

Seminararbeit: Process Cruise Control und der Einfluss von Event-Driven Clock Scaling

Maximilian Meidenbauer

Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

Die Anforderungen an die Energieautarkie von Eingebetteten Systemen steigen immer weiter, was zur Folge hat, dass das Energiemanagement immer mehr in den Fokus rückt. Um der steigenden Herausforderung eines effektiven Managements der Energie zu begegnen, kann dynamisches Leistungsmanagement eingesetzt werden. Als effiziente und kostengünstige Methode hat sich dynamic voltage und frequency scaling bewährt. Sollte eine Technik zur inter-task "Dynamische Spannungs- und Frequenzskalierung (engl. dynamic voltage and frequency scaling, DVFS)" gebraucht werden, bietet sich die Methode Process Cruise Control von den Autoren Weissel und Bellosa an. Sie stellen eine Technik vor, bei der die Event-Counter eines Systems dazu genutzt werden können energietechnische Informationen auszulesen. Dies kann dann dazu genutzt werden um die Taktfrequenz des Prozessors während der Laufzeit anzupassen um Energie zu sparen. Das Papier wurde vor 20 Jahren veröffentlicht und wird seitdem oft zitiert. Die Nutzung der Event-Counter als Anhaltspunkt für den Energieverbrauch war ein Novum, was bis heute in vielerlei Systemen eingesetzt werden kann. Die vorgestellte Technik konnte sich als Paradebeispiel für inter-task DVFS etablieren und ist in der Lage bis zu 22% Energie einzusparen, bei einem Verlust der Performance von nur mehr 10%.

1 EINLEITUNG

Die Weiterentwicklung von Computern hat es Herstellern ermöglicht, die Größe von Computerchips und Boards immer weiter zu verringern. Dies führte letztendlich zu tragbaren Versionen von z.B. Telefonen oder Radios. Um die Lebens- und Nutzungszeit dieser zu erhöhen rückt das Energiemanagement eingebetteter Systeme vermehrt in den Fokus. Ein Lösungsansatz für effektives Energiehandling ist dynamisches Leistungsmanagement (engl. dynamic power management). Dieses hat zum Ziel den Energieverbrauch eines Systems möglichst gering zu halten, indem die Betriebszustände des Systems während der Laufzeit dynamisch angepasst werden. Zum Beispiel können Ressourcenabruf und -verteilung in verschiedenen Szenarien angeordnet werden, um einen optimalen Energieverbrauch zu erzielen. Eine oft benutzte Technik zur Ressourcenverteilung ist Dynamische Spannungs- und Frequenzskalierung bei der die (Versorgungs-)Spannung oder die Taktfrequenz (oder beides gleichzeitig) des Prozessors während der Laufzeit angepasst werden.

2 DYNAMISCHE SPANNUNGS- UND FREQUENZSKALIERUNG

Dynamische Spannungs- und Frequenzskalierung ist eine weitverbreitete Technik und wird auf vielerlei Systemen angewendet.[16] Dies reicht von kleinen eingebetteten Systemen, über Desktop- bis hin zu ganzen Serversystemen. Die Technik konzentriert sich dabei

besonders darauf die Versorgungsspannung und/oder die Taktfrequenz des Prozessors verändern. Dies passiert entweder dynamisch während der Ausführung von Tasks oder zwischen den auszuführenden Tasks. Dabei ist die Technik in der Lage den Stromverbrauch einer CMOS-Schaltung nach folgender Formel zu verändern:

$$P = C f V^2 + P_{static} \quad (1)$$

Dabei bezeichnet V die Versorgungsspannung, f die Frequenz der Transistorgates und C die aggregierte Kapazität, die den durchschnittlichen Energieverbrauch pro Schaltvorgang repräsentiert. Die für einen stabilen Betrieb nötige Versorgungsspannung wird bestimmt durch die Frequenz, bei der die Schaltung eingestellt ist. Wird diese verringert, ist es möglich die Versorgungsspannung und damit den Energieverbrauch zu senken. Dies ist besonders bei der Versorgungsspannung lohnenswert, da diese in Formel (1) quadratisch eingeht. Nachteilig wirkt sich aus, dass die Performanz der Schaltung dadurch sinkt, da die gestellten Aufgaben langsamer ausgeführt werden. Dies macht es nötig eine Balance zwischen dem Herunterregeln von Frequenz bzw. Spannung und der vom System geforderten Leistungsfähigkeit zu finden, welches häufig anhand einer festgesetzten Heuristik geschieht. Diese wird abhängig von den an das System gestellten Ansprüchen ausgewählt.

3 PROCESS CRUISE CONTROL

Eine Heuristik um dynamisches Leistungsmanagement zu betreiben stellen die Autoren Andreas Weissel und Frank Bellosa in ihrem Papier "Process cruise control: Event-driven clock scaling for f" vor.[24] Als Basis nutzen die Autoren dabei Ereigniszähler oder Event-Counter (engl. event counter). Diese werden normalerweise dazu genutzt um die Performance des Systems einschätzen zu können. Jedoch zeigen die Autoren auf, dass Event-Counter ebenso Informationen über den Energieverbrauch des Systems preisgeben können. Diese Informationen werden dann dazu benutzt um die Taktfrequenz des Prozessors zwischen den Tasks dynamisch anzupassen und so den Energieverbrauch zu senken. Dafür haben die Autoren das Model "Process Cruise Control" entwickelt, welches in der Lage ist, ähnlich eines Autopiloten in einem Fahrzeug, die Taktfrequenz in Echtzeit anzupassen.

3.1 Verwendete Hilfsmittel

Als Hilfsmittel werden hier hauptsächlich bereits vorhandene Systembestandteile genutzt mit Ausnahme einer Messeinheit Intel IQ80310, welche extra für diese Forschungszwecke an das System angebunden ist. Die Aufgabe der Messeinheit besteht hier darin, die durch das Betriebssystem und den Scheduler getroffenen Maßnahmen zu überwachen. Zur Verarbeitung von Tasks wird ein Intel XScale 80200 Prozessor mit Taktfrequenzen von 333 MHz bis 733 MHz

verwendet. Das bedeutet, er ist in der Lage die verarbeitende Taktfrequenz im Bereich 333 MHz und 733 MHz in 66-MHz-Schritten zu ändern.

3.2 Extraktion von Informationen aus Event-Countern

Zunächst identifizieren die Autoren verschiedene grundlegende Tasks, welche immer wieder gebraucht werden und in jeglichen Arten von Code vorkommen können.[24] Dies umfasst zum Beispiel das Aufrufen von Funktionen ("call function"), das Lesen des Caches oder das Schreiben in den Arbeitsspeicher ("r/w memory"). Für diese Auswahl an Funktionen wird der Energieverbrauch auf allen dem Prozessor verfügbaren Taktfrequenzen vermessen wie in Bild 1 zu sehen ist. Aufgetragen ist die benötigte Energie zweier Tasks über die verfügbaren Taktfrequenzen des Prozessors. Dabei zeigt sich, dass der Energieverbrauch bei Beteiligung des Speichers deutlich höher liegt, als beim Aufruf von Funktionen, was alleinig durch die CPU, ohne Beteiligung anderer Bauteile verarbeitet wird.

