Figure 6: Key components of the POWER7 power management infrastructure

and TPMD via external ports and to the hypervisor via memory-mapped I/O operations. Pervasive infrastructure operations can also be carried across the SMP interconnection fabric in multi-node systems.

6.2. I2C Slave and OCA

As with POWER6, power management policies can be implemented completely out-of-band by the TPMD communicating with the POWER7 over an I2C link. The I2C Slave on the POWER7 provides the TPMD access to the POWER7's pervasive infrastructure, allowing it to read and write select, on-chip registers.

In POWER7 this access is aided by an *On-Chip Communications Assist* (OCA). The OCA collects power, performance, and temperature information from throughout the POWER7 using the pervasive infrastructure, and stores it into a central location on a precise, programmable schedule. Up to 1024 bytes of sensor data can then be streamed out to the TPMD in a single high-level I2C access. Given the larger number of cores per chip and the wider range of sensors on the POWER7, the OCA is critical in reducing the overhead for sensor data acquisition and improving the scalability of the power management firmware.

7. Experimental Control Loops and Capabilities

POWER7 implements some experimental autonomous frequency control loops in hardware, taking advantage of the low-latency, fine-grained frequency control provided by the DPLL clocking scheme. The controllers described below rely on the fact that two frequencies can be communicated to the frequency arbiter inside each core chiplet (in the OHA) – the nominal frequency for the current voltage, as well as a minimum frequency for power reduction. The local hardware control loops can move frequencies between these two limits. These capabilities are currently being evaluated for efficacy and suitability for future product-level use, with the goals of greater power efficiency and increased performance in power-constrained environments.

7.1. Low-activity Detect (LAD)

The POWER7 low-activity detect (LAD) mechanism implements a hardware frequency-control loop that measures active cycles/instruction throughput over a programmable interval, as short as a few µs,

lowers the frequency temporarily if the count falls below a threshold, and scales it back up once the interval count exceeds the threshold. This mechanism permits very rapid frequency scaling to exploit low compute phases that are too short for firmware or system-software level to detect. While capable of providing power reduction on its own, it can also supplement more sophisticated but longer time-scale software control loops by providing implicit feedback to them of opportunities to use lower frequencies by changing frequencies whenever the thresholds are crossed. The software control loops can then lower the maximum frequency and associated voltages suitably, having observed that the effective average frequency is lower as a result of LAD actions.

7.2. Soft capping

In a power-constrained system environment, it may be necessary to reduce core frequency to meet socket- and system-level power caps. Given that the power proxy discussed earlier provides a per-core power consumption estimate, POWER7 implements an autonomous hardware mechanism that controls core frequency in direct response to the current estimated power consumption. This hardware capping support extends power capping capabilities to a finer granularity (chiplet-level) and smaller timescales than possible with just larger domain physical measurements and software control loops. The power cap implemented by this mechanism is soft in the sense that response time, estimation error, and workload variation render the power cap a statistical property rather than an absolute, physical limit.

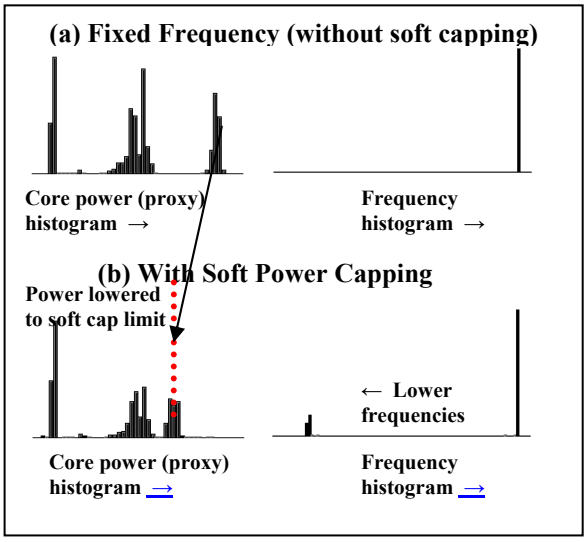

Figure 7: Soft capping – Histograms of power proxy and frequency for a specific workload

The soft capping mechanism consists of upper and lower power proxy thresholds and a mechanism that reduces (increases) frequency by programmable steps if estimated power exceeds (falls below) the upper (lower) threshold. Soft capping operates on timescales as small as 32 µs. Varying the threshold location, threshold separation and frequency response could allow several different management schemes, including centering maximum estimated power consumption within a range, or managing the tail of the estimated power distribution above the upper threshold.

