HAL
open science

# Effectiveness of Power Strategies for Video Applications: A Practical Study

Sébastien Bilavarn, J.J. Khan, Cécile Belleudy, K. Bhatti

## To cite this version:

Sébastien Bilavarn, J.J. Khan, Cécile Belleudy, K. Bhatti. Effectiveness of Power Strategies for Video Applications: A Practical Study. Journal of Real-Time Image Processing, 2014, pp.1. 10.1007/s11554-013-0394-6 . hal-00961508v1

**HAL Id: hal-00961508**

**https://hal.science/hal-00961508v1**

Submitted on 28 Feb 2015 (v1), last revised 11 Jan 2017 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Journal of Real-Time Image Processing manuscript No.**
(will be inserted by the editor)

**Sébastien Bilavarn · Jabran Khan · Cécile Belleudy · Muhammad Khurram Bhatti**

# Effectiveness of Power Strategies for Video Applications: A Practical Study

Received: 5 July 2013 / Revised: 06 November 2013

**Abstract** This study examines the practical effectiveness of power strategies for video applications. Based on real implementations of three power strategies using representative platforms and H.264 applications, we analyse platform and application level parameters affecting the operability and efficiency of power strategies. Results show that, in the same conditions, a strategy might offer highly variable results and sometimes increases energy, depending on the characteristics of the platform. Therefore, we report different measurement results which lead to useful guidelines for succesful power management and show the potential benefits of advanced power strategies over currently available approaches for demanding workloads like video applications.

## 1 Introduction

In recent years, there has been a rapid and widespread growth of non traditional computing platforms such as wireless handheld computing devices. Multimedia services, video broadcasting and streaming represent heavy workloads for these systems that critically affects their battery automomy. Ensuring the best possible use of the limited energy budget is thus essential for this type of application and requires the use of advanced dynamic energy aware processing.

S. Bilavarn
LEAT, CNRS UMR7248, University of Nice Sophia Antipolis
Tel.: +33(0)4 92 38 85 16
Fax: +33(0)4 92 38 85 02
E-mail: bilavarn@unice.fr

J. Khan
LEAT, CNRS UMR7248, University of Nice Sophia Antipolis

C. Belleudy
LEAT, CNRS UMR7248, University of Nice Sophia Antipolis

M.K. Bhatti
COMSATS Institute of Information Technology, CIIT, Lahore

Two main techniques are generally employed to let a computing system adapt its power states. The first one relies on powering down unused resources (Dynamic Power Switching) and the second one, on reducing processor speed (Dynamic Voltage and Frequency Scaling). The decision of which power state to use and when is under the control of a dynamic policy (or strategy) which is therefore determining the actual energy efficiency. Defining advanced power strategies has been an active topic of research for the last couple of decades, however they also come with strong development cost and complexity that are probably responsible for their relative lack of practical implementation and adoption.

In this study, we investigate the actual efficacy of fully implemented dedicated power strategies, using representative up to date platforms and H.264 based video coding applications. DVFS and DPS energy gains are measured on a real development board (Platform Baseboard ARM1176JZF-S) and virtual SystemC cycle acurate MPSoC platforms (QEMU processor emulator including models of ARM11 and Cortex A9). Detailed analysis of results provide reliable numbers on expectable savings as well as useful conditions, guidelines and perspectives for the pratical effectiveness of power strategies, especially for demanding workloads like video processing.

The outline of the paper is the following. First, we review existing efforts in the field of power management and introduce the context and objectives of this work. Then we present detailed experimental results and analysis of three advanced power strategies applied over applications related to the H.264/AVC coding standard. In section 5, we provide a global discussion of the results in terms of conditions for power management operability and effectiveness. Finally, we present a conclusion and perspectives from this work.

## 2 Power management challenges

### 2.1 Overview of general purpose strategies

When it comes to power management, DPS and DVFS are two popular techniques that are broadly employed. In early stages, power switching came first and was under the control of the BIOS Advanced Power Management (APM) developped by Intel and Microsoft (Intel and Microsoft 1996). ACPI (Intel, Microsoft, and Toshiba 1992) was later adopted to allow direct power management by the Operating System. ACPI is a hierarchical technique dividing the overall system components into Global G-states, System S-states, Processor P-states, Busses B-states, Links L-states and Devices D-states. In particular, the number of P-states are processor specific, where each state corresponds to a different frequency and power consumption level. In general, power states are defined regardless of the power management technology used, and are controlled by a software power policy. A typical example is OS-directed power management (OSPM) using ACPI, which defines a policy based on different user, application or environmental parameters. The policy manager selects sleep states when there is no workload on the processor, whereas P-states are employed through DVFS when the processor is active to reduce the overall power. In Linux for instance, the CPUFreq infrastructure is used to set a static or dynamic DVFS power policy for the system. In-kernel governors (i.e. strategies) are used and can change the CPU frequency based on different criteria. Each of five possible governors, Performance, Powersave, OnDemand, Conservative and Userspace, has its own unique behavior, purpose, and suitability in terms of workload.