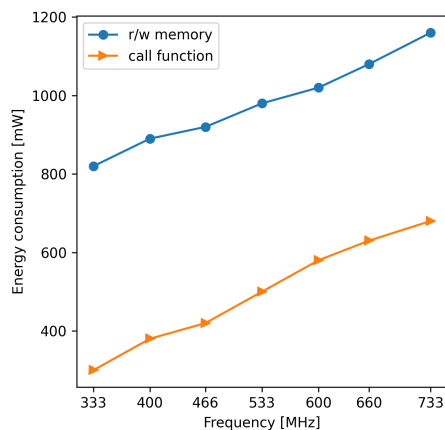


Figure 1: Energieverbrauch zweier Tasks über die Taktfrequenz nach [24]

Eine Veränderung der Taktfrequenz wirkt sich allerdings auch auf die Performance der Tasks und damit der Applikation aus. Da die Performance der Anwendungen durch die Energiesparmaßnahmen nicht einbrechen soll, wird die Performance der Tasks über alle Taktfrequenzen ausgewertet (Bild 2). Der Graph zeigt, dass die Performance für den Aufruf von Funktionen annähernd linear mit der Taktfrequenz ansteigt, während sie für die das Lesen/Schreiben in den Speicher mit Zunahme der Taktfrequenz abflacht. Dies liegt in höheren Wartezeiten der CPU begründet, welche aufgrund des Geschwindigkeitsunterschieds zwischen Speicher und CPU auftreten.

Diese Abweichung im Verhalten der beiden Tasks haben sich die Autoren zunutze gemacht. Bild 3 zeigt das Verhältnis des Energieverbrauchs gegenüber der Performance. Das Verhältnis bleibt weitgehend gleich für den Aufruf von Funktionen, während der Energieverbrauch bei einem Speicheraufruf gegenüber der Performance mit sinkender Taktfrequenz abnimmt.

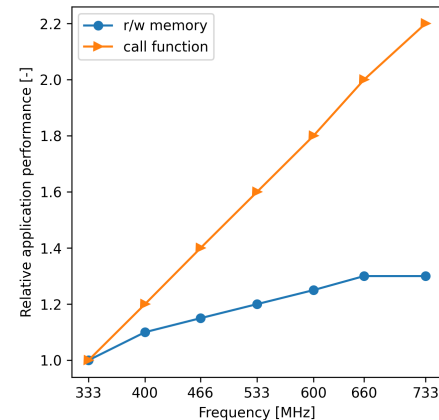


Figure 2: Performance zweier Tasks über die Taktfrequenz nach [24]

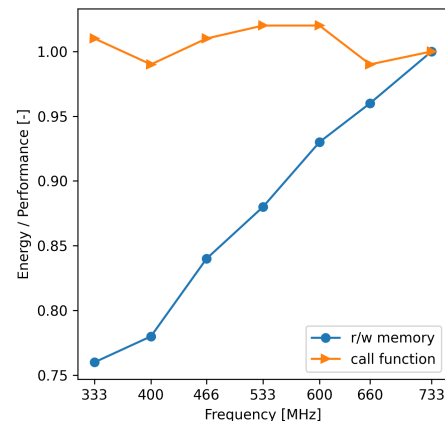


Figure 3: Energie-/Performancerelation zweier Tasks über die Taktfrequenz nach [24]

Daraus lässt sich ableiten, dass hier ein Einsparpotential existiert, sobald in einem Task die Nutzung des Speichers involviert ist. Die Nutzung des Speichers wiederum kann mithilfe von Event-Countern überwacht werden. Aus diesem Sachverhalt haben die Autoren ein Diagramm (siehe Bild 4) entwickelt, welches, abhängig von den durch die Event-Counter gezählten Vorkommnisse eine einzustellende Taktfrequenz vorschlägt. Dabei hat die Benutzung des Speichers entscheidenden Einfluss. Beträgt die Anzahl an Speicheranfragen nur 3% an einem Taktzyklus, wird die Taktfrequenz schon auf die niedrigste Stufe eingestellt. Durch die Beobachtung der Event-Counter ist der Scheduler des Systems dann in der Lage die Taktfrequenz entsprechend zu regeln.