Figure 7 shows the soft capping logic in action. It presents distributions of the power proxy estimates of core power consumption over time for a particular workload, along with corresponding distributions of frequency, under two scenarios. Figure 7(a) presents the distributions when the core is run at a fixed frequency, while Figure 7(b) presents the distributions when soft capping has been enabled with the indicated target power cap. The introduction of a soft power cap reduces core frequency for the phases of the workload with power estimates that exceed the soft cap. As can be seen in Figure 7(b) soft capping reduces the estimated maximum power, which is now centered on the power cap target, without significantly impacting the lower-powered workload phases.

7.3. Critical Path Monitoring

Voltage and frequency operating points are typically characterized based on worst-case assumptions and include worst-case margins. This may lead to less than optimal power and performance characteristics under typical scenarios. The *critical path monitors* (CPM) introduced in POWER6 [5] continue to evolve and exist throughout the POWER7 design to provide insight into actual required voltage margins. The CPMs synthesize critical path timing under current operating conditions, and output an indication of available timing slack. The TPMD can use this information to more effectively implement power and performance efficiency policies.

8. Power Management Firmware

The EnergyScale functions for POWER7-based machines run primarily in the firmware executing on the FSP and the TPMD. The FSP provides the primary control and communications pathways between the TPMD and higher-level systems management software. The TPMD executes power and performance management algorithms using hard real-time

measurements of power, temperatures, and performance.

Like on POWER6, the TPMD communicates with the POWER7 processor over I2C, but at a faster rate. The TPMD manages an I2C connection to each processor chip and accesses the sensors and actuators inside the chip through the pervasive infrastructure. POWER7 benefits from the presence of the OCA, which allows the transfer of more data in a single operation and dramatically increases the effective bandwidth the TPMD has to chip-level information.

With the reduction of Nap latencies and the addition of Sleep, there are increased opportunities for both the hypervisor and the operating systems to reduce power consumption during periods of reduced load. Which state the software chooses depends on for how long it expects the processor to be idle. Nap, with its lower entry and exit latencies, is designed to support short-term idle states that occur as the result of ordinary OS and hypervisor scheduling. Sleep provides superior power savings when the system software has specific knowledge to conclude that the idle period will be of relatively longer duration.

Operating systems for POWER architecture machines offer accurate accounting using a per-SMT-thread processor register to track how much execution time each software thread receives. On the POWER6, this register was supplemented with the "SPURR" register, which the firmware uses to scale the usage counts by the current frequency to ensure that the time reported better reflects the amount of computation that the machine can perform in the interval the thread executes. The POWER7 maintains the SPURR register, but adds additional capability for auto-adjusting the scale-factor by the exact core frequency for some select usage scenarios.

The POWER6 design contains a single 64-bit register for direct communication between the hypervisor and the TPMD. The POWER7 adds three 64-bit registers that can be read and written by both the hypervisor and the TPMD. These registers serve as a quick communication path between the hypervisor and the TPMD for cooperative power-management functions for both multi-node power management (with a TPMD per node) as well as exposing power management information from the TPMD to higher level software via the hypervisor and operating systems.

9. Challenges Ahead

This section briefly describes three areas that we believe demand significant research attention from a power management perspective.

9.1. Virtualization-driven consolidation and dynamic power management

A key feature of POWER family systems is that they are easy to virtualize and support multiple logical partitions, each with its own operating system image and isolated from all of the others. Virtualization allows server consolidation, which reduces their cost and total energy consumption, and often reduces management overheads for larger installations. Consequently, its adoption is increasing across all server system architectures.

Power management systems are increasing in sophistication in an attempt to satisfy the real-time performance needs and power/cooling constraints of modern server environments. The introduction of virtualization introduces significant new challenges to achieving these diverse objectives.

