Characterizing Display QoS Based on Frame Dropping for Power Management of Interactive Applications on Smartphones

Kuan-Ting Ho, Chung-Ta King, Bhaskar Das
Department of Computer Science
National Tsing Hua University, Hsinchu, Taiwan

Yung-Ju Chang
Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan

***Abstract*—User-centric power management in smartphones aims to conserve power without affecting user's perceived quality of experience. Most existing works focus on periodically updated applications such as games and video players and use a fixed frame rate, measured in frame per second (FPS), as the metric to quantify the display quality of service (QoS). The idea is to adjust the CPU/GPU frequency just enough to maintain the frame rate at a user satisfactory level. However, when applied to aperiodically-updated interactive applications, e.g. Facebook or Instagram, that draw the frame buffer at a varying rate in response to user inputs, such a power management strategy becomes too conservative. Based on real user experiments, we observe that users can tolerate a certain percentage of frame drops when running aperiodically updated applications without affecting their perceived display quality. Hence, we introduce a new metric to characterize display quality of service, called the frame drawn ratio (FDR), and propose a new CPU/GPU frequency governor based on the FDR metric. The experiments by real users show that the proposed governor can conserve 17.2% power in average when compared to the default governor, while maintaining the same or even better QoE rating.**

***Index Terms*—Power management, user experience, CPU/GPU frequency scaling, smartphones, frame rate**

I. Introduction

User-centric power management for smartphones, which takes user experiences into account while trading off system performance and power, has attracted much attention recently [1]–[7]. Majority of those works focus on the display quality and use *frame rate*, which is measured in *frame per second* (FPS), as the metric for user quality of experiences (QoE). Most previous works on user QoE-aware power management considered *periodically updated applications* [4], which continuously draw and update the frame buffer at fixed rate, such as gaming and video streaming applications. However, many applications, such as Facebook and Instagram, are *aperiodically updated* [4] which render the frames in response to user inputs. The target QoE is no longer a fixed frame rate. On the other hand, if we observe user behaviors more closely, we can see that most users can tolerate some frame drops without affecting their perception of the visual quality, when performing activities such as scrolling or clicking. Thus, CPU/GPU frequency scheduling should consider not only how many frames are to be drawn, but also how many may be dropped. We propose in this paper a new QoE metric for power management on smartphones for interactive applications, which is called *Frame Drawn Ratio* (FDR). A user satisfaction-aware dynamic frequency governor, called *FDR CPU-GPU governor*, for interactive applications is also presented. The governor takes account of heterogeneous multicores. The proposed governor was implemented and real users were asked to use the smartphone and evaluate the new governor. Experimental results show that comparing to the default governor on Android, the FDR governor can improve power consumption by 17.13% on average, while maintaining the same or even better QoE rating. We have conducted an experiment with 14 real users for two weeks and asked their experiences in using Facebook and Instagram on smartphones under varying frame drops. We have developed a data collection service on Android 6.0.1 which would randomly choose and fix a CPU/GPU frequency to run the applications and record the amount frames drawn and dropped per unit time. When the user left the application, our service would pop up a questionnaire randomly to ask for user satisfaction, which is divided into five scales: "Excellent", "Good", "Fair", "Poor", and "Bad". Participants normally needed to fill out 8 questionnaires per day in average, with 1 to 3 questionnaires per hour. The experiment was designed based on a method called Experience Sampling Method (ESM) outlined in [8]–[10]. Table I shows the results of our experiment. The fractions indicate the average fractions of frames that were dropped per unit time when the user gave the corresponding satisfaction score. The number inside the parentheses show the amount of questionnaires generating that fraction. From the table, one can observe that perceived display quality is very user-dependent, but in general all users can tolerate a certain percentage of frame dropping without noticing a degradation in display quality. Based on these observations, we have proposed a novel QoS metric for power management on smartphones for aperiodically-updated interactive applications and the corresponding user satisfaction-aware dynamic frequency governor considering heterogeneous multicores.

II. Methodology

In this section, we will describe the FDR metric and present the FDR CPU/GPU frequency governor.