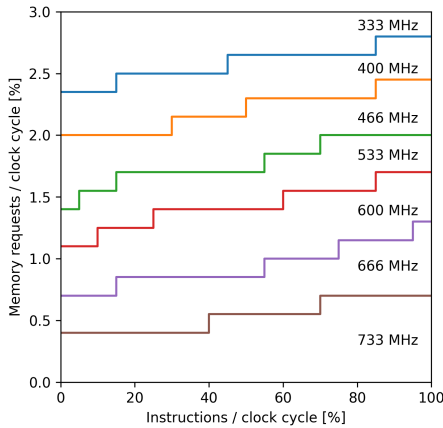


Figure 4: Taktfrequenzdomänen abhängig von der Nutzung des Speichers und Anzahl der Instruktionen eines Taktzyklus nach [24]

3.3 Automatisierung der Taktfrequenzeinstellung

Mit der in Bild 4 vorgestellten Lookup Tabelle haben die Autoren die Voraussetzung geschaffen ein automatisiertes System, die sogenannte "Process Cruise Control" einzusetzen.[24] Dieses ist anhand der vorliegenden Event-Counter in der Lage, wie in Bild 5 dargestellt, die Taktfrequenz des Prozessors zwischen jedem Task immer wieder anzupassen. Das Regelwerk des Modells setzt dabei die von den Event-Countern abhängigen Taktfrequenzen für das System um. Von den Autoren durchgeführte Benchmarks haben gezeigt, dass durch die automatisierte Anpassung der Taktfrequenz bis zu 22% an Energie eingespart werden kann, wenn ein Verlust der Performance von bis zu 10% toleriert wird.

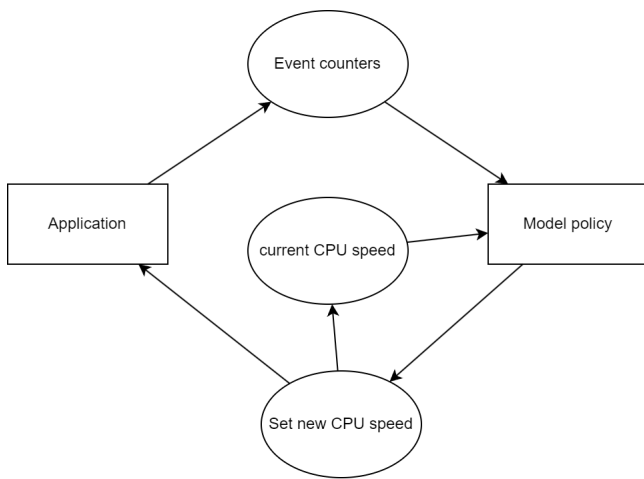


Figure 5: Ablaufdiagramm des Modells nach [24]

4 DISKUSSION DES WISSENSCHAFTLICHEN KONTEXTS

Da das Papier mehr als zwanzig Jahre alt ist, soll in diesem Rahmen die wissenschaftliche Einordnung und dessen Einfluss auf die Fachwelt diskutiert werden. Um das Papier wissenschaftlich einzuordnen wird dabei zuerst betrachtet, welche Neuerungen die Veröffentlichung des Papiers erzielt hat. Zusätzlich werden außerdem verwandte Arbeiten miteinbezogen, um das Papier in den Kontext zeitgenössischer wissenschaftlicher Arbeit zu setzen. Weiterhin wird der Einfluss des Papiers auf darauffolgende Arbeiten dargelegt. Abschließend wird dargestellt welche langfristige Einflüsse heute noch existieren.