In virtualized systems, each partition can have a unique workload with resource needs quite different from those of other partitions sharing the same power and thermal environment. The POWER7 processor makes some inroads into this problem by enabling core-level power-performance adaptation and real-time, core-level, diverse sensor feedback. However, there is an inherent tension between fine-grain physical resource sharing and system-level energy optimization on the one hand and the need to isolate partitions with diverse workload needs on the other. A tightly coupled, cross-stack integrated power management solution, where applications and/or OSes specify requirements and the hypervisor and TPMD coordinate partition scheduling with hardware state/activity controls to achieve these requirements in the most energy-efficient manner, is conceptually feasible. However, it is unclear precisely what architecture elements are needed to support this design, and how such an integrated solution should balance the trade-offs between design complexity, satisfying diverse performance goals, and the desire to minimize energy usage in this context.

9.2. Resource utilization and management by application and system software

The power-oblivious way in which modern applications consume resources is an imposing challenge for even the best power management system. Software typically manages resources to maximize performance with no consideration for energy usage, e.g., employing spin locks in the (often vain) hope of minimizing stall time. This "greedy" approach to resource usage can mask the fact that certain resources

(e.g., the cpu or memory) are underutilized, thereby interfering with a power management system that wishes to place underutilized resources in to low power states (e.g., reducing core frequency).

Certain programming styles also work against effective power management. For example, the "lock step" approach used with many scientific computing codes has programmers (or compilers) carefully balancing the execution time of each thread fearing load imbalance impact on performance. Such codes can become "brittle" when run on autonomic systems that dynamically adjust each processor's frequency independently to achieve specified energy efficiency goals. As a result, it is not uncommon for users running such codes to disable power management governors to avoid perturbing fixed execution rate assumptions, thereby eliminating any opportunity to improve energy efficiency during periods of low or imbalanced load.

To address this problem, programmers, applications, middleware, and/or system software should become more energy-aware. Just as we currently optimize for performance, considering algorithmic efficiency and tuning to specific architectural constructs such as cache size, programmers should also address energy efficiency. Two major research challenges in this space are (i) developing models that make it simple for programmers to reason about energy efficiency and (ii) revamping the resource allocation and scheduling policies used by middleware and system software to exploit, rather than defeat, energy management mechanisms.

9.3. Memory power in future balanced system designs

The multi-core approach to increased socket-level performance places higher demands on both off-chip memory capacity and bandwidth than the previous frequency-driven approach to socket-level performance growth. Given the multi-core approach is the only feasible alternative from a power perspective for processor performance growth, memory sub-system energy efficiency is now much more important.

Figure 8 shows how a potential next-generation high-end POWER7 server has to grow its memory power budget to meet the expected system-level application performance growth suggested by the socket-level core growth (4X) compared to a POWER6 design. While the power demands grow, the supply is limited by the power distribution subsystem within the machine and the (un)willingness of datacenter operators to provision more power.

It is clear that DRAM energy efficiency improvements cannot alone keep up with the memory power needed due to the increased performance and capacity demands from growing number of cores/socket. The memory power and performance requirements of future servers require re-thinking, re-architecting, and new power-aware approaches to memory hierarchy design.

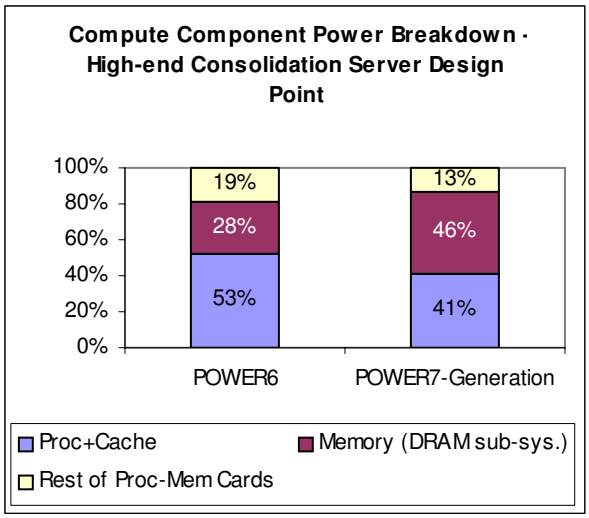

Figure 8: Growth in memory power – A high-end IBM server example

10. Conclusions

The POWER7's power management capabilities result from experience gained with product implementations and research on the POWER6 processor. The facilities of the POWER7 extend the ability of the firmware and the software to control the power state of the processor and the machine dramatically. Key enhancements include the introduction of a second, deeper low-power mode and the ability to scale the operating frequency on a per-core basis rather than restricting the machine to a single, global frequency.