978-3-9819263-0-9/DATE18/©2018 EDAA

	Good	Excellent
User1	0.34 (8)	X
User2	0.09 (1)	0.24 (39)
User3	0.36 (9)	0.32 (65)
User4	0.26 (29)	X
User5	0.37 (3)	0.27 (33)
User6	0.36 (32)	0.27 (27)
User7	X	0.29 (14)
User8	0.37 (40)	0.30 (3)
User9	0.12 (33)	0.10 (4)
User10	0.11 (29)	0.08 (75)
User11	0.35 (11)	0.28 (63)
User12	X	0.32 (13)
User13	0.36 (8)	0.30 (66)
User14	0.34 (1)	0.16 (10)

TABLE I
FRACTION OF FRAME DROPS AND USER SATISFACTION

A. FDR Metric

The FDR metric, which is the ratio of number of frames actually drawn into the frame buffer and number of frames scheduled to be drawn per unit time, is defined as follows.

$$FDR = a/(a+b), \tag{1}$$

where a is the number of frames that are actually drawn at a given interval, b is the number of frames dropped during that interval, and thus $a+b$ is the number of frames that the application is intended to draw in the given interval. FDR is a value between 0 and 1. The higher the ratio, the fewer number of frames are dropped.

The FDR value that satisfies the participants can be calculated as follows.

$$FDR_{good} = \sum_i FDR\_good_i / \sum_i QT\_good_i, \tag{2}$$

where FDR_good_i is the i^{th} participant's FDR value under the "Good" score and QT_good_i is the number of questionnaires that the i^{th} participant gave that score. The result shows that the participants consider the display quality as "Good" if FDR is above 0.73. Therefore, we use $FDR_{good} = 0.73$ as the optimization target for the FDR CPU/GPU frequency governor.

B. FDR Frequency Governor

The basic idea of the FDR CPU/GPU frequency governor is to set the frequencies of CPU and GPU based on the FDR metric, FDR_{good}. Since most smartphones have multiple heterogeneous CPU cores, e.g. big and little cores, our governor should also take this factor into account.

The proposed governor, described in Algorithm 1, consists of two parts: achieving FDR_{good} (Line 3 to Line 27 in Algorithm 1) and maintaining FDR_{good} (Line 29 to Line 40 in Algorithm 1).

1) Achieving FDR_{good}: In the CPU cost part, we have to consider both the little core and the big core capability, because the contribution of the little core and the big core to the CPU cost is not equal, even when they are at the same frequency level. Thus, we need to know the performance difference between the little core and the big core caused by the frequency. Geekbench [11], which is the processor performance measurement benchmark, is used to measure the little core and the big core processor performance scores. The score is calculated based on processor computation capability and pre-defined computation capability score. Thus, the score can represent the performance. We have measured the little core and the big core performance at three frequency levels; lowest, medium, and highest and used weka [12], a data mining tool with a collection of machine learning algorithm, to build a performance score regression model for big/little core based on frequency. The polynomial regression model is used because of the lowest error rate. Based on the performance score model, we could know the little core and the big core performance score for the unit of frequency and find that the performance difference between them is 1.5. After evaluating the CPU cost contributed by the little core and the big core, the respective core frequency can be determined.

2) Maintaining FDR_{good}: When the FDR_{good} is achieved, it does not imply that the chosen CPU/GPU frequency is the best, because CPU/GPU utilization may be too low and are not fully utilized. Actually, we can save more power by using a higher CPU/GPU utilization and lower CPU/GPU frequency [1], [13], [14]. The work in [15] proposed an adaptive on-line CPU-GPU DVFS algorithm for 3D gaming applications that tried to maintain FPS at a user satisfactory level. They determined whether the current state was CPU or GPU dominant. If it was CPU dominant, when the CPU utilization exceeded 80%, the FPS was determined by the CPU. The algorithm tried to scale up the CPU frequency to meet FPS requirement. At the same time, the GPU frequency was scaled down to meet the target utilization (80%) to save more energy. Besides, the coefficient of correlation between CPU/GPU frequency and utilization, and the coefficient of correlation between CPU/GPU frequency and FPS were used to select the appropriate CPU/GPU frequency that was sufficient to achieve the target FPS. In our FDR-based algorithm, we have adopted the algorithm in [15] to meet the target CPU/GPU utilization to shorten CPU/GPU idle time.

III. EVALUATION

A. Experiment Setup

FDR CPU/GPU governor is implemented on Samsung Galaxy S7 to evaluate the power efficiency and user satisfaction. We have also compared it to the default governor. The default independent CPU and GPU governors of S7 are Interactive [16] and OnDemand [16] governor respectively. Both governors make DVFS decision based only on CPU and GPU utilization. We have also compared proposed governor to FPS_30 and FPS_60 CPU/GPU governors [1], which try to maintain the FPS at 30 and 60 respectively. Facebook and Instagram, which are the two most popular social networking APPs, are used for experiments. A power monitor from Monsoon [17] is used to measure the power consumption of the smartphone.