4.1 Hervorgebrachte Neuerungen

Die Autoren haben gezeigt, wie die Event-Counter eines Betriebssystems dazu genutzt werden können um Energie während der Ausführung von Tasks einzusparen. Dies ist insofern bedeutsam, da Event-Counter eher aus Performancegründen und nicht aus energietechnischer Sicht ausgewertet werden. Allerdings wird bei dieser Technik die Anpassung der Spannung nicht hauptsächlich untersucht. Da die Spannung quadratisch in den Energieverbrauch einer CMOS-Schaltung eingeht (siehe Formel (1)) hätte es sich gelohnt auch diesen Aspekt weiter zu betrachten. Zudem wird auf Mängel am Betriebssystemdesign bezüglich der vorgestellten Technik hingewiesen. Da die Event-Counter nicht in der Lage sind einzelne Befehle, Abruf aus dem Speicher oder Schreiben in den Speicher zu unterscheiden, sondern nur Instruktionen und Speicherzugriffe zählen, geht ein großer Teil des Energiesparpotentials verloren. Die Technik könnte gegebenenfalls noch weiter ausgebaut werden, wenn die verwendeten Event-Counter noch tiefergehende Informationen über die ausgeführten Befehle ausgeben könnten ohne den Overhead des Systems zu vergrößern.

4.2 Verwandte Arbeiten

Der Grundstein für dieses Papier sind die Arbeiten [1] und [2] von Frank Bellosa. In diesen wird dargelegt, welchen großen Einfluss die Verwendung von Event-Countern bei der Einschätzung des Energieverbrauchs im laufenden System haben. Dabei konkretisiert das Papier die bereits dargelegten Vorteile der Event-Counter und Energieverbrauchsmustern von einzelnen Tasks und setzt dies in eine anwendbare Technik um.

Eine andere Herangehensweise an Energiemanagement wurde auch von Farkas et al. in [8] gewählt, jedoch stützt sich diese Arbeit beim Energiemanagement auf das Aufbereiten und Handling des von der Applikation verwendeten Codes. Zum Beispiel kann hier das Vorladen von Klassen dabei helfen auftretende Verbrauchsspitzen zu vermeiden. Auch diese Arbeit erzielt damit eine Energieeinsparung in derselben Größenordnung von bis zu 25%.

Event-Counter werden auch von Russ Joseph und Margaret Martonosi zur Einschätzung des Energieverbrauchs eines gesamten Systems in [15] verwendet. Das Ziel dieser Arbeit liegt jedoch darauf, den Energieverbrauch eines Systems mittels Softwaremethoden einzuschätzen, nicht um an bestimmter Stelle Energie einzusparen. Eine Einschätzung des Energieverbrauchs wird auch von Russ Joseph et al. in [14] angestrebt. Der Fokus der Arbeit lag hier aber darauf,

den Energiehaushalt des gesamten Systems mit Hardwarekomponenten zu erfassen.

Die Arbeit [19] von Pering und Brodersen beschäftigt sich ebenso damit, die Arbeitsweise des Systemschedulers anzupassen, allerdings konzentriert diese sich auf die Anpassung der Versorgungsspannung während der Ausführung. Auch in dieser Arbeit wurde die Erkenntnis gewonnen, dass bei Wartezeiten der CPU eine Drosselung der Voltzahl energetisch von Vorteil ist.

In der Arbeit [12] von Sudhanva Gurumurthi et al. wird ein Entwurf vorgestellt, bei dem ein System und dessen Energiehaushalt ganzheitlich simuliert wird. Die Verwendung dieser Technik würde die unzureichenden Informationen von Event-Countern umgehen und tieferliegende Daten freigeben, was zu einer genaueren Abschätzung führt. Eine Kombination des Papiers mit dieser Arbeit könnte weitere Erkenntnisse liefern.

4.3 Zitate von anderen Arbeiten

Nachfolgende Arbeiten ordnen das Papier der Technik DVFS zu. Diese Technik beschäftigt sich damit, den Energieverbrauch eines Systems während dem Betrieb anzupassen und zu optimieren. Dabei wird die Spannung und die Taktfrequenz des Prozessors während der Verarbeitung angepasst. Das Papier wird hier als Beispiel für inter-task DVFS zitiert. Inter-task DVFS bezeichnet die Technik, die Spannung und Frequenz des Prozessors nicht während eines laufenden Threads sondern dazwischen anzupassen. [26, 9, 10, 13, 21, 25, 17, 6, 20]

