CompatPM: Enabling Energy Efficient Multimedia Workloads for Distributed Mobile Platforms

Ripal Nathuji, Keith J. O'Hara, Karsten Schwan, Tucker Balch

College of Computing
Georgia Institute of Technology
Atlanta, GA 30032
{rnathuji, kjohara, schwan, tucker}@cc.gatech.edu

ABSTRACT

The computation and communication abilities of modern platforms are enabling increasingly capable cooperative distributed mobile systems. An example is distributed multimedia processing of sensor data in robots deployed for search and rescue, where a system manager can exploit the application's cooperative nature to optimize the distribution of roles and tasks in order to successfully accomplish the mission. Because of limited battery capacities, a critical task a manager must perform is online energy management. While support for power management has become common for the components that populate mobile platforms, what is lacking is integration and explicit coordination across the different management actions performed in a variety of system layers. This papers develops an integration approach for distributed multimedia applications, where a global manager specifies both a power operating point and a workload for a node to execute. Surprisingly, when jointly considering power and QoS, experimental evaluations show that using a simple deadline-driven approach to assigning frequencies can be non-optimal. These trends are further affected by certain characteristics of underlying power management mechanisms, which in our research, are identified as groupings that classify component power management as "compatible" (VFC) or "incompatible" (VFI) with voltage and frequency scaling. We build on these findings to develop CompatPM, a vertically integrated control strategy for power management in distributed mobile systems. Experimental evaluations of CompatPM indicate average energy improvements of 8% when platform resources are managed jointly rather than independently, demonstrating that previous attempts to maximize battery life by simply minimizing frequency are inappropriate from a platform-level perspective.

Keywords: Power management, distributed mobile systems

1. INTRODUCTION

Technological advancements in the areas of computational and communication resources have led to the deployment of increasingly sophisticated mobile distributed systems. Since many of the applications running across these systems are cooperative in nature, there is considerable flexibility in distributing the associated computations to improve performance, gain fault tolerance, or, the focus of this work, reduce energy consumption. As a concrete example, cooperative multimedia applications can be found in autonomous mobile robots performing critical search and rescue missions. In this scenario, robots use sensors as input for navigation algorithms and situational awareness. The resulting multimedia data flows are coordinated by a system manager controlling the deployment and configuration of the associated software components.

For modern robotic systems, the energy used by distributed multimedia tasks can substantially contribute to total power consumption. Since mobile robots have a limited energy supply, efficient energy management translates into prolonged lifetimes and more successful missions. The computing platforms used in these systems enable energy efficiency with integrated components that provide multiple power management states. Power management of these resources is typically performed at multiple layers. Indeed, for certain components, control naturally belongs in one layer over another. For example, in order to obtain memory power savings during active periods, the time granularity of management is much finer than the operating system can attain. Therefore, the management of this resource must be performed in the platform itself via integrated hardware solutions. Utilizing dynamic voltage and frequency scaling (DVFS) is interesting as its control can feasibly be assigned to multiple layers. Since frequency scaling of processors directly affects the end performance of applications, in a distributed environment where global time constraints are not known at each node, the control of processor operating points naturally belongs in the system management component of the architecture.

Our previous work on energy efficient software systems[1] has investigated the deployment of application tasks across participants in a distributed system, including considering the tradeoffs between energy overheads of computation and communication when offloading work. This paper considers the additional gains possible when allowing the system deployment manager to specify a processor operating point at which to execute a multimedia workload deployed onto a robot. We begin by analyzing the energy tradeoffs of frequency scaling without the use of other power management mechanisms and find that: (1) energy savings attained with DVFS on the processor need not directly translate to platform-level savings, and therefore, (2) even when the lowest operating point can be used, from a platform-level perspective, it may actually consume more energy than a higher frequency. To effectively consider how these trends are affected by underlying power management mechanisms, we propose the notion of power management compatibility. In particular, there are those power management schemes whose achievable savings are not affected by frequency scaling and can be called "compatible" with DVFS (VFC), and others whose effects vary significantly depending on processor modes and are "incompatible" (VFI).