The POWER7 extends the range of frequency, and, thus, voltage scaling dramatically, allowing for much greater energy savings on workloads that are active in bursts or bound to memory or I/O. Key to the power management for POWER7-based systems is the instrumentation used to determine processor and memory activity. With the POWER7, there are improvements in both the data collection and communication through the introduction of the OCA. Mechanisms such as automated frequency and voltage slewing, conversion to degrees Celsius, and power proxy calculations built into the hardware offload

many functions from the power management firmware. Multiple voltage interfaces to meet distinct needs of lower-end and high-end systems, significant enhancements to memory power management, and scaling the sensor collection infrastructure all contribute to making the processor suitable for a variety of dynamic power management needs from those of blade systems to large-scale consolidation platforms and supercomputers. The design of the POWER7 also anticipates some power management firmware and software evolution with privileged communication registers between TPMD and the hypervisor for cooperative management.

The POWER7 provides IBM a strong basis for highly energy-efficient high-performance server systems. The processor also facilitates novel schemes in firmware and software to control both the power and the performance of machine in a more precise manner than was previously possible.

Acknowledgment

The authors of this paper would like to thank the IBM Research and Design teams who made POWER7 power management possible. Pradip Bose, Alper Buyuktosunoglu and Lorena Pesantez, made several contributions in the area of power proxy and micro-architectural techniques for core power management. Jose Tierno drove the very successful DPLL design. Todd Rosedahl, as Chief Engineer for EnergyScale Firmware development, was instrumental in interlocking design requirements and capabilities in the context of the different system designs. Josh Friedrich and Victor Zyuban led the efforts to manage the TDP of the chip and supported our adaptive techniques. Also a special appreciation to the design team responsible for implementation of many elements of this architecture, specifically to Tilman Gloekler, Birgit Schubert, Bruno Spruth, Cedric Litchenau, and Thomas Pflueger.

This material is based upon work supported by the Defense Advanced Research Projects Agency under its Agreement No. HR0011-07-9-0002.

References

[1] M. S. Floyd, S. Ghiasi, T. W. Keller, K. Rajamani, F. L. Rawson III, J. C. Rubio, M. S. Ware, "System power management support in the IBM POWER6 microprocessor", *In IBM Journal of Research and Development* 51(6): 733-746 (2007).

[2] H.-Y. McCreary, M A. Broyles, M. S. Floyd, A. J. Geissler, S. P. Hartman, F. L. Rawson III, T. J. Rosedahl, J. C. Rubio, M. S. Ware, "EnergyScale for IBM POWER6 microprocessor-based systems", *In IBM Journal of Research and Development* 51(6): 775-786 (2007).

[3] G. Contreras and M. Martonosi, "Power prediction for Intel XScale® processors using performance monitoring unit events", *In Proceedings of the 2005 international Symposium on Low Power Electronics and Design,* San Diego, CA, pp. 221-226.

[4] F. Bellosa, "The benefits of event-driven energy accounting in power-sensitive systems", *In Proceedings of the 9th Workshop on ACM SIGOPS European Workshop*, September, 2000.

[5] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, V. Pokala, "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor", *In proceedings of the IEEE International Solid-State Circuits Conference, 2007,* February 2007.

[6] Power ISA™ Version 2.06. Available at *http://www.power.org/resources/downloads/PowerISA_V2.06_PUBLIC.pdf*

[7] K. Rajamani, C. Lefurgy, J. Rubio, S. Ghiasi, H. Hanson and T. Keller, "Power management for computer systems and data centers", *Tutorial presented at the 2008 International Symposium on Low Power Electronics and Design*, August, 2008.

[8] J. Barth, et al.,"A 500MHz Random Cycle 1.5ns-Latency, SOI Embedded DRAM Macro Featuring a 3T Micro Sense Amplifier", *in the 2007 IEEE International Solid-state Circuits Conference Digest of Technical Papers,* February, 2007.

[9] A.V. Rylyakov, J. A. Tierno, G. J. English, D. Friedman, M. Megheli, "A Wide Power-Supply Range (0.5V-to-1.3V) Wide Tuning Range (500 MHz-to-8 GHz) All-Static CMOS AD PLL in 65nm SOI", *in the 2007 IEEE International Solid-state Circuits Conference Digest of Technical Papers,* February, 2007.