In addition to ACPI or Linux, modern microprocessors come with their own power management hardware and software to handle power consumption. Examples of such processors include Intel SpeedStep (Intel 2004), AMD Cool'n'Quiet (AMD 2002) and PowerNow (AMD 2000), IBM EnergyScale (Broyles et al. 2011) and ARM IEM (ARM 2006). In most cases, existing power management policies rely on the total workload of a system (which is more related to the hardware state), during a certain period to select the right power state. These policies present the advantage of being applicable in all cases (general purpose), but the drawback is a certain level of inefficiency because they lack of advanced and dynamic knowledge of applications. Besides these standards, a lot of academic research has also been carried out on the subject. Surveys have been proposed for example in (Benini et al. 2000) and (Singh and Chinta 2008) from the abundant literature. As some works pointed out recently the benefits of application awareness in power management compared to OS-level and hardware-level schemes (Liu et al. 2008), we focus in the following on dedicated strategies, especially those which can apply to video processing. These fall into two categories: strategies defined specifically for video applications and deadline (or real time) scheduling.

### 2.2 Dedicated strategies

#### *2.2.1 Video specific strategies*

Video applications are challenging because they represent typical workloads that bring CPUs close to their maximum level of power consumption. However, inherent video properties can be exploited to define efficient DVFS and DPS strategies. Indeed large variations of processing complexity are present in actual video standards that can be used to idle a processor or to lower clock frequencies when decoding less complex frames. This is used for example in (Pouwelse et al. 2001) to regulate voltage/frequency for individual frames based on a prediction, half way decoding the frame, of the remainder of the encoded frame. This prediction is based on the complexity ratio observed in the previous frame of the same type (I, P, PB) and results report up to 40% energy gains. In (Choi et al. 2002), a DVFS scheme also uses the frame type and a more refined prediction of the frame decoding time to scale down voltage/frequency while meeting the predicted time. This technique provides up to 50% energy gains for a MPEG decoder on a StrongARM-based platform. A DPS strategy is proposed in (Javaid et al. 2011) for adaptive pipelined MPSoCs executing multimedia applications. This method is able to exploit different power states, based on the application execution history or predictions of the upcoming workload. 40% energy savings are reported for a H.264 video encoder from cycle acurate simulations of a Tensilica processor.

These works demonstrate undeniable interests in considering application knowledge for critical workloads like video processing. However it is worth noting the relative low number of works in the direction of dedicated strategies, especially considering multimedia and video services in the current context of multiprocessor System-on-Chips and mobile computing.

#### *2.2.2 Deadline scheduling*

Energy-efficient deadline scheduling, also referred to as low power scheduling, is part of a more theoretical field of study on real time scheduling that has attracted a lot of attention. These techniques also apply to video applications as frames are processed under typical constraints of 33 or 40ms (for 30 and 25 frames per second). The general principle is to exploit variations of the actual execution time of jobs in real-time applications to dynamically turn off or change the speed of one or several processors, while ensuring all jobs complete by their deadlines. (Yao et al. 1995) is historically the first theoretical study of energy efficient deadline scheduling. They

consider a single processor with speed scaling, assuming continuous and infinite variation of processor speed. The method proposes an offline minimum energy schedule and two online heuristics called AVR (Average Rate) and Optimal Available (OA), with detailed theoretical models and proofs. An extension to manage temperature is proposed in (Bansal et al. 2007) and more realistic models of bounded speed processors have been considered in (Bansal et al. 2008) and (Chan et al. 2009). Approaches combining both speed scaling and sleep state energy-efficient deadline scheduling (Han et al. 2010), or addressing multiprocessor scheduling without migration (Greiner et al. 2010) have also been proposed.

These works have in common to define advanced theoretical and formal approaches which help to build mathematical proofs, that can be used to address scheduling analysis for example. The counterparts to this formalism are important simplifying hardware asumptions, like ignoring state transition latencies or assuming continuous and infinite variation of processor speed. As processor frequencies are discrete and limited in practice, there can be no implementation and verification of energy gains. Another issue is related to the implementation complexity of realtime scheduling that adds to the feasibility and effectiveness issues in these methods.

### 2.3 Work context and objectives

For previous reasons, this study promotes a practical approach based on concrete experimentation of power strategies to quantify the actual energy gains that can be expected and examine the conditions of efficiency, applicability and variation in representative platforms. Three different types of power strategies based on DVFS and DPS are addressed in the particular context of video processing. These include one video specific strategy and two deadline scheduling strategies implemented using a prototyping framework based on Linux userspace scheduling (Chéour et al. 2011).

The platforms used for experiments combines a real Platform Baseboard ARM 1176JZF-S development board including built-in resources for power monitoring, and two QEMU based virtual platforms including models of previous ARM1176JZF-S and for the Cortex A9 based ST-Ericsson Nova A9500 application processor. As QEMU platforms provides SystemC cycle acurate MPSoC simulations supporting power / energy management and estimation (Pétrot et al. 2011) (Gligor and Pétrot 2011), they were used profitably in the absence of equivalent real platforms at the start of this work. In the following, we investigate these strategies using both a functional H.264 decoder for the ARM11 based Platform Baseboard and a task model of an H.264 encoder on virtual platforms, to be able to process QEMU based simulations. To be representative, we considered strategies of different types, but falling under the same category of dedicated strategies. Following previous classification, we address first strategies that are specific to video processing (using one custom DVFS based strategy for video decoding), and then two strategies for real time scheduling (using a DVFS and a DPS deadline scheduling strategy called DSF and AsDPM), as they also apply considering typical 40ms frame processing constraints.