Prior work has already shown it useful to have power management integrate across different OS subsystems for online power management, including rescheduling processes to improve power consumption of peripheral devices.[2,3] Therefore, in this research we adopt a vertically integrated solution for power management in cooperative distributed systems and propose the CompatPM architecture. The approach requires attributes of platforms, workloads, and underlying power management schemes to be exported so that the CompatPM enabled system manager can properly assign performance points given the QoS constraints of the distributed system and applications. In our evaluation we find average improvements of 8% in energy savings compared to a simple heuristic that selects the lowest operating point which meets latency constraints.

2. RELATED WORK

In a study of the Pioneer DX-3 mobile robot, motion was 12.1%–44.6% of the energy used depending on speed, and the embedded computer accounted for 33.3%–65.3% of the energy.[4] Since energy for computation will increase as robots become more autonomous and use more powerful computer platforms such as multicore systems, the importance of power management for sustaining missions of multi-robot teams will increase in future platforms. Towards reducing the energy consumption of mobile devices, Zeng *et al.*[5] make power a first class resource in order to provide system lifetime guarantees, and describe a scheduler[6] that allocates power to processes, considering various scheduling algorithms. Another energy accounting approach[7] utilizes hardware performance counters to estimate energy usage and schedules accordingly.

Since the dynamic power consumption of a CPU is proportional to the product of frequency and voltage squared, dynamic voltage and frequency scaling can be effective in reducing power consumption during program execution.[8] A design framework for exploring power/performance trade-offs when developing hard real-time systems has been proposed,[9] as well as an offline scheduler coupled with online slack reclaiming for enhancing the benefits of DVFS.[10] Our own past research exploits the ability to obtain energy savings by performing application-level adaptations for multimedia applications.[11] Other research aggressively pursues reduced frequencies within the constraints of application deadlines using memory access information.[12] In all of these DVFS approaches, the effect of frequency scaling on the CPU energy signature is considered independently of the rest of the system. Fan *et al.*[13] begin to move away from this assumption by investigating the synergy between DVFS and power-aware memory systems. Miyoshi *et al.*[14] question the underlying assumption of DVFS, that frequency should be reduced whenever possible, by isolating poor performance points created by efficient idle modes. Similarly, the optimality of lower frequency points when considering system sleep modes has also been studied,[15] as well as the tradeoff of overheads such as processor leakage power and extended resource standby times.[16,17] As part of this research, we evaluate similar tradeoffs in the context of multimedia workloads.

3. ENERGY EFFICIENT DEPLOYMENT OF MULTIMEDIA TASKS

3.1. Multimedia Workloads in Cooperative Distributed Systems

An underlying assumption of this research is that it is possible to exploit the cooperative nature of distributed multimedia applications, subject to some set of joint operating constraints. An example in mobile robotics is a search and rescue scenario in which robots have to preserve the energy of all team members in order to sustain a longer application lifetime. The multimedia workloads of these robotic systems subject to runtime management may be divided into two groups: (1) those responsible for providing situational awareness for human operators and (2) those in which robots consume and process media data. Workloads in (1) are similar to traditional multimedia workloads, since a human consumes the data. For example, the robot may use a microphone along with a camera to relay information from a victim to a human operator. For this task, the robot would have to perform encoding and transmission of the input data. The audio and image encoding and decoding algorithms we use in our evaluation are representative of such situational awareness workloads. Workloads in (2) use some combination of sensors such as cameras, GPS, sonar, and odometers for robot navigation. These sensor data flows must be processed by perceptual algorithms, either locally or on another robot in a cooperative team, to be useful for navigation. As an example, in our previous work[1] we considered a robot to robot data flow for blob finding analysis of images, where energy was reduced by offloading image processing. Such offloading relies on image encoding and decoding to minimize communication energy consumption. The JPEG workloads used in our experiments are typical of multimedia data flows for distributed image processing.

We capture the distributed nature of our multimedia workloads by representing them as “task chains” of computation and communication. Tasks are units of functionality (e.g., JPEG encoding, edge detection) as well as units of allocation. That is, each task may be allocated anywhere in the mobile distributed system, and adjacent tasks are connected via communication operations across a wireless network when necessary. Our previous work considered the allocation of an application task chain among multiple mobile robots so as to maximize system lifetime. In this paper, we consider increasing the power efficiency at nodes after allocation by exploiting local power saving mechanisms.

