Politecnico di Torino

Energy management for IoT
01UDGOV

Master's Degree in Computer Engineering

Lab 01 - DPM
Group 08

Github repository: `https://github.com/akhre/EnergyManagementForIoT/tree/main`

Candidates:
Diego Porto (s313169)
Mohamed elsayed Mahmoud elsayed elsisy (s330451)

Referee:
Prof. Massimo Poncino
Dr. Matteo Risso

Contents

Abstract

In case transitions are costless in terms of time and power, the DPM policy is trivial because the idea is to stop a component as soon as it goes idle.

But in real cases, as in the one depicted in our PSM, the transitions costs in terms of time and power are not zero and for this reason they must be amortized.

Knowing this, the DPM policy based on timeout shutdown only when idleness is long enough to make the transition convenient. In doing this, mispredictions overhead must be taken into account.

CHAPTER 1

Assignment 1 - Part one

In this section, we will execute the simulator using two distinct workloads derived from sensor responses. The simulation will adhere to the default timeout policy. Transition constraints are configured to permit state changes only between RUN and IDLE states (no other state transitions are allowed).The Power State Machine (PSM) file used for this experiment is located at dpm-simulator/example/psm.txt.

1.1 Workload analysis - impact on DPM

The differences in the sensor's response times between `workload_1.txt` and `workload_2.txt` can significantly impact the effectiveness and behavior of the DPM policy. For example:

Fast Sensors (`workload_1.txt`): With a response time of 4ms, it is better to the system to stay in RUN state compared to transition to IDLE, due to the overhead in the transitions. But, in case the timeout is between 0 to 1 it may be better to move to idle since there is saving

Slow Sensors (`workload_2.txt`): With a response time of 100ms, it could be convenient to the system to switch to IDLE in the meantime as the overhead caused by the transition may still be less than staying in the RUN state.

Shorter timeout values may be more effective in managing power consumption, as the system can quickly transition to low-power states when the sensors are inactive. However, we must also avoid frequent transitions, which can be counterproductive due to the high transition overhead.

1.2 Workload 1 - fast sensor response's

This workload uses "fast" sensors so the value is returned in 4 ms. The file has two values in each line: the first value is the arrival time of the task and the second value stands for the duration of that task. The following timeout values are considered: 0ms ,1ms, 4ms, 20ms, 40ms and 60ms.
The command to run the first workload with a timeout of 1ms is:

```
./dpm_simulator -t 1 -psm example/psm.txt -wl \\
../workloads/workload_1.txt > results/workload_1/workload_1ms
```

The results of the simulation are stored in the **results** folder, divided by workload. In the output, the [**sim**] section provides detailed results of the simulation:

- Active and Inactive Time: The total time the system spent in active and inactive states.
- Total Time: The total simulation time, both with and without DPM.

- State Times: The total time spent in each state (Run, Idle).
- Timeout Waiting Time: The time spent waiting for the timeout to expire.
- Transitions Time: The total time spent in state transitions.
- Number of Transitions: The total number of state transitions.
- Energy for Transitions: The total energy consumed during state transitions.
- Total Energy: The total energy consumption with and without DPM.

For the first workload, we can observe for different timeouts the following values 1.1:

Timeout (ms)	Number of transitions	Total energy with DPM (J)	Timeout waiting time (s)
0ms	170	0.6979953200J J	7.922800 s
1ms	138	0.6997882530 J	0.086400 s
4ms	18	0.7032023760J	0.256100s
20ms	18	0.7070946960J	0.400100s
40ms	18	0.7119600960J	0.580100s
60 ms	18	0.7168254960J	0.760100s

Table 1.1: Output for workload 1

As shown, different timeout values affect the number of state transitions, total energy consumption, and the time spent in each state. Shorter timeouts lead to more frequent transitions and overall lower energy consumption, while longer timeouts result in fewer transitions and higher energy consumption. This occurs because the overhead caused by the transition between the RUN and IDLE states has a lower impact on energy consumption compared to staying in the RUN state during that time.

From those results, we observed that lower timeout values lead to more frequent state transitions, resulting in higher transition time values and energy consumption. However, they also lead to lower overall energy consumption.