## 3 Analysis of a video specific strategy

### 3.1 A DVFS strategy for video decoding

#### *3.1.1 Strategy principle*

Previous works on a H.264 decoder reported that the frame rate was sensitive to the motion properties in usual video sequences, with variations of about +/-20% (Bilavarn et al. 2008). The principle of the power strategy is thus to exploit these variations to smooth the frame rate using DVFS.

The processor might run slower when the frame rate is above the average frame rate to reduce power and energy. This adaptation is based on a frame rate constraint to satisfy (e.g. 9fps) which defines distinct operating zones in terms of *fps* for each processor frequency. The tresholds delimiting these zones are determined by the following expression: $tresh_i = adaptation_const * f_{nom}/f_i$, where $f_{nom}$ is the processor nominal frequency. For the ARM1176JZF-S, four thresholds (13.50, 10.05, 9.0 and 8.15fps) can be derived from the four operating points of the processor (160, 215, 240 and 265MHz) for an adaptation constraint of 9fps for example. If the decoder frame rate is between 0.0 and 8.15fps during 250 frames (to minimize frequency switching), the operating point is set to 160MHz, between 8.15 and 9.0, it is set to 215MHz, etc. In the following, we investigate the ability of this policy to save power using an H.264 video profile decoder.

#### *3.1.2 Energy gains on PB ARM1176JZF-S*

A summary of measurement results on a Platform Baseboard ARM1176JZF-S development board is reported in table 1, showing the total energy, mean power, total decoding time, and time per frame for different adaptation constraints. Surprisingly, these results show an increase in energy utilization for adaptation constraints of 4, 8 and 9fps. In each case, the required constraint is under the average performance (11.6fps at nominal frequency of 240MHz), so the CPU frequency actually drops but this does not provide energy savings.

The reason is how downscaling frequency increases the execution time compared to power reduction. As $E = P * T$, the decrease of power is not enough to compensate the increase of execution time on the Versatile

| Adaptation constraint (fps) | Total energy (J) | Average power (mW) | Decoding time (s) | Average time/frame (ms) | fps | Energy gain (%) |
|---|---|---|---|---|---|---|
| not used | 52 | 311 | 168 | 86.4 | 11.6 | 0 |
| 4 | 64 | 262 | 244 | 126.1 | 7.9 | -23 |
| 8 | 62 | 270 | 228 | 117.6 | 8.5 | -19 |
| 9 | 56 | 290 | 196 | 101.4 | 9.9 | -8 |
| 11 | 52 | 311 | 168 | 86.5 | 11.6 | 0 |

**Table 1** Energy gains of DVFS video decoding on ARM1176JZF-S

PB1176JZF-S. This adverse effect is due to the characteristics of the operating points on this platform. For example, moving from 240 to 215MHz raises execution time by 10.4% for a decrease of 6.45% in power (plot labelled ARM1176JZF-S in figure 1). Therefore, the corresponding product $P*T$ remains greater after than before the decrease of frequency. In the next section, we further investigate this by experimenting the same strategy on a virtual platform (QEMU ARM1176) that can let us change the power levels of operating points.

### *3.1.3 Further investigations with QEMU ARM1176*

Figure 1 shows the original characteristics of operating points for the ARM1176JZF-S processor. Three configurations labelled $config_1$, $config_2$ and $config_3$ are added to reflect different levels of load power for each processor frequency. The four configurations are set in a way to homogeneously increase the initial power gap between frequencies. For example switching from 240 to 215MHz implies a reduction of 20, 30, 50 and 90mW respectiveley for $ARM1176$, $config_1$, $config_2$, $config_3$.

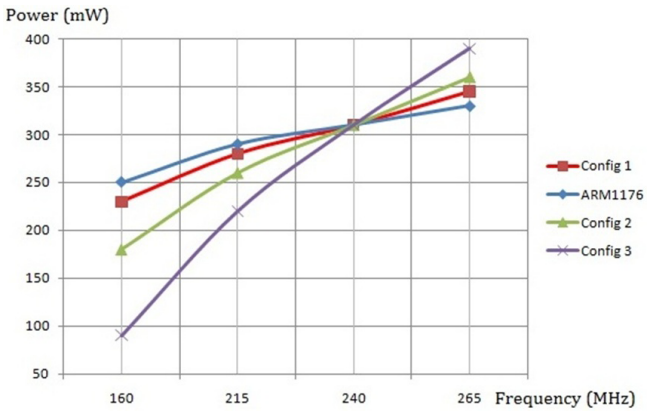


**Fig. 1** QEMU operating point configurations

The energy savings reported with these configurations on the H.264 decoder are shown in figure 2, for three adaptation constraints of 4, 6 and 8fps. Since they are below the original nominal frame rate of 11.6fps, the DVFS strategy actually decreases processor frequency in each case. However, the results indicate that only $config_2$ and $config_3$ succeed at saving energy because for $ARM1176$ and $config_1$, the differences of power consumption in consecutive operating points can not compensate the increase of execution time.

