Microelectronics System Design Research Group
School of Electrical, Electronic & Computer Engineering
Merz Court
Newcastle University
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Game Theoretic Power Management Techniques for Real-Time Scheduling

James Docherty, Alex Yakovlev

james.docherty@ncl.ac.uk, alex.yakovlev@ncl.ac.uk

NCL-EECE-MSD-MEMO-2011-002

January 2011

Abstract

This paper summarizes the initial investigation into game theory applications for real time systems. A brief overview of game theory is presented, along with possible uses to manage energy consumed by processes within real-time processing and ensure maximum reliability while managing all available power from a source such as an energy harvesting device. Possibilities of retaining MTTF and MTBF while applying low power techniques such as DVFS are also proposed as investigation items.

Introduction

Power Management

The past decade has seen a proliferation of battery-powered devices for both commercial and domestic applications. Users of these devices expect greater functionality with each evolution in design, placing severe design constraints on the processors used. The chips have increased in both computation speed and power consumption, meaning power management has now become a critical aspect of modern system design.

Many techniques exist for reducing the power consumed by a chip, which can be applied at all levels of design abstraction. Some techniques aim to reduce the static power consumption, as up to 70% of energy in a chip is wasted in standby [1], so items such as clock gating and sleep modes are commonly used to reduce this value to a more manageable level [2]. However, if a chips duty cycle is high, the time where these methods could be used is offset by the power required to store data in sleeping cores or to “wake up” areas in dormant states. Therefore, dynamic power dissipation must also be minimized, especially within real-time systems [3]. Commonly, Dynamic Voltage Frequency Scaling (DVFS) or critical speed strategies are used within real-time systems for power management. DVFS reduces power consumption at the penalty of increased execution time, while critical speed balances the duty cycle to minimize the active energy. Both techniques have their benefits and must be managed based on the jobs and schedule loaded into the processor. This means schedulers must now be energy aware and provide balance between the power consumption within a device and its quality of service, measured by its reliability and execution speed.

As recent developments have allowed the capture and conversion of small volumes (tens of Joules) of “wasted” energy from kinetic, thermal or solar sources [4, 5], this can now also be incorporated into the energy used to power a microprocessor. However, using this presents difficulties in power management due to the inconsistent delivery and availability of energy. Therefore, new techniques for managing and delivering energy to the sections of an integrated circuit are a key development item for the next generation of embedded systems.

Real Time Systems

Real-time systems run a set of tasks; all of which have a desired start, execution and end time (deadline). If a task takes longer than its end time to complete, external items could be jeopardized. Where a missed deadline could be catastrophic, it is said to be a “hard” deadline, while one that can be missed at a reduction of usefulness is said to be “soft” [6].

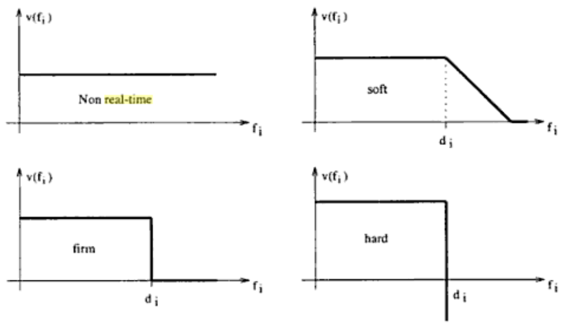

Figure 1 - Diagram showing usefulness of tasks after their deadline. Non-real-time tasks have no deadline, so no reduction in usefulness. Soft tasks reduce in usefulness after their deadline, eventually reaching zero. Firm tasks immediately drop to zero, showing the data can still be used but is essentially useless. Hard tasks drop to minus infinity, as a miss of these can lead to serious issues.

To ensure tasks do not miss deadlines (overrun), scheduling algorithms measure internal metrics and adjust parameters such as execution order to ensure as many tasks as possible can complete before their deadline.

Game Theory

Game theory was developed in the 1940’s by John von Neumann and initially published in [7]. It has been used in a wide variety of fields from economics to evolutionary biology, with some focus on its use within electronic engineering for queuing theory within protocols such as TCP/IP [8]. The primary idea of game theory is that multiple players decide a strategy to maximize their return, which is typically a theoretical “util” value. If both players decide on strategies to maximize their return irrelevant of their opponents strategy, the game is said to contain a Nash Equilibrium (named after John Forbes Nash – the original proposer [9]).

Game theory is altered according to the “rules” of the game. For example, players may know the strategy chosen by their opponent and therefore alter their strategy accordingly. Games can be split into non-cooperative and cooperative, with a great deal of work concentrating on non-cooperative games for power management. However, some anecdotal evidence [10] suggests that a cooperative game could lead to greater savings for the small overhead increase of a power management kernel.

Methodology

The initial goal is to investigate the use of game theory within schedulers for real-time systems and asynchronous arbiters, particularly where power management is used [11]. For this, the proposed model assumes energy is finite and to be shared by all tasks within a time frame. If the energy available is greater than the energy required for all tasks, then every task will complete on time (provided a feasible schedule exists) and no issues will occur. However, if the energy collected can only run a percentage of tasks, then tasks must enter negotiations to ensure they secure energy for their execution. This differs from previously published works [12, 13] as tasks enter a cooperative game to maximise their share. This game takes the form of a negotiation, with a fixed time limit declared by the power management kernel. If the jobs cannot agree to a fair share of the available energy, a Disagreement Payoff Point (DPP) is used. Figure 2 shows a two-job example.