3.2. Power Management Architecture

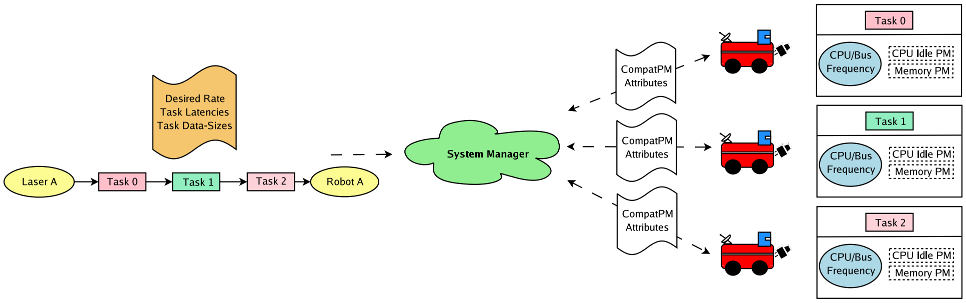


Figure 1. System Power Management Infrastructure

Our power management architecture assumes that task chains are known to, and manipulated by, system managers as shown in Figure 1. Such managers are useful because they can exploit their global knowledge to perform better allocations and select appropriate operating points of tasks on participating platforms. System managers are also necessary because allocation and selection decisions are not just based on task workloads, but also on end-to-end application-level constraints such as meeting some maximum delay for gaining perceptual insight from a set of raw sensor data readings. A common approach in past work has been to compute the minimum frequencies needed along the chain to meet end-to-end delay requirements, thereby exploiting the slack available in distributed multimedia or real-time applications. Unfortunately, as our results will show, this can result in suboptimal system-level energy consumption. In addition, such an approach does not capture the interactions between the execution speed chosen and the savings achieved by the underlying power management mechanisms integrated into the mobile platform. CompatPM utilizes vertical integration to exploit these relationships. Towards this end, we identify various attributes that must be provided to the software management layer so that it may make effective decisions in a scalable manner. This approach is summarized in Figure 1.

3.3. Evaluation Infrastructure

Experimental results are based on measurements obtained from representative embedded hardware, using Intel’s Sitsang-400 evaluation platform designed around the PXA255 processor. The PXA255 supports multiple operating points that vary CPU frequency as well as the frequency to the internal PXA bus, thereby affecting latency to memory and I/O devices. These points, along with the associated core voltages, are (core/bus@voltage): 400MHz/200MHz@1.3V, 400MHz/100MHz@1.3V, 300MHz/100MHz@1.1V, 300MHz/50MHz@1.1V, 200MHz/100MHz@1.0V, 200MHz/50MHz@1.0V, 150MHz/50MHz@1.0V, and 100MHz/50MHz@1.0V. Our experimental workloads consist of benchmarks from the Mediabench suite that exemplify the two types of multimedia workloads for robotic systems discussed in Section 3.1 including both the encoders and decoders for `adpcm`, `g721`, `gsm`, and `jpeg`. Benchmark execution time and power consumption are monitored for every available operating point. All power measurements are performed using a Tektronix TDS5104B oscilloscope, Tektronix TCP202 current probes, and Tektronix P6139A voltage probes.

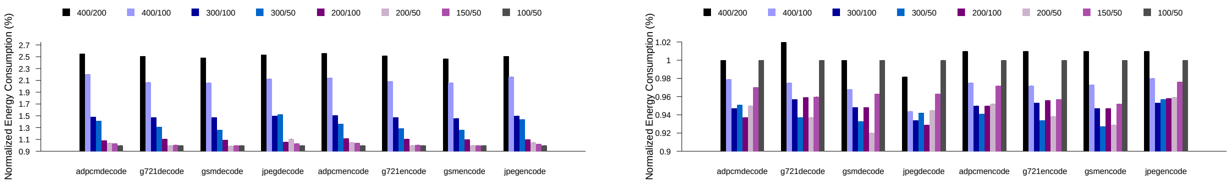
4. ENERGY ANALYSIS OF MULTIMEDIA APPLICATIONS

Table 1. Sitsang Power Consumption Overview

	Average Power Consumption (mW) per Operating Point (Core/Bus Frequency)							
Scenario	400/200	400/100	300/100	300/50	200/100	200/50	150/50	100/50
System Idle	1339	1281	1255	1223	1229	1213	1213	1213
System Active	2351	2230	1985	1858	1798	1714	1666	1587
CPU Idle	126	90	63	44	44	33	33	32
CPU Active	449	400	235	206	140	126	106	82