### *3.1.4 Main results and analysis*

These results raise three main points. First, actual platforms may not always provide energy gains when reducing dynamically the frequency due to an inefficiency of the operating points. This is the case for the Versatile Baseboard ARM1176JZF-S used for the experiments, but also on other platforms for various hardware reasons.

The Versatile Baseboard has a development chip that does not operate at real system speed due to prototyping and debugging constraints. The maximum frequency is limited to 265MHz while a production device would be able to run at 800 MHz, and this affects the relevance of operating points. Secondly, the characteristics

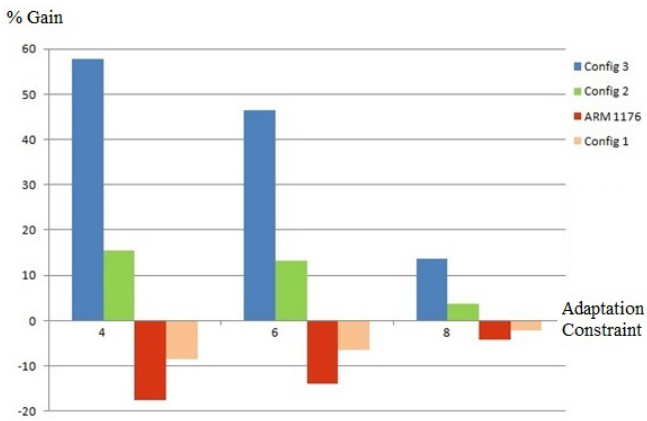


**Fig. 2** Energy gains of DVFS video strategy

of operating points are key parameters in the efficiency of DVFS. In particular, the differences of power levels between consecutive frequencies have a determining impact on the amount of possible energy savings. Under previous platform conditions of efficiency, variations of energy gains due only to the operating point characteristics can reach several orders of magnitude (e.g. 3.6 times between $config_2$ and $config_3$).

Finally, the results also report more than 50% possible energy gains, depending on the application driving parameter of the strategy (frame rate constraint). For

typically high workloads such as video processing, existing workload based approaches would make a CPU to switch at maximium frequency for the duration of the application, resulting in maximum power consumption. This points out the additional benefits of dedicated strategies over existing general purpose strategies for demanding applications.

## 4 Analysis of two deadline scheduling strategies

### 4.1 A DVFS deadline scheduling strategy - DSF

#### *4.1.1 Scheduling principle*

Deterministic Stretch-to-Fit (DSF) is a type of deadline scheduling algorithm intended to reduce power consumption using DVFS (Bhatti et al. 2010). It exploits the fact that the Actual Execution Time (*AET*) of application tasks is actually less or equal to the Worst Case Execution Time (*WCET*). The principle is illustrated in figure 3 where the early completion of a current *Task A* produces time slack *Tslack* to be used by the next priority ready *Task B*. As this allocation provides more time for the execution of *Task B*, the processor speed can be decreased as well as the associated power and energy.

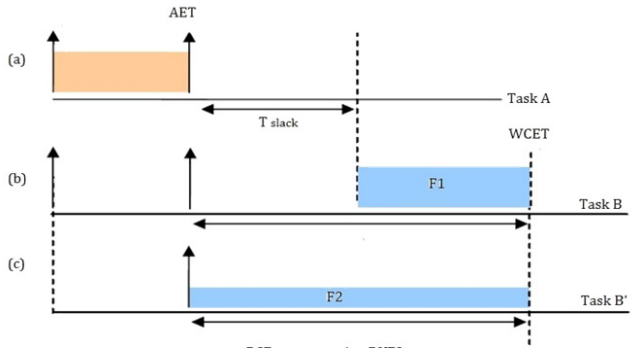


**Fig. 3** DSF scheduling principle

To verify its practical efficiency, DSF was implemented with the afore-mentioned Linux userspace scheduling framework (section 2.3). The application use case is a parallelised H.264 encoder supporting dual core execution which corresponding block diagram is given in figure 4. In this implementation, the motion estimation engine which is the most processing component of the encoder, is split in two parts for acceleration. A task model of the encoder is actually used, composed of the same tasks as that of the real H.264 encoder but doing simplified computing to ease virtual platform simulations. Timing values of the task model (*WCET*, *BCET*, deadline and period) are those of a real encoder profiled on a ST-Ericsson Nova A9500 processor (Dual Cortex A9). Thus varying the slacks from *BCET* to *WCET* for the two motion estimation tasks $T_1$ and $T_2$ let us know the range of DSF achievable gains for ARM1176 and Cortex A9 platforms, that are both in dual core configuration.