(1)

		B	
		Share	Steal
A	Share	***100,100***	***100,100***
	Steal	***100,100***	***100,100***

(2)

		B	
		Share	Steal
A	Share	***20,20***	100,10
	Steal	10,100	0,0

Figure 2 – A two job game with High Energy (1) and Low Energy (2) availability. In (1) all tasks can 100% execute, but in (2) only 20% each will be executed if the tasks share the energy, while stealing the energy will allow a 100%/10% split between the stealer and sharer. DPP is set within the kernel as 40% each

For a non-real time system, example two could be deemed acceptable, but if process A in the above example is a hard real-time process, the results could be catastrophic. In this case, A's ideal tactic would be to steal all the energy and prevent B from executing, despite this not being a Nash Equilibrium. While only 10% of B would be executed; if B were a soft real-time event, completing 10% of the task could be seen as acceptable providing A completely executes. A graph of the possible values for negotiation can be seen in Figure 4. However, negotiation will not allow 100% execution of any job in the current game matrix. For this to happen, A must request more energy than it needs to negotiate an acceptable settlement.

To overcome this, tasks may form cartels and share energy to ensure the most important item executes successfully. Figure 3 shows the possible game matrix with cartel forming. (1) shows the same model as Figure 2 with A as a hard-real-time process, but (2) adds cooperation. B yields to A's importance and uses DVFS to reduce its energy use, meaning both can execute to completion

This investigation aims to develop a microprocessor capable of working with minimal variation in reliability for critical tasks independent of the available energy within the system.

The reliability will be measured by the Mean Time to Failure (MTTF), found through experimental runs of the real-time system, regarding a failure as a miss of any hard task. Mean Time Between Failures (MTBF) can also be calculated by monitoring the system following a failure to see when a return to normal operation takes place [14].

It is hoped that this work could lead to the development of a dynamic scheduling algorithm for use within real time systems where power is not consistently delivered.

(1)

		B: Share	B: Steal
A	Share	20,20	100,0
A	Steal	0,100	0,0

(2)

		B: Share	B: Steal
A	Share	**100,100**	100,0
A	Steal	0,100	0,0

Figure 3 - Game Matrix for Figure 2 with Cartel Forming, allowing execution of Hard Real Time Task

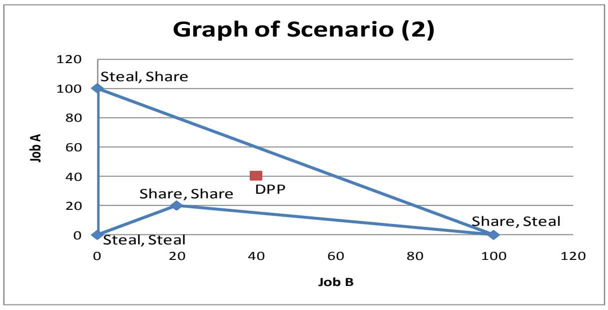

Figure 4 - Graph showing all possible positions for Figure 2 in a Low-Energy State

Conclusions

As this investigation continues, it is hoped reliability can be retained with a reduced energy budget. It is hoped that a game theoretic algorithm can be developed which gives improved performance across systems of unspecified size. It is probable that the game used will change as factors within the entire system alter, with larger models possibly requiring the use of arbiter systems to ensure fairness with minimal overhead penalty for their use. These would act as “men-in-the-middle” and have the final decision on power assignment in the event of a tie between two processes. While this would have a negligible effect on small systems, larger systems may experience reliability and overall power consumption benefit for the extra energy this arbiter would consume.

References

[1] A. Janek, *et al.*, "Power Management Strategies for Battery-driven Higher Class UHF RFID Tags Supported by Energy Harvesting Devices," in *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, 2007, pp. 122-127.

[2] G. K. Yeap and F. N. Najm, *Low power VLSI design and technology*. Singapore ; River Edge, N.J.: World Scientific, 1996.

[3] N. Linwei, "Overall energy reduction for hard real-time systems based on threshold work-demand analysis," in *Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference on*, 2010, pp. 295-300.

[4] D. C. Hoang, *et al.*, "Thermal energy harvesting from human warmth for wireless body area network in medical healthcare system," in *Power Electronics and Drive Systems, 2009. PEDS 2009. International Conference on*, 2009, pp. 1277-1282.

[5] "Harvest for the world [energy harvesting techniques]," *Power Engineer,* vol. 20, pp. 34-37, 2006.

[6] G. C. Buttazzo, *Hard real-time computing systems : predictable scheduling algorithms and applications*, 2nd ed. New York: Springer, 2005.

[7] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*, 60th anniversary / ed. Princeton, N.J. ; Woodstock: Princeton University Press, 2007.

[8] E. Altman, "Applications of dynamic games in queues," *Advances in Dynamic Games,* vol. 7, pp. 309-342, 2005.

[9] K. G. Binmore, *Fun and games : a text on game theory*. Lexington, Mass.: D.C. Heath, 1992.

[10] R. Gillman and D. Housman, *Models of conflict and cooperation*. Providence, R.I.: American Mathematical Society, 2009.

[11] F. Xia, *et al.*, "Towards Power Elastic Systems Through Concurrency Management," *Newcastle University School of EECE Technical Report,* 2010.

[12] B. Foo and M. van der Schaar, "A queuing theoretic approach to processor power adaptation for video decoding systems," *IEEE Transactions on Signal Processing,* vol. 56, pp. 378-392, 2008.

[13] D. Puschini, *et al.*, "Adaptive energy-aware latency-constrained DVFS policy for MPSoC," 2009, pp. 89-92.

[14] H. Kopetz, *Real-time systems : design principles for distributed embedded applications*. Boston: Kluwer Academic Publishers, 1997.