Andere Arbeiten, welche sich nur entfernt mit Systemtechnik auseinandersetzen haben das Papier zitiert und nennen es als Beispiel für den Bereich "dynamisches Leistungsmanagement". [23, 11] Außerdem werden verschiedentlich Messwerte bis hin zu mehreren Erkenntnissen aus dem Papier zitiert, worauf sich verschiedene Arbeiten stützen. [3, 18, 5]

In der Arbeit [4] von Choi, Soma, und Pedram dient das Papier als eine Grundlage. Diese Arbeit benutzt dabei intra-task DVFS (im Gegensatz zu inter-task DVFS des Papiers) und eine ähnlich aufgebaute Hardware. Dabei stützt sich die Arbeit komplett auf die Überwachung der Speicherzugriffe und passt zusätzlich auch die Versorgungsspannung des Prozessors an. Hier konnte eine Energieeinsparung zwischen 15% und 70% erzielt werden. Dies kann als Erfolg gewertet werden, da es eine Weiterentwicklung der bestehenden Forschung darstellt und entschieden bessere Ergebnisse erzielen konnte.

4.4 Langfristige Auswirkungen

Zu einem späteren Zeitpunkt wird der Einfluss der im Papier vorgestellten Technik von Le Sueur und Heiser in [16] eher als geringer angesehen, da sich für die heutigen Prozessoren die Einsparmöglichkeiten in Grenzen halten. Dies liegt darin begründet, dass mit Weiterentwicklung von Prozessoren deren Strukturgröße immer weiter verringert wird. Damit einher geht ein anwachsender statischer Energieverbrauch, sowie eine immer kleinere dynamic power range, was die Vorteile bzw. Wirksamkeit der DVFS-Technik weiter verringert. Diese Arbeit hat auch ermittelt, dass eine Drosselung der Taktfrequenz auch mit einem höheren Stromverbrauch einhergehen kann, was den Energiesparmaßnahmen entgegen wirkt. Dabei beschränken sich die Beobachtungen in dieser Arbeit jedoch

auf die Anwendung der DVFS-Technik auf Plattformen, welche hauptsächlich auf Servern verwendet werden. Es wird zusätzlich erwähnt, dass kleiner Systeme, wie z.B. Smartphones immer noch von DVFS profitieren können.

In einer anderen Arbeit [7] von Economou et al. wird die Verwendung von Event-Countern zur Charakterisierung des Energieverbrauchs als zu ungenau angesehen. Es wird argumentiert, dass nur eine begrenzte Anzahl an gleichzeitigen Zählerablesungen in einem System stattfinden kann. Dadurch ist nicht gewährleistet, dass genug Informationen zur Verfügung stehen um die Taktfrequenz effektiv einzustellen. Zudem wird der Einfluss der I/O-Systeme außer Acht gelassen, da diese nicht von Event-Countern abgedeckt sind. Die Funktionalität der Event-Counter sei damit bis zum heutigen Zeitpunkt nicht umfangreich genug.

In [22] von Venkatachalam und Franz wird auf die Nachteile von inter-task DVFS eingegangen, bei dem das Papier als Beispiel genannt wird. Zum einen muss, um wirklich effektiv zu sein, der Verlauf des Threads im Vorfeld genau bekannt sein, da die Taktfrequenz nur zwischen den Threads festgelegt werden kann. Sollte die Applikation einen unbekanntem Verlauf während der Ausführung nehmen, kann es passieren dass entweder ein Verlust der Performanz auftritt oder mehr Energie verbraucht wird als nötig. Zum anderen wird der Code der im Thread verwendeten Applikation außer Acht gelassen. Sollte hier zum Beispiel auf den Speicher oder die Festplatte zugegriffen werden, würde dies - falls bekannt - erhebliche Energieeinsparungen mit sich bringen.