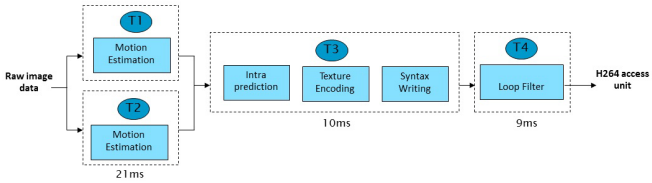


**Fig. 4** H.264 encoder block diagram

#### *4.1.2 Energy gains on QEMU ARM1176 and QEMU Cortex A9*

Figure 5 reports measurement results with energy savings ranging between 5.93% and 16.70% on QEMU ARM 1176, and between 10.21% and 51.46% on QEMU Cortex A9. Cortex A9 outperforms ARM1176 in efficiency by 1.8 times in average, despite the same application (slack) and measurement conditions. As frequency is set upon the slack of a previous task and the *WCET* of the next task, the power levels associated with frequencies account for this difference. Indeed we can check in table 2 that comparatively, operating points have quite different characteristics on both platforms. For example, when the

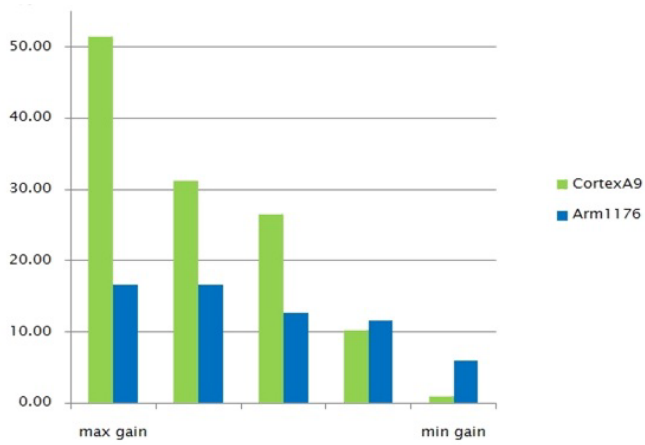


**Fig. 5** DSF energy savings for H.264 encoder

Cortex A9 is downscaled from maximum (1GHz) to minimum (300MHz) frequency, the active (load) power goes 177mW down. The corresponding power gap is 80mW on the ARM1176 processor (from 265MHz to 160MHz). As the gap in power levels between operating points are more important, more power can be saved when decreasing frequency with the Cortex A9 and net energy gains are improved.

Another factor impacting the efficiency of DSF is illustrated in figure 6. In these measurements, we have used a simple example (2 tasks, 1 CPU) to check how frequency switching latencies affects the time granularity of tasks. Timung values of tasks are set by a scaling factor of $10^{-0}$, $10^{-1}$, $10^{-2}$ or $10^{-3}$. It is clear from the

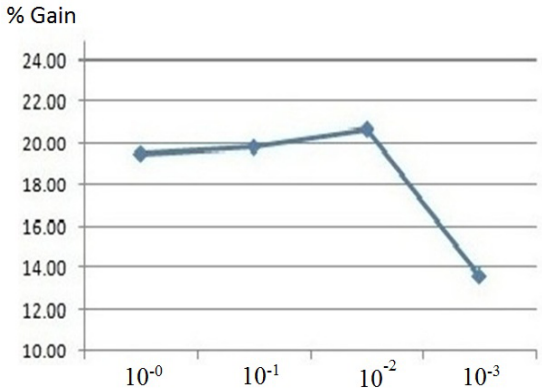


**Fig. 6** Energy gain vs. time granularity of tasks (sec)

results that tasks in the order of milliseconds are rapidly loosing their effectiveness. This comes from the fact that, as they get closer to switching delays (typically around hundreds of microseconds), the energy cost of switching frequency become increasingly sensitive and alters greatly the efficiency of DSF.

#### *4.1.3 Main results and analysis*

These measurements have shown the ability of DSF to save energy for two platforms, with gains that are up to 18% for ARM1176 and 52% for Cortex A9. It emphasizes again the potential of advanced strategies to further reduce energy, whereas a general purpose (workload based) approach would lead to operate at maximum frequency and power for high workloads like video processing. DSF driving parameter is the application dynamic slack. Under the same conditions of slack and measurement, there is a notable facotr of 1.6 in energy efficiency between ARM1176 and Cortex A9. Like for previous DVFS video strategy, the level of energy gains are firmly related to the platform characteristics, namely the differences in power levels between operating points. Finally, the execution time of application tasks must also exceed sensitively the latency of frequency switching delays in order to be applicable. In other words, not all applications are compatible with DVFS if this task requirement is not verified.

### 4.2 A DPS deadline scheduling strategy - AsDPM

#### *4.2.1 Scheduling principle*

Assertive Dynamic Power Management is a strategy intended to be used for multiprocessor low power scheduling (Bhatti et al. 2011). The general principle is based on maximising the execution of application tasks on a minimum number of processors in order to set unused cores into low power sleeping states. The example of figure 7(a) shows the distribution of three tasks on two processors with segmented idle times under a standard EDF schedule. In figure 7(b), the associated AsDPM schedule groups the execution of the tasks on processor $\pi_1$, therefore substantial energy can be saved by putting processor $\pi_2$ into a sleep state for an extended duration. In the following, we examine the effectiveness of this scheduling strategy using the Linux userspace framework mentioned in section 2.3.