Algorithm 1 FDR CPU/GPU frequency governor

Require:
FDR: frame drawn rate
FDR_{good}: FDR value with good QoE
U_C (U_G): CPU (GPU) utilization
F_{BC} (F_{LC}): CPU big (little) core frequency
F_G := GPU frequency
$\alpha_{LC} = \Delta F_{LC}/\Delta U_C$
$\alpha_{BC} = \Delta F_{BC}/\Delta U_C$
$\alpha_G = \Delta F_G/\Delta U_G$
Ensure: Finding lowest CPU/GPU frequency that maintains FDR
for every epoch **do**
  Achieving the target FDR, FDR_{good}
  if $FDR \neq FDR_{good}$ **then**
    CpuCost = $U_C \times F_{LC} + C_u \times F_{BC} \times 1.5$
    GpuCost = $U_G \times F_G$
    if $a == 0$ **then**
      Choose the lowest CPU/GPU frequency
    else
      PC = CpuCost/a; PG = GpuCost/a
      TargetFrameRate = $(a + b) \times FDR_{accept}$
      TargetCpuCost = PC $\times$ TargetFrameRate
      TargetGpuCost = PG $\times$ TargetFrameRate
      Choose the lowest GPU frequency, F'_G, such that
      $F'_G \times G_U \geq$ TargetGpuCost
      LittleCoreMaxCost = MaximumLittleCoreFrequency $\times C_U$
      if LittleCoreMaxCost $\geq$ TargetGpuCost **then**
        Choose the lowest little core frequency, F'_{LC}, such that
        $F'_{LC} \times C_U \geq$ TargetCpuCost
        $F_{LC} = F'_{LC}$
        $F_{BC} = 0$
      else
        Choose the lowest big core frequency, F'_{BC}, such that
        $F'_{BC} \times C_U \times 1.5 \geq$ TargetCpuCost $-$ LittleCoreMaxCost
        F'_{LC} = LittleCoreMaxCost
        $F_{BC} = F'_{BC}$
      end if
    end if
  else
    Maintaining the target FDR, FDR_{good}
    if $C_u \leq 80\%$ **then**
      $F'_{LC} = (U_C - 80\%) \times \alpha_{LC} + F_{LC}$
      $F'_{BC} = (U_C - 80\%) \times \alpha_{BC} + F_{BC}$
      $F_{LC} = F'_{LC}$
      $F_{BC} = F'_{BC}$
    end if
    if $G_u \leq 80\%$ **then**
      $F'_G = (U_G - 80\%) \times \alpha_G + F_G$
      $F_G = F'_G$
    end if
  end if
end for

In this experiment, the 12 participants evaluate and rate their satisfaction levels for their experiences in using we recruited 12 participants to evaluate and rate their satisfaction levels in their daily lives using Facebook and Instagram under the four governors: FDR, Default, FPS_30, and FPS_60.

B. Evaluation Results

1) Power consumption: Figure 1 illustrates the comparison of normalized power consumption for each governor across different applications. The power consumption value is normalized to FPS_60 governor because it consumes the largest power. The orange bar is the normalized power consumption of FDR governor and it consumes less power compare to other governors on both Facebook and Instagram. For Facebook, compare to Default, FPS_30, and FPS_60, FDR governor saves average 15.88%, 22.16%, and 42.21% respectively. For Instagram, compare to Default, FPS_30, and FPS_60, FDR governor saves 18.39%, 35.00%, and 51.82% respectively. On average across Facebook and Instagram, our FDR governor can improve 17.13% energy efficiency against Default governor. We also show the problem mentioned earlier that if we employ FPS governors, which use FPS as QoE metric, for power management on aperiodic-update applications, it would cause a waste of power. The FDR governor can improve average 25.58% and 47.01% power efficiency compare to FPS_30 and FPS_60, respectively.