On the other hand, higher timeout values result in fewer transitions, which means lower transition overhead consumptions, but they also lead to higher overall energy consumption.

Given these results, for the first workload (workload_1, fast sensors), despite the high number of transitions, it is better to keep the timeout threshold at lower values, such as 1ms.

1.3 Workload 2 - slow sensor response's

This workload uses sensors that return a value after 100 ms. The file **workload_2.txt** contains two values on each line: the first value is the arrival time of the task, the second one represents the duration of that task.

Due to the longer response time of the sensors, the timeout values will differ from those used in Workload 1. The choosen timeout values are:0ms, 80ms, 100ms, 120ms, 140ms, 160ms, 180ms, and 200ms.

The results in 2.4 suggest that as the timeout value increases, the total energy consumption with DPM slightly increases. For instance, the energy consumption with DPM is 0.9073J at 80ms and increases to 0.9769J at 200ms. This indicates that shorter timeout values are more energy-efficient.

The number of state transitions decreases significantly as the timeout value increases. For example, there are 194 transitions at 80ms compared to only 20 transitions at 200ms. While fewer transitions can reduce the overhead and energy consumption associated with state changes, it is still beneficial to switch to the IDLE state despite the transition overhead.

Timeout	# of transitions	Total energy with DPM	TO waiting time	Time in RUN	Time in IDLE
0ms	198	0.3685548779 J	0.009900s	10.469800s	1088.938200s
80ms	194	0.9073267669 J	7.922800s	10.469800s	1081.439100s
100ms	56	0.9464934979 J	9.421700s	11.968700s	1079.995400s
120ms	20	0.9552641029 J	9.759200s	12.306200s	1079.671900s
140ms	20	0.9606701029 J	9.959200s	12.506200s	1079.471900s
160ms	20	0.9660761029 J	10.159200s	12.706200s	1079.271900s
180ms	20	0.9714821029 J	10.359200s	12.906200s	1079.071900s
200ms	20	0.9768881029 J	10.559200s	13.106200s	1078.871900s

Table 1.2: Output for workload 2

As expected, the timeout waiting time increases with higher timeout values, indeed higher timeout values lead to longer periods of inactivity before transitioning to a low-power state.

CHAPTER 2

Assignment 2 - Part 2

In this chapter, the transition between RUN and SLEEP state is implemented. To modify the timeout policy to enable transitions to SLEEP, the following steps need to be taken:

1. Modify the **dpm_decide_state** function inside **dpm_policies.c**: add the sleep state defined in **psm.h** file.
2. Modify **dpm_timeout_params** struct inside **dpm_policies.h**. The struct was updated to create an array of two double numbers timeout[2], which represents the timeouts for both RUN and SLEEP states.
3. Update the struct inside **utilities.h** to reflect the changes made inside dpm_policies.h

The results of the new simulations are saved inside the **results/workload_2** folder. The primary focus will be on energy consumption and the time spent in different states (RUN, IDLE, SLEEP).

2.1 Workload 1

Timeout	# of transitions	Total energy with DPM	TO waiting time	Time in SLEEP
0ms	82	0.2607854300J J	7.906200s	1079.443000s
1ms	76	0.2559152120J J	0.047300s	1079.414800s
4ms	18	0.2022010260J J	0.223700s	1079.383400s
20ms	18	0.2061624660J J	0.367700s	1079.239400s
40ms	18	0.2111142660J J	0.547700s	1079.059400s
60ms	18	0.2160660660J J	: 0.727700s	1078.879400s

Table 2.1: Output for workload 2

The simulations with the "sleep" parameter (timeout = 4ms) show significantly lower total energy consumption with DPM compared to their counterparts without the "sleep" parameter. Similarly, the simulations with the "IDLE" parameter (timeout = 0ms) show significantly lower total energy consumption with DPM compared to their counterparts without the "IDLE" parameter. Therefore, the best timeout value for Sleep mode with respect to workload_1 is 4ms, which is particularly convenient given that the average response time is 4ms for fast sensor responses. In conclusion, the optimal configuration is either:

- 4ms Sleep / 4ms Idle
- 4ms Sleep / 0ms Idle

	Timeout 0 ms IDLE	Timeout 0 ms SLEEP
Total time in states	1079.576000s idle, no sleep	1079.443000s in sleep, no idle
Timeout Waiting Time	7.922800s	7.906200s
Transitions Time	0.068000s	0.205000s
Number of Transitions	170	82
Energy for Transitions	0.0017000000J	0.0828200000J
Total Energy with DPM	0.6979953200J	0.2607854300J

Table 2.2: Timeout 0 ms: IDLE vs SLEEP

	Timeout 4 ms IDLE	Timeout 4 ms SLEEP
Total time in states	1079.388800s idle, no sleep	1079.383400s in sleep, no idle
Timeout Waiting Time	: 0.256100s	0.223700s
Transitions Time	0.007200s	0.045000s
Number of Transitions	18	18
Energy for Transitions	0.0001800000J	0.0181800000J
Total Energy with DPM	0.7032023760J	0.2022010260J

Table 2.3: Timeout 4 ms: IDLE vs SLEEP

2.2 Workload 2

Timeout	# of transitions	Total energy with DPM	TO waiting time	Time in SLEEP
0ms	198	0.3685548779J J	0.009900s	1088.938200s
80ms	184	0.5716452409J J	7.906200s	1081.076900s
100 ms	56	0.4822010109J J	9.353200s	1079.949900s

Table 2.4: Output for workload 2

	Timeout 0 ms IDLE	Timeout 0 ms SLEEP
Total time in states	1089.350400s idle, no sleep	1088.938200s in sleep, no idle
Timeout Waiting Time	0.009900s	0.009900s
Transitions Time	0.079200s	: 0.495000s
Number of Transitions	198	198
Energy for Transitions	0.0019800000J	0.1999800000J
Total Energy with DPM	0.6934801680J	0.3685548779J

Table 2.5: Timeout 0 ms: IDLE vs SLEEP

The simulations with the “sleep” parameter (timeout = 0ms_sleep) show significantly lower total energy consumption with DPM compared to those without the “sleep” parameter. Similarly, the simulations with the “IDLE” parameter (timeout = 0ms_IDLE) also show significantly.

2.3 Three PSM STATE MACHINE

In this three-state PSM transitions can move from IDLE to RUN to SLEEP, or from SLEEP to RUN to IDLE, with flexibility in the state transitions. Additionally, the system allows transitions from RUN to any state, subject to constraints. The code edits are designed to handle timeouts for both

the SLEEP and IDLE states. Specifically, tparams.timeout[1] corresponds to the SLEEP timeout, while tparams.timeout[0] corresponds to the IDLE timeout. The transition logic works as follows: if the SLEEP timeout is greater than the IDLE timeout, the system will prioritize the SLEEP state. Otherwise, it will transition to the IDLE state. During the workload, the PSM process will adjust based on the system's idle duration. This means the system will alternate between some time spent in IDLE and some time in SLEEP, optimizing energy savings. The state transitions will follow the constraints described in the report, ensuring the best energy-saving configuration while respecting the transition conditions.

2.3.1 Examples

For example, when the best value for SLEEP is 4ms and IDLE is either 4ms or 0ms for workload_1, and when the values for the IDLE timeout are set to 0ms and the SLEEP timeout to 4ms in the three-state system, the result will be as follows:

- Energy w DPM = 0.1976708950J
- Energy for transitions = 0.0195400000J
- N. of transitions = 154
- Transitions time = 0.099400s
- Timeout waiting time = 0.006800s
- Total time in state Sleep = 1079.376200s
- Total time in state Idle = 0.170100s
- Total time in state Run = 2.930800s

As shown, the resulting energy value will be lower when the system is in SLEEP mode during idle time with a 4ms timeout, which saves energy compared to other configurations. For instance, with timeout_4ms_sleep, the energy consumption is 0.2022010260J, whereas setting the IDLE timeout to 0ms results in energy consumption of 0.6979953200J.

The second example is for workload_2, where both the SLEEP and IDLE timeouts are set to 0ms. When these values are passed to the three-state machine, the result is that the PSM is always in SLEEP mode. However, the outcome is either good or at least comparable to the SLEEP mode. In fact, the result of using 0/0 timeouts will be equivalent to setting the system to SLEEP mode for workload_2.