5 ZUSAMMENFASSUNG

Die Autoren stellen eine wirksame Technik vor, mit der es unter Nutzung von Event-Countern und einer Prozesseinheit mit skalierbarer Taktfrequenz möglich ist, Energie einzusparen. Der Beziehung zwischen Verlust an Performanz und Energieverbrauch kommt hier besondere Bedeutung zu. Ist ein gewisser Verlust an Performanz tolerierbar, kann im Gegenzug an entscheidenden Stellen Energie eingespart werden. Dies ist besonders bei einer Interaktion zwischen Prozessor und (Arbeits-)Speicher festzustellen. Durch Speicheraufrufe entstehen Wartezeiten an der CPU. In diesen Wartezeiten kann die CPU-Frequenz ohne größere Leistungsverluste heruntergedrosselt werden, was eine Energieeinsparung bedeutet. Das Papier konnte darin überzeugen, dass eine Energieeinsparung mit dieser Technik möglich ist und wird in verschieden starkem Umfang von anderen Arbeiten zitiert. Es wird als Paradebeispiel für inter-task-DVFS gesehen. Außerdem konnte es zu weiterer Forschung in diesem Bereich motivieren und die vorgestellte Technik in Abwandlungen verbessert werden. Im Kontext von neuerer Formen von Prozessoren wird die im Papier vorgestellte Technik als etwas veraltet angesehen, da neuere Bauweisen von Prozessoren nicht unbedingt von der Drosselung der Frequenz energetisch profitieren. Jedoch gibt es immer noch valide Anwendungsmöglichkeiten, vor allem bei kleineren Systemen. Da die Technik nur mit grundlegenden Bausteinen arbeitet und eine inter-task-Technik darstellt, kann es zu unvorhersehbaren Ergebnissen hinsichtlich des Energieverbrauchs kommen, sollte der Code/Thread komplexer in der Ausführung werden. Daher ist diese Technik sehr gut geeignet für einfache, mobile, eingebettete Systeme mit übersichtlichem Applikationsablauf wie z.B. Smartphones.