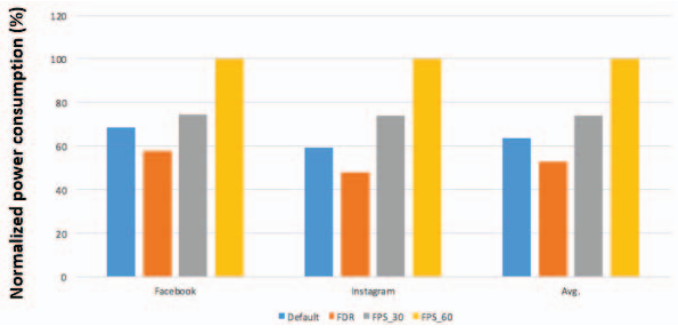

Fig. 1. The normalized power consumption with different governor across different applications.

2) User satisfaction: Figure 2 and Figure 3 show the 12 participants' QoE rating for each governor while using Facebook and Instagram. The results indicate that all of the participants gave positive QoE rating (above 4) to the FDR governor, which represents *good*. None of the participants felt poor or bad with the performance provided by FDR governor.

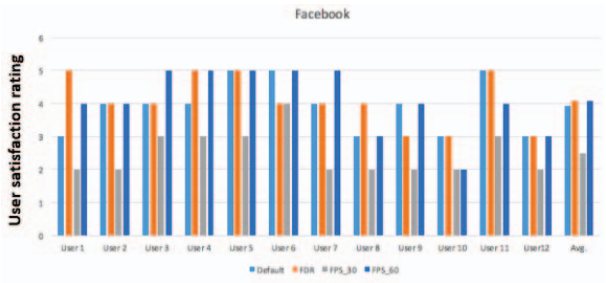

Fig. 2. The participants' QoE rating with each governor while using Facebook.

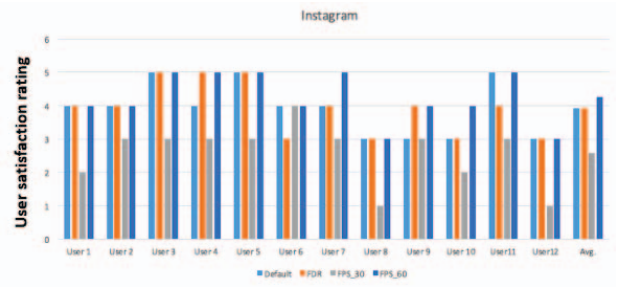

Fig. 3. The participants' QoE rating with each governor while using Instagram.

Figure 4 shows the average user's QoE rating for each governor across Facebook and Instagram of all 12 participants, and shows the standard deviations using error bars. According to standard deviations bar, we can see that there is a significant variation of QoE for each participant. Some participants were very sensitive to the performance while others were less sensitive to the performance. The average user's QoE rating was 3.95 for default governor, 4.04 for FDR governor,

2.52 for FPS_30 governor, and 4.08 for FPS_60 governor. FPS_60 governor received the highest QoE rating because it tried to meet 60 FPS QoE requirements and always set CPU/GPU frequency at highest level, but it was not energy efficient. Compared to the FPS_60 governor, the proposed FDR governor consumed less power by 47.01% while all the participants also felt satisfied with it.

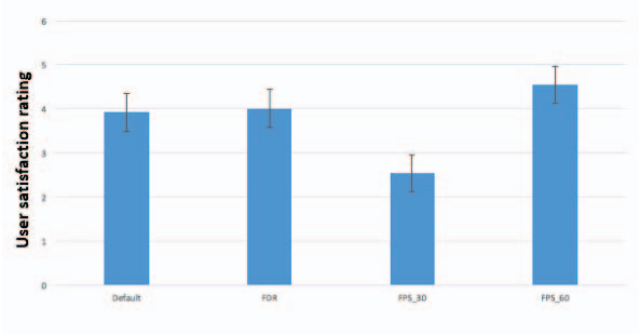

Fig. 4. Average QoE rating across 12 participants for each governor.

The experimental result of power efficiency and user's QoE show that the proposed FDR governor could achieve 17.13% power improvement on average against Default governor without sacrificing the user experience.

3) FDR goverenor overhead: Figure 5 shows the overhead of FDR governor each times while using Facebook. We measure CPU utilization occupied by FDR governor, which is calculated by the ratio of busy time taken by FDR governor over CPU total time, to represent the computation overhead of the FDR governor. The overhead consists of three parts. 1) Detect the current use of application is Facebook or Instagram. 2) Access the required input information. 3) Estimate the desired CPU/GPU frequency and set frequency. Over all, the computation overhead of our approach is 2.25% on average.