When managing the power consumption of computational platform components, the use of frequency reduction must be balanced with the resulting performance degradation. In periodic real-time systems, this tradeoff can be formulated in a precise manner: if the execution time of a task at frequency f_i is less than or equal to the period (deadline) T, then f_i is a plausible frequency for execution. For the remainder of this section, we assume that all of the operating modes supported by the PXA255 are plausible for the execution of the assigned task. In particular, we assume the deadline/periodicity for the multimedia component of a deployed application chain to be the execution time at the lowest frequency. An overview of the Sitsang platform power characteristics when active and idle at various operating points is provided in Table 1. As expected, the power consumption of both the CPU and system decrease with reduced operating points. In the entirety of this paper, energy results for the CPU and system are normalized to the execution energies measured at the 100MHz/50MHz operating point.

Frequency Scaling and Energy Tradeoffs. Due to the periodic nature of multimedia workloads, the metric for energy consumption can be formalized using the notion of cycle energy. The *cycle energy*, E_{cycle}, consists of the sum of the execution energy E_{exec} and the idle time energy E_{idle} for a given task. We assume that the deadline of each task is its execution time at 100MHz in order to calculate cycle energy values.

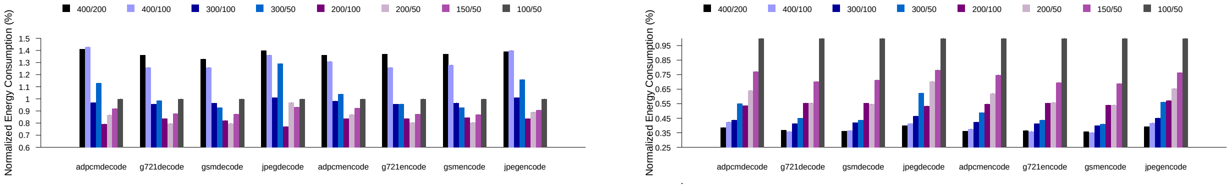


(a) CPU Energy Results

(b) System Energy Results

Figure 2. 'Cycle Energy' Consumption

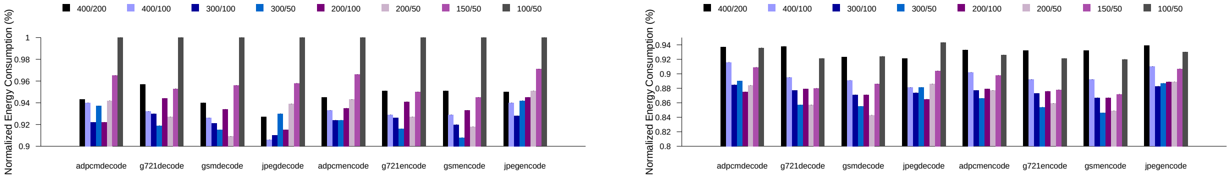
Figure 2 provides the CPU and system-level results for cycle energy. We observe a clear trend in CPU cycle energy, where consumption generally decreases with lower performance points. These results support the intuition used in various real-time and multimedia schedulers that attempt to minimize frequency whenever it is possible to do so without violating performance constraints. Indeed, we observe CPU energy benefits of up to 60% when executing at the lowest frequency as opposed to the highest. Unfortunately, in practice, this intuition is only partially correct. Figure 2(b) provides system-level ‘cycle energy’ data. An important result in this figure is that the optimal operating point is never the lowest frequency. Indeed, the optimal frequency provides benefits of up to 8% compared to the lowest frequency. This result contradicts the assumptions made in prior work that reducing processor frequency is always beneficial to system energy consumption. We also observe interesting trends with bus frequencies. The system-level cycle energies of the `adpcmdecode` and `jpegdecode` benchmarks are reduced when executing at 300/100 versus 300/50, while the reverse is true for `g721decode`. This illustrates that the relative benefits of operating points are workload dependent for actual systems and applications.