REFERENCES

- [1] Frank Bellosa. 2000. The benefits of event-driven energy accounting in power-sensitive systems. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, 37–42.
- [2] Frank Bellosa. 2001. The case for event-driven energy accounting. Tech. rep. Technical Report TR-14-01-07, University of Erlangen, Department of Computer ...
- [3] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. 2004. Dynamic voltage and frequency scaling based on workload decomposition. In *Proceedings of the 2004 international symposium on Low power electronics and design*, 174–179.
- [4] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. 2004. Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times. *IEEE transactions on computer-aided design of integrated circuits and systems*, 24, 1, 18–28.
- [5] Walteneagus Dargie. 2011. Dynamic power management in wireless sensor networks: state-of-the-art. *IEEE Sensors Journal*, 12, 5, 1518–1528.
- [6] Gaurav Dhiman and Tajana Simunic Rosing. 2007. Dynamic voltage frequency scaling for multi-tasking systems using online learning. In *Proceedings of the 2007 international symposium on Low power electronics and design*, 207–212.
- [7] Dimitris Economou, Suzanne Rivoire, Christos Kozyrakis, and Partha Ranganathan. 2006. Full-system power analysis and modeling for server environments.
- [8] Keith I Farkas, Jason Flinn, Godmar Back, Dirk Grunwald, and Jennifer M Anderson. 2000. Quantifying the energy consumption of a pocket computer and a java virtual machine. In *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 252–263.
- [9] Rong Ge, Xizhou Feng, and Kirk W Cameron. 2005. Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters. In *SC'05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. IEEE, 34–34.
- [10] Sriram Govindan, Anand Sivasubramaniam, and Bhuvan Urgaonkar. 2011. Benefits and limitations of tapping into stored energy for datacenters. In *Proceedings of the 38th annual international symposium on Computer architecture*, 341–352.
- [11] Sriram Govindan, Di Wang, Anand Sivasubramaniam, and Bhuvan Urgaonkar. 2012. Leveraging stored energy for handling power emergencies in aggressively provisioned datacenters. In *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, 75–86.
- [12] Sudhanva Gurumurthi, Anand Sivasubramaniam, Mary Jane Irwin, Narayanan Vijaykrishnan, and Mahmut Kandemir. 2002. Using complete machine simulation for software power estimation: the softwatt approach. In *Proceedings Eighth International Symposium on High Performance Computer Architecture*. IEEE, 141–150.
- [13] Canturk Isci, Gilberto Contreras, and Margaret Martonosi. 2006. Live, runtime phase monitoring and prediction on real systems with application to dynamic power management. In *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*. IEEE, 359–370.
- [14] Russ Joseph, David Brooks, and Margaret Martonosi. 2001. Live, runtime power measurements as a foundation for evaluating power/performance tradeoffs. In *Workshop on Complexity Effective Design WCED, held in conjunction with ISCA*. Vol. 28.
- [15] Russ Joseph and Margaret Martonosi. 2001. Run-time power estimation in high performance microprocessors. In *Proceedings of the 2001 international symposium on Low power electronics and design*, 135–140.
- [16] Etienne Le Sueur and Gernot Heiser. 2010. Dynamic voltage and frequency scaling: the laws of diminishing returns. In *Proceedings of the 2010 international conference on Power aware computing and systems*, 1–8.
- [17] Lianpeng Li, Jian Dong, Decheng Zuo, and Jin Wu. 2019. Sla-aware and energy-efficient vm consolidation in cloud data centers using robust linear regression prediction model. *IEEE Access*, 7, 9490–9500.
- [18] Andreas Merkel, Jan Stoess, and Frank Bellosa. 2010. Resource-conscious scheduling for energy efficiency on multicore processors. In *Proceedings of the 5th European conference on Computer systems*, 153–166.
- [19] Trevor Pering and Robert Brodersen. 1998. Energy efficient voltage scheduling for real-time operating systems. In *Proceedings of the 4th Real-Time Technology and Applications Symposium RTAS*. Vol. 98. Citeseer.
- [20] Basireddy Karunakar Reddy, Amit Kumar Singh, Dwaipayan Biswas, Geoff V Merrett, and Bashir M Al-Hashimi. 2017. Inter-cluster thread-to-core mapping and dvfs on heterogeneous multi-cores. *IEEE Transactions on Multi-Scale Computing Systems*, 4, 3, 369–382.
- [21] Suzanne Rivoire, Mehul A Shah, Parthasarathy Ranganathan, and Christos Kozyrakis. 2007. Joulesort: a balanced energy-efficiency benchmark. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 365–376.
- [22] Vasanth Venkatachalam and Michael Franz. 2005. Power reduction techniques for microprocessor systems. *ACM Computing Surveys (CSUR)*, 37, 3, 195–237.
- [23] Kun Wang, Miao Du, Dejun Yang, Chunsheng Zhu, Jian Shen, and Yan Zhang. 2016. Game-theory-based active defense for intrusion detection in cyber-physical embedded systems. (2016).
- [24] Andreas Weisell and Frank Bellosa. 2002. Process cruise control: event-driven clock scaling for dynamic power management. In *Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems*, 238–246.
- [25] Qiang Wu, Vijay J Reddi, Youfeng Wu, Jin Lee, Dan Connors, David Brooks, Margaret Martonosi, and Douglas W Clark. 2005. A dynamic compilation framework for controlling microprocessor energy and performance. In *38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05)*. IEEE, 12–pp.
- [26] Zhou Zhou, Jemal Abawajy, Morshed Chowdhury, Zhigang Hu, Keqin Li, Hongbing Cheng, Abdulhameed A Alelaiwi, and Fangmin Li. 2018. Minimizing sla violation and power consumption in cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, 86, 836–850.