#### *4.2.2 Energy savings on QEMU ARM1176 and QEMU Cortex A9 platforms*

Based on its definition, AsDPM scheduling needs a platform with at least two processors to be applicable. The scheduler is implemented on QEMU ARM1176 and QEMU Cortex A9 in a dual core configuration each, and applied to the previous H.264 encoder example. At the beginning of an hyperperiod, the encoder requires two processors to execute parallel motion estimation tasks *T1* and *T2*, while *T3* and *T4* are sequential and only need one processor. The second processor can thus be idled for a period of time that depends on the value of dynamic slacks produced by tasks *T1* and *T2*. We have set six different values of $AET$s for *T1* and *T2* corresponding to the six solutions reported in figure 8 to fully characterize the range of energy gains. Varying the application slack this way also provide reliable comparison of energy gains on both platforms.

Figure 8 shows that AsDPM results in actual energy gains for both platforms, ranging from 24.05% to 46.73% on ARM1176 and 15.32% to 42.72% on Cortex A9. AsDPM is 1.4 times more efficient on ARM1176, whereas DSF was previously better on Cortex A9.

This difference comes again from the characteristics of operating points, but this time in terms of load and idle power levels. Measurements have been carried out at maximum frequency for both platforms, respectively 265MHz and 1GHz for ARM1176 and Cortex A9. It is worth noting from table 2 that idle power of the ARM1176 (252mW) is important compared to load power (330mW), while there is more difference for Cortex A9 (respectively 90mW and 320mW). Therefore putting a core into a nearly zero Watt sleeping state reduces more power on ARM1176 than on Cortex A9. The important idle power on ARM1176 results again from the constraints of an early prototyping platform. However, these

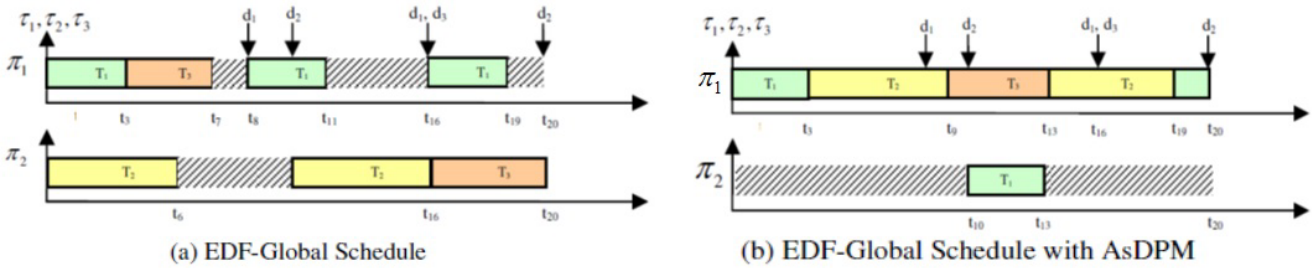


**Fig. 7** AsDPM scheduling principle

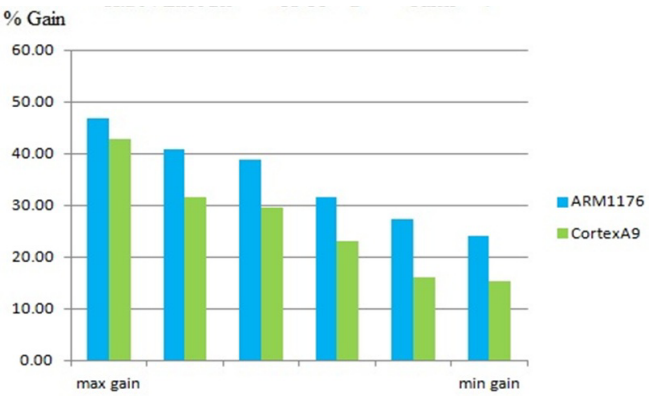


**Fig. 8** AsDPM energy savings for H.264 encoder

| Platform | Freq. (MHz) | Idle P. (mW) | Load P. (mW) |
|---|---|---|---|
| **ARM1176** | 160 | 223 | 250 |
| | 215 | 238 | 290 |
| | 240 | 245 | 310 |
| | 265 | 252 | 330 |
| **Cortex A9** | 300 | 38 | 143 |
| | 600 | 60 | 215 |
| | 1000 | 90 | 320 |

**Table 2** Idle versus load power for QEMU ARM1176 and QEMU Cortex A9

results show the ability of AsDPM to address systems with an important part of idle power consumption.

*4.2.3 Main results and analysis*

Practical effectiveness of AsDPM has been shown on the basis of a scheduler prototype operating on two platforms, with net energy gains up to 46% for ARM1176 and 42% for Cortex A9 depending on the variability of the application driving parameter (dynamic slack). The amount of energy gains is also determined by the idle power levels of the platform which are operating point dependent. AsDPM is especially suitable for systems with a sensitive share of idle power, which is highly related to standby leakage power and typical of deeply scaled nanometer technologies. Like for DVFS strategies, the latencies of entering/resuming from different power states should be negligible against the minimium Best Case Execution Time ($BCET$) of all application tasks. If this condition is not met, the overheads of switching sleep/active states will affect the energy efficiency of the strategy, and also probably lead to improper application execution. As these latencies are usually greater than for changing frequency, AsDPM and more generally DPS based techniques might be less applicable than DVFS especially to cope with typical 40ms frame processing constraints of video applications.