(a) CPU Energy Results (b) System Energy Results

Figure 3. Execution Energy Consumption

There is increasing support towards minimizing the power consumption of components when the system is not actively executing applications. We consider this trend by observing execution energies only. In Figure 3(a), the CPU execution energy generally decreases with core frequency (as expected), but this holds only until the 150MHz operating point due to the fact that while performance continues to degrade in proportion to frequency, the power benefits decrease as shown in Table 1. The reason is that the core voltage is the same for the lower frequencies, thereby removing any voltage scaling benefits*. With regard to system execution energy trends, it can be seen that execution energy is minimized at the maximum core frequency (the particular bus frequency depending on the benchmark). These results support an approach that executes programs quickly in order to optimize idle periods. Since our idealistic assumption of zero idle energy does not hold for current platforms, the optimal operating point will fall somewhere in between the minimum and the maximum. We therefore identify CompatPM attributes that allow for the dynamic assessment of tradeoffs between these operating points.



(a) DVFS with CPU Sleep Management (b) DVFS with Memory PM

Figure 4. DVFS with Underlying PM

Interaction of DVFS and Incompatible Schemes. DVFS inhibits the use of VFI mechanisms, for instance, due to a reduction in the slack time t_{idle}. As an example, we consider the use of utilizing CPU sleep state into our measurements. For the PXA255, the sleep mode consumes approximately 45μA, a relatively negligible amount of power. We utilize this value to project the cycle energy savings when combining the use of processor sleep state with DVFS. Figure 4(a) provides the normalized system-level energy savings when using both DVFS and CPU sleep power management. Comparing it to Figure 2(b), we see that the benefits of higher operating points increase quite dramatically in relation to lower frequencies. We conclude that when performing frequency scaling in conjunction with VFI power management mechanisms, the cycle energy trends move towards the execution

---

*The PXA255 electrical specifications prescribe the operating points used, along with the 1V minimum requirement.

energy trends of a platform. This is key background to our argument for vertical integration: our management layer should be made aware of these types of underlying power managers.

Interaction of DVFS and Compatible Schemes. While certain types of power management mechanisms do not coexist well with DVFS, others work synergistically with processor management. These VFC power management mechanisms can reduce the energy expended during execution periods by performing active management. Moreover, a VFC mechanism not only reduces energy during execution time, but can also be used during idle periods. Indeed, VFC power management mechanisms simply shift savings from idle to execution periods when frequency is scaled down creating independence from DVFS. Previous work has shown that memory power management can be utilized simultaneously with frequency scaling.[13] Therefore, we use this power management scheme as our VFC example. Using XScale performance counters, we obtain execution traces of memory usage with our workloads at each operating point. These traces are then used to estimate the power savings that can be incurred with the two Samsung K4S561632D[†] parts incorporated into the Sitsang platform using timing values similar to those assumed in previous studies (100ns cache miss penalty and 10ns transition time from low power to active mode).[13] The resulting power signatures are presented in Figure 4(b). We observe from the figure that memory power management can provide significant power savings on the system across all frequencies, compared to the default platform. Indeed, the benefits achieved are up to 16% in system energy when compared to executing at 100MHz with no memory management. The general trends between operating points, however, are similar to Figure 2(b). This result illustrates the relative independence of VFC power mechanisms and shows that this class of power managers can exist transparently to frequency assignment decisions.

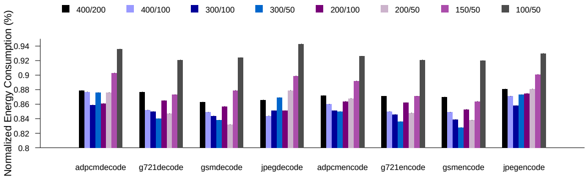


Figure 5. Complete System PM

Combining VFI and VFC Methods. We next obtain results when both of our power management examples are incorporated with DVFS. In Figure 5 we observe that the trends closely resemble the VFI results in Figure 4(a), and the optimal operating point shifts towards higher frequencies. This tells us that the tradeoff trends between operating points when using DVFS with both types of management schemes continue to be influenced by VFI power management mechanisms. Moreover, even when coupled with VFI mechanisms, the independence of VFC managers remains intact.