CHAPTER 3

Assignment 1 - Part 3

3.1 Predictive Policy

The History Policy is a predictive policy that estimates the transition time based on the previous workload pattern. The computed value, Tpred, is compared to the thresholds for IDLE and SLEEP. After making a decision, the system can independently transition to one of the low-power states. In the code section related to DPM history in the dpm_policies.c file, Tpred is calculated using one of two possible methods: the polynomial method and the regression method. Both methods can be activated using #ifdef macros in the header section of dpm_policies.c. Once the time prediction is calculated, the next step is to decide whether the system should transition to SLEEP or IDLE. This decision is made by evaluating specific conditions. For example, to transition to SLEEP, the predicted time must be higher than the time break-even and the sleep threshold, and there must be higher priority for SLEEP than for IDLE. If these conditions are met, the system will transition to SLEEP. If not, the system will transition to IDLE, provided that the time prediction is higher than both the break-even time for IDLE and the IDLE threshold. If the prediction fails to meet the conditions for either SLEEP or IDLE, the system will transition to RUN. This ensures that the transition is either from SLEEP to RUN, IDLE to RUN, or vice versa, as necessary. The code has been edited to allow transitions between the three states, based on these conditions. The code works in two state transition as well.

3.1.1 Examples and results

With workload_1, the best threshold timeout policy was 4ms for SLEEP mode and 0ms for IDLE. When combining 4ms for SLEEP mode with 0ms for IDLE in the three-state timeout policy, the resulting energy consumption was 0.1976708950J.

In the case of the History Policy using the polynomial method, the following coefficients were used:

- k1 = 0.9
- k2 = 0.80
- k3 = 0.70
- k4 = 0.60
- k5 = 0.50

3.1.2 Example 1:

With a threshold of 4ms for SLEEP and 0ms for IDLE, the results are as follows:

- Energy w DPM = 0.2587870080J
- Energy for transitions = 0.0808200000J
- N. of transitions = 82
- Transitions time = 0.200800s
- Timeout waiting time = 0.004100s
- Total time in state Sleep = 1079.444700s
- Total time in state Idle = 0.002500s
- Total time in state Run = 2.928100s

The result is generally quite good and close to the result of the three-state timeout policy. Changes in the coefficients lead to either more energy savings or an increase in energy consumption for DPM. However, based on practical examples, in the case of 4ms for sleep and 0ms for idle, the energy consumption typically ranges from 0.25J to 0.27J.

3.1.3 Example 2:

Since the predictive three-state machine can also function as a binary machine—transitioning between RUN and IDLE, or SLEEP and RUN—with a threshold of 1ms for the IDLE period, the system can operate under predictive history DPM. If the SLEEP threshold is ignored (disabled, set to 0), the machine will only trigger between IDLE and RUN.

Using the following coefficients:

- $k_1 = 0.9$
- $k_2 = 0.80$
- $k_3 = 0.70$
- $k_4 = 0.60$
- $k_5 = 0.50$

The following energy values were observed:

- Predictive_1ms_idle = 0.6980541630J
- Predictive_1ms_sleep = 0.2587870080J
- Timeout_1ms = 0.6997882530J
- Timeout_1ms_sleep = 0.2559152120J

For workload 1, the results of the predictive model are close to the timeout policy. The predictive model performs better in terms of the IDLE threshold compared to the timeout idle policy, but it doesn't perform as well as the timeout sleep policy, although the results are still close.

3.1.4 Example 3:

For workload 2, using the same coefficients, the results for the IDLE and SLEEP thresholds are better than using the timeout policy for either IDLE or SLEEP.

The following energy values were observed:

- Predictive_1ms_idle = 0.7203257410J
- Predictive_1ms_sleep = 0.3670321360J
- Timeout_1ms = 0.9073267669J
- Timeout_1ms_sleep = 0.5716452409J

3.1.5 Example 4:

The following energy values were observed:

- Predictive_1ms_idle = 0.7203257410J
- Predictive_1ms_sleep = 0.3670321360J
- Timeout_1ms = 0.9464934979J
- Timeout_1ms_sleep = 0.4822010109J