## 5 Effectiveness of power strategies

### 5.1 Platform level conditions

The majority of DVFS power strategies are based on the hypothesis that decreasing processor frequency results in saving energy. Although this fundamental assumption is verified on most platforms, section 3.1 has shown that there are some cases where it is not due to the operating points. Indeed, their characteristics determine the actual saving ability and to a great extent the amount of energy gains. A determining factor in DVFS efficiency is the relative difference of load power levels compared with that of the corresponding frequencies. Large differences are a prerequisite for meaningful energy gains while too small differences might result in practically inefficient DVFS. DPS strategies are subject to similar platform conditions, a prominent criterion in this case is the power levels of idle states. DPS is more effective when operating on high values of idle power, which is typically the case in recent deep sub-micron process technologies with large static power consumption.

The applicability of power strategies also strongly depends on the transition costs between power states. DVFS policies relies implicitly on the asumption that latencies of changing frequency, usually estimated at around a few hundreds of microseconds, can be neglected, i.e., when the application can afford those waiting times. This does not apply for example if these latencies exceed a few milliseconds regarding typical 40ms frame processing constraints of video applications. Latencies of DPS state transitions are often higher, due to complex shutdown, wakeup and context saving schemes. This might

further restrict the use of DPS compared to DVFS, especially in video applications. Therefore, llatencies of state transitions should be analyzed thoroughly against the time granularity of tasks before developing DVFS or DPS strategies.

### 5.2 General purpose vs. domain specific strategies

General purpose workload based approaches (e.g. Linux OnDemand) would typically set the CPU frequency level to its maximum for H.264 coding/decoding, driving an increased power demand. Advanced DVFS and DPS strategies exploit closer application awareness and can therefore perform better. In results of sections 3.1, 4.1 and 4.2, they show an ability to actually reduce the energy use up to 50%. The effectiveness of these application driven power strategies relies on their capacity to exploit a form of execution variability. They are based on a driving parameter (e.g. frame rate, dynamic slack) which variations at runtime determine the actual amount of energy gains. A reasonable quantity of energy reduction that can be expected is situated between zero and a maximum of 50%, based on data-dependent execution. This effectiveness is also subject to the afore-mentioned conditions on time definition of tasks that must be large enough to be able to neglect the cost of processor power state transitions.

### 5.3 DVFS versus DPS

A question that often arises is which of DVFS or DPS is more energy efficient. We can provide here a reliable comparison of two related strategies from the results of DSF and AsDPM experimentations on the common H.264 encoder example.

Energy gains have been reported previously in figures 5 and 8 respectively for DSF and AsDPM on both ARM1176 and Cortex A9. They show that AsDPM outperforms DSF energy gains in all cases. However they also indicate that the difference in efficiency might vary greatly from a platform to another. As stated previously, the influencing factors are the power level gaps of operating points for DVFS, and the idle versus load power levels for DPS strategies. Though it is likely that powering down unused processors (DPS) has potential for higher gains than reducing their speed (DVFS), generalization has to be considered with caution. Indeed the actual characteristics of the platform (operating points, power levels and state transition latencies) can reverse the situation in some cases, or prevent proper application of a strategy. DPS should be more efficient, but more limited in terms of applicability, especially to cope with frame rate processing constraints of video applications.

## 6 Conclusion and perspectives

This paper described an analysis of power strategies for video applications based on actual implementations and fine measurements. Three types of power strategies have been investigated with H.264 applications on representative ARM based platforms, in a way to deal reastically with power strategies and study the conditions influencing high energy gains. The outcome of this experimental study are a better understanding of practical constraints affecting power management efficiency and opportunities for improvement. Actual energy savings have been shown to be strongly dependent on platform conditions, that are the levels and transition delays associated with power states for both DVFS and DPS based strategies. This underlines the importance of a first and foremost analysis of hardware characteristics to define strategies that are successful in practice. If these conditions are met, the efficiency of strategies ultimately depend on their scope and relevance too. The ability of more dedicated policies to exploit fine application knowledge gives room for further energy gains, up to 50% according to our measures for video applications where general purpose strategies would be unsuited and inefficient.

These results suggest that higher levels of energy gains can be reached using a joint contribution of strategies. As an important concern in the improvement of energy cost is to make an optimal use of power management features, power management can build on finer workload awareness using for example general purpose policies for most applications and dedicated (DVFS and/or DPS based) strategies for power sensitive workloads. We have already started to investigate a cooperation of DSF and AsDPM strategies in this context (Bhatti et al. 2011).

While very promising, these perpectives let us expect greater complexity. The near-term propects of integration technology (below 22nm FinFET, 3D stacking) and the power constraints associated (power density, thermal management, heterogeneity) will also add significantly to this complexity. In the face of these perspectives, it is quite likeky that existing power management solutions, already very complex, reach their limit. More advanced dynamic energy-aware supervision will probably have to be redefined, we have started to investigate this addressing runtime analysis for multiobjective optimization.