5. EVALUATION OF COMPATPM

The previous section presented an in depth study of the system-level power tradeoffs when using different operating points for a processor, as well as the interactions between utilizing this power management support with other underlying mechanisms. The goal of CompatPM is to take these types of tradeoffs and interactions into account in order to assign efficient operating points at runtime. It is clear that utilizing a profile-based approach where the manager is provided all of these tradeoffs explicitly does not scale. Therefore, we begin by defining a limited set of attributes that should be exported to the system manager as per Figure 1, and then determine the effectiveness of decisions based upon these inputs.

5.1. CompatPM Attributes

In terms of platform hardware, the system manager must be made aware of the inherent power tradeoffs of the system independent of any other power management. In our evaluation, we use a simple approach wherein average active/idle power values per operating mode for the platform are specified in the CompatPM attributes.

[†] We define 165mW idle power and 40mW sleep power for these chips based upon datasheet values.

For improved accuracy, a more sophisticated specification could be used. An example is a parameterized power model that is a function of workload characteristics such as memory access rate. Our experimental results have highlighted the need for the system manager to understand the underlying power management methods supported by a robot's computational platform. Moreover, the results have shown the benefits of our classification scheme for these power managers into VFI and VFC mechanisms, as the CompatPM based management can essentially ignore the existence of the latter type of schemes. Since savings from VFI mechanisms can only be obtained during idle periods, in its CompatPM attributes list, a node should specify the average power savings achievable when idle for each VFI mechanism it supports. Two workload specific items must be exported to the management layer. First, the latency at the lowest operating point of a platform should be specified. Second, in our measurements, we observed that given a processor frequency, the particular bus frequency that is most energy efficient is workload specific. Therefore, we require workload specific memory access behavior in order to differentiate what bus frequency to utilize. For our system, the memory access rate (MAR), or the number of memory accesses per instruction, is provided for each benchmark. Since memory access behavior is independent of frequency, this attribute does not need to be specified on a per operating point basis.

5.2. Results

Given the four CompatPM attributes: (1) active/idle hardware power characteristics, (2) average power savings for VFI schemes, (3) workload specific execution time at the lowest frequency, and (4) memory access rate information for applications, the management layer attempts to determine an operating point for a workload that will maximize system-level power efficiency including the possible benefits of underlying power management schemes on a platform. This process is driven by comparing the estimated energy savings at various points to running at the lowest operating point which meets latency requirements. In order to perform the estimation, the manager must determine what fraction of the period will be spent actively executing workloads, and what portion will be idle. We propose to use a model driven approach to estimate the active time for execution in various operating modes using limited information, e.g. given the execution time, t_{min}, at the lowest performance state provided by a platform in its CompatPM attributes. Instead of benchmarking each workload at every possible frequency, for scalability reasons, we hope to use a small number of features to predict the performance at different operating points. In our evaluation, we analyzed different learning algorithms and different feature sets for predicting the workload latencies at different operating points. Using data from workloads at various frequencies, we found a simple function of t_{min} and the processor core frequency f_{core} under consideration to be the best predictor: $t_x = 99.88 * \frac{t_{min}}{f_{core}} + 0.01$.

Once the manager has determined the division of a period into its active and idle times, it uses the power related attributes to estimate power savings and thereby the optimal frequency point. As a last step of the decision process, we propose a heuristic that allows the system to determine an appropriate bus frequency given a core frequency. By analyzing the MAR values of each of our workloads, we find that, generally, workloads with relatively high MAR (greater than 1%) benefit from a higher bus frequency, and vice versa. Therefore, we use this correlation between MAR and efficient bus frequencies to decide the bus operating point for an application.