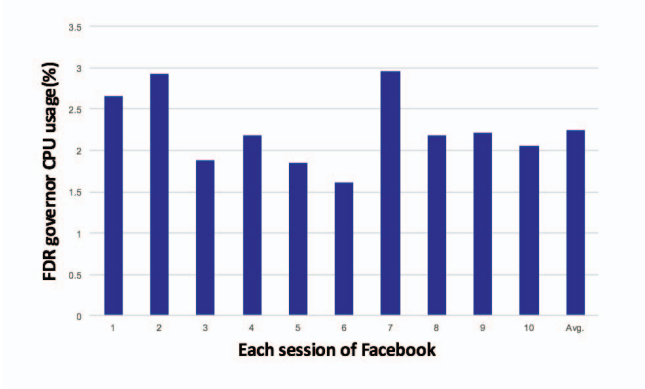

Fig. 5. The FDR governor overhead while using Facebook.

IV. Conclusion

In this paper, we propose a new metric, called FDR, to characterize user perceived display quality for aperiodically-updated interactive applications. The proposed FDR CPU/GPU frequency governor is compared with Default on power consumption. We also ask real users to use the smartphone and evaluate the proposed FDR governor by rating their satisfaction with the display quality, which show that the proposed FDR governor can improve 17.13% power consumption on average against the Default governor. We also show that employing FPS as the metric for power management on aperiodically updated application would cause energy inefficiency. The proposed FDR governor could save 28.58% and 47.01% power on average when compared with the FPS_30 and FPS_60 governors respectively.

V. Acknowledgment

This work was financially supported by the Ministry of Science and Technology of Taiwan under Grants MOST 105-2622-8-002-002, and sponsored by MediaTek Inc., Taiwan.

References

[1] A. Pathania, Q. Jiao, A. Prakash, and T. Mitra, "Integrated cpu-gpu power management for 3d mobile games," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.
[2] L. Yang, R. P. Dick, G. Memik, and P. Dinda, "Happe: Human and application-driven frequency scaling for processor power efficiency," *IEEE transactions on mobile computing*, vol. 12, no. 8, pp. 1546–1557, 2013.
[3] S. Bischoff, A. Hansson, and B. M. Al-Hashimi, "Applying of quality of experience to system optimisation," in *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2013 23rd International Workshop on*. IEEE, 2013, pp. 91–98.
[4] W.-M. Chen, S.-W. Cheng, P.-C. Hsiu, and T.-W. Kuo, "A user-centric cpu-gpu governing framework for 3d games on mobile devices," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2015, pp. 224–231.
[5] Y. Ahn and K.-S. Chung, "User-centric power management for embedded cpus using cpu bandwidth control," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2388–2397, 2016.
[6] Y. Zhu, M. Halpern, and V. J. Reddi, "Event-based scheduling for energy-efficient qos (eqos) in mobile web applications," in *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*. IEEE, 2015, pp. 137–149.
[7] W. Song, N. Sung, B.-G. Chun, and J. Kim, "Reducing energy consumption of smartphones using user-perceived response time analysis," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, p. 20.
[8] J. M. Hektner, J. A. Schmidt, and M. Csikszentmihalyi, *Experience sampling method: Measuring the quality of everyday life*. Sage, 2007.
[9] K. Wac, S. Ickin, J.-H. Hong, L. Janowski, M. Fiedler, and A. K. Dey, "Studying the experience of mobile applications used in different contexts of daily life," in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack*. ACM, 2011, pp. 7–12.
[10] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank, "Factors influencing quality of experience," in *Quality of experience*. Springer, 2014, pp. 55–72.
[11] Geekbench 4. [Online]. Available: https://browser.primatelabs.com/
[12] Weka. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/
[13] J.-G. Park, C.-Y. Hsieh, N. Dutt, and S.-S. Lim, "Quality-aware mobile graphics workload characterization for energy-efficient dvfs design," in *Embedded Systems for Real-time Multimedia (ESTIMedia), 2014 IEEE 12th Symposium on*. IEEE, 2014, pp. 70–79.
[14] B. Dietrich and S. Chakraborty, "Lightweight graphics instrumentation for game state-specific power management in android." *Multimedia Systems*, vol. 20, no. 5, 2014.
[15] P.-K. Chuang, Y.-S. Chen, and P.-H. Huang, "An adaptive on-line cpu-gpu governor for games on mobile devices," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 2017, pp. 653–658.
[16] Android governor. [Online]. Available: http://bit.ly/2vBGxy4
[17] Monsoon solutions inc. [Online]. Available: https://www.msoon.com/