**Acknowledgements** This work was carried out under the COMCAS project (CA501), a project labeled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

## References

Intel and Microsoft (1996) Advanced Configuration and Power Interface Specification, Revision 1.0, December 1996

Intel, Microsoft, and Toshiba (1992) Advanced Power Management (APM) : BIOS Interface Specification, Revision 1.0, January 1992

Intel (2004) Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor, March 2004

AMD (2002) AMD Cool'n'Quiet Technology, http://www.amd.com/us/products/technologies/cool-n-quiet/Pages/cool-n-quiet.aspx

AMD (2000) AMD PowerNow! Technology, November 2000

Intel Corporation (2004) Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor, March 2004

Martha Broyles, Chris Francois, Andrew Geissler, Gordon Grout, Michael Hollinger, Todd Rosedahl, Guillermo J. Silva, Mark Vanderwiel, Jeff Van Heuklon and Brian Veale (2011) IBM EnergyScale for POWER7 Processor-Based Systems, IBM journal of Research and Development, 55, 3, May 2011, pp 220–232

ARM (2006) Intelligent Energy Manager (IEM) Hardware Control System in the ARM1176JZF-S Development Chip, November 2006

Luca Benini, Alessandro Bogliolo and Giovanni De Micheli (2000) A survey of design techniques for system-level dynamic power management, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8, 3, 2000, pp 299–316

Pushkar Singh and Vinay Chinta (2008) Survey Report on dynamic Power Management, Survey report of University of Illinois, Chicago (ECE Department), Chicago, USA 2008.

Xiaotao Liu, Prashant Shenoy, and Mark Corner (2008) Chameleon: Application Level Power Management with Performance Isolation, IEEE Transactions on Mobile Computing, vol. 7, no. 8, 2008, pp. 995-1010

Johan Pouwelse, Koen Langendoen, Inald Lagendijk, Henk Sips (2001) Power-Aware Video Decoding, 22nd Picture Coding Symposium, Seoul, Korea

Kihwan Choi, Karthik Dantu, Wei-Chung Cheng and Massoud Pedram (2002) Frame-Based Dynamic Voltage and Frequency Scaling for a MPEG Decoder, Proceedings of the International Conference on Computer-Aided Design (ICCAD'02), 2002, pp 732–737.

Haris Javaid, Muhammad Shafique, Jörg Henkel, and Sri Parameswaran (2011) System-Level Application-Aware Dynamic Power Management in Adaptive Pipelined MPSoCs for Multimedia, Proceedings of the International Conference on Computer-Aided Design (ICCAD'11), 2011, San Jose, pp 616–623.

F. Yao, A. Demers, and S. Shenker (1995), A scheduling model for reduced CPU energy, Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS), 1995, pp. 374–382.

N. Bansal, T. Kimbrel, and K. Pruhs, Speed scaling to manage energy and temperature, Journal of the ACM (2007), 54.

N. Bansal, H.L. Chan, T.W. Lam, and L.K. Lee, Scheduling for speed bounded processors, Proceedings of International Colloquium on Automata, Languages and Programming (ICALP), (2008), pp. 409–420.

H.L. Chan, W.T. Chan, T.W. Lam, L.K. Lee, K.S. Mak, and P.W.H. Wong, Optimizing throughput and energy in online deadline scheduling, ACM Transactions on Algorithms, 6 (1) (2009).

G. Greiner, T. Nonner, and A. Souza, The bell is ringing in speed-scaled multiprocessor scheduling, Proceedings of ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), (2010), pp. 11–18.

Xin Han, Tak Wah Lam, Lap-Kei Lee, Isaac Kar-Keung To, and Prudence W. H. Wong, Deadline scheduling and power management for speed bounded processors, Journal of Theoretical Computer Science, (2010), pp. 3587–3600.

Rym Chéour, Sébastien Bilavarn and Mohamed Abid (2011) Exploitation of EDF Scheduling in Wireless Sensor Networks, International Journal on Measurement Technologies and Instrumentation Engineering (IJMTIE), 1, 2, 2011, pp 14–27

Frédéric Pétrot, Nicolas Fournel, Patrice Gerin, Marius Gligor, Mian-Muhammed Hamayun and Hao Shen (2011) On MPSoC Software Execution at the Transaction Level, IEEE Design & Test, 28, 3, May 2011

Marius Gligor and Frédéric Pétrot (2011) Handling dynamic frequency changes in statically scheduled cycle-accurate simulation, Proceedings of the 16th Asia and South Pacific Design Automation Conference (ASPDAC'11), Jan. 2011

Sébastien Bilavarn, Cécile Belleudy, Michel Auguin, Thibault Dupont and Anne-Marie Fouilliart (2008) Embedded Multicore Implementation of a H.264 Decoder with Power Management Considerations, Proceedings of the 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools (DSD'08), 50, 1, Sept. 2008, pp 124–130

Khurram Bhatti, Cécile Belleudy and Michel Auguin (2010) An Inter-Task Real Time DVFS Scheme for Multiprocessor Embedded Systems, Proceedings of International Conference on Design and Architectures for Signal and Image Processing (DASIP'10), Oct. 2010, Edinburgh, UK

Khurram Bhatti, Cécile Belleudy and Michel Auguin (2011) Hybrid Power Management in Real Time Embedded Systems: An Interplay of DVFS and DPM Techniques, International Journal of Time-Critical Computing - Real Time Systems, 47, 2, March 2011, pp 143–162