Given the decision process based upon CompatPM attributes, we evaluate the resulting chosen operating point for a workload. The mode is compared to the optimal found by doing a full system analysis as in Section 4, and the mode that would be chosen according to traditional constraints, i.e. the lowest possible operating point which meets performance requirements. We observe two conclusive results. First, we observe an average improvement of 8% in energy consumption with CompatPM compared to utilizing the minimum frequency. Second, our CompatPM enabled system is often able to accurately determine the optimal operating point for each workload. Even including the discrepancies between CompatPM and the optimal energy consumption, our system consumes less than 1% more energy on average. These results show that by utilizing our in depth measurements and analysis, we have successfully isolated a set of CompatPM attributes that can be easily provided to the management layer of a distributed system to create an effective, scalable, and vertically integrated power management approach.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigate multimedia applications in cooperative distributed systems using the example of a team of mobile robots deployed in a search and rescue mission. Due to power constraints on mobile systems, performing energy management can directly affect the success of this cooperative scenario. Towards this end, we investigate the ability to optimize energy efficiency by developing CompatPM, a component of a distributed system manager which utilizes vertical integration into the underlying system layers that perform power management for mobile platforms in order to optimize overall power performance when assigning operating points for workload execution. For scalability we define a set of attributes that should be provided to CompatPM by participating nodes. Our experimental results highlight the ability of CompatPM to then help provide improved energy efficiency compared to previously used heuristics for frequency scaling. Our future work will extend these results by considering multiple tasks executing concurrently on a mobile robot as well as integrating operating point assignment with task chain allocation for increased effectiveness.

REFERENCES

1. K. O'Hara, R. Nathuji, H. Raj, K. Schwan, and T. Balch, "Autopower: Toward energy-aware software systems for distributed mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.
2. Y. Lu, L. Benini, and G. Micheli, "Low-power task scheduling for multiple devices," *In 8th International Workshop on Hardware/Software Codesign* , pp. 39–43, 2000.
3. R. Nathuji and K. Schwan, "Reducing system level power consumption for mobile and embedded platforms," *In Proceedings of the International Conference on Architecture of Computing Systems (ARCS)* , March 2005.
4. Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
5. H. Zeng, X. Fan, C. Ellis, A. Lebeck, and A. Vahdat, "Ecosystem: Managing energy as a first class operating system resource," *In Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)* , October 2002.
6. H. Zeng, C. Ellis, A. Lebeck, and A. Vahdat, "Currentcy: A unifying abstraction for expressing energy management policies," *In Proceedings of USENIX* , pp. 43–56, 2003.
7. F. Bellosa, "The benefits of event-driven energy accounting in power-sensitive systems," *In Proceedings of the 9th ACM SIGOPS European Workshop* , September 2000.
8. M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced cpu energy," *In Proceedings of the First Symposium on Operating Systems Design and Implementation* , pp. 13–23, November 1994.
9. P. Chou, J. Liu, D. Li, and N. Bagherzadeh, "IMPACCT: Methodology and tools for power-aware embedded systems," *Design Automation for Embedded Systems, Special Issue on Design Methodologies and Tools for Real-Time Embedded Systems* , October 2002.
10. H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, "Power-aware scheduling for periodic real-time tasks," *In IEEE Transactions on Computers* **53**, May 2004.
11. C. Poellabauer and K. Schwan, "Power-aware video decoding using real-time event handlers," *In Proceedings of the 5th International Workshop on Wireless Mobile Multimedia (WoWMoM)* , September 2002.
12. L. Singleton, C. Poellabauer, and K. Schwan, "Monitoring of cache miss rates for accurate dynamic voltage and frequency scaling," *In Proceedings of the Multimedia Computing and Networking Conference (MMCN)* , 2005.
13. X. Fan, C. Ellis, and A. Lebeck, "The synergy between power-aware memory systems and processor voltage scaling," *In Proceedings of the Workshop on Power-Aware Computer Systems (PACS)* , December 2003.
14. A. Miyoshi, C. Lefurgy, E. Van Hensbergen, R. Rajamony, and R. Rajkumar, "Critical power slope: Understanding the runtime effects of frequency scaling," *In Proceedings of the 16th Annual ACM International Conference on Supercomputing* , June 2002.
15. S. Irani, S. Shukla, and R. Gupta, "Algorithms for power savings," *In Proceedings of the 14th Symposium on Discrete Algorithms* , 2003.
16. R. Jejurikar and R. Gupta, "Dynamic voltage scaling for system-wide energy minimization in real-time embedded systems," *In Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)* , August 2004.
17. R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," *In Proceedings of the IEEE Design Automation Conference (DAC)* , 2004.