REQUESTED-QOS DRIVEN RUNTIME RECONFIGURATION OF MOBILE DEVICES

Hiren Joshi, S. S. Verma

Information Technology Department,
MLV Textile Institute,
Bhilwara, India.
Email: hiren@mlvti.ac.in, ssv445@rediff.com

G. K. Sharma

Information Technology Department,
ABV-Indian Institute of IT and Management,
Gwalior, India.
Email: gksharma@iiitm.ac.in

ABSTRACT

The present paper embodies a technique for on-the-fly remote reconfiguration of mobile device for power optimization according to the Requested Quality of Service (RQoS) from user. The method is initiated by mobile device user who sends RQoS parameters to remote server, where the multimedia algorithms are re-optimized as per RQoS, and a new bit-stream is transmitted to reconfigure the device for low power with acceptable degradation in quality. Experimental results show that the approach is able to reduce bitwidth by 64.31% to 73.95% for varying PSNR from 40dB to 26dB. Much higher power savings can be achieved if user is willing to tolerate more quality degradation.

1. INTRODUCTION

The user of any system has moral right to get services of expected quality level. To fulfill the requirement of Quality of Services (QoS), designers reserve extra resources. Nevertheless, this attitude results into an inefficient design in terms of power. This penalty of power for getting higher quality is acceptable if user is in a position to afford it, but problem starts when the user is out of power and he still requires to pay for quality. Therefore, the system must adapt itself with respect to changes in user requirement of QoS. We require a system-on-chip (SoC) architecture, which facilitates QoS driven runtime reconfiguration in order to save the energy. When user feels that a device is running out-of-power, he can save power by requesting low quality of service. The proposed well-structured framework presents a practical approach for run-time high-level optimization for power in multimedia devices. Presently, this framework is experimented for bit width optimization of MPEG-2 and results agreed with the practical importance of runtime and remote optimization of bitwidth for power saving. We are intending to implement the other high-level optimization e.g. Loop-unrolling and DVS etc. in the time to come.

Bit-width optimization (BWO) is the process of analyzing the program code and finding out exact bit width required for representation of each variable. This is used to optimize multimedia algorithms during high-level synthesis. Several loss less static BWO techniques have already been proposed, but the overall saving can be increased if lossy approach is adapted according to run-time requirements. Although the application of lossy approach results in a moderately degraded quality but one can save significant amount of power with this acceptance. In lossy approach we reduce the bit width of different variables of multimedia algorithms beyond the permissible limit and achieve significant power saving.

The variables are classified as critical and non-critical. Critical variables in an algorithm are variables whose accuracy must not be alerted in order to retain functionality. We have developed a unique approach, “Critical Variable Partitioning” for the required sub grouping of variables. One can reduce the accuracy of non-critical variable but upto what extent? This is the next major issue to be concerned and is incorporated as “Bit width v/s Quality trade off” in our approach. Further, the framework suggests that the runtime BWO must be done at remote server resulting in power, time and resource saving of mobile device and then the result may be transmitted as bit stream for dynamic reconfiguration.

2. PROPOSED FRAMEWORK

The overall approach is explained in fig 1.Server possesses the results of “Critical Variable Partitioning” (CVP) and “Bit width v/s Quality Trade off” (BQT) processes, which are statically done only once for a particular multimedia algorithm. The server process is initiated by the client's request. Whenever user feels that the device is running out-of-power, he selects the required level of QoS, which is converted into technical specifications by client interface of the framework and parameters are sent to the remote server through communication network. The RQoS parameters received from client are used as a guideline for lossy BWO. It reduces the accuracy of non-critical variables resulting in to graceful degradation of quality. The optimization is done on program code equivalent SUIF-IR obtained using SUIF compiler. This design is then synthesized into reconfiguration bit stream using SUIF2VHDL converters and standard synthesis tools. Now the bit-stream is sent to

remote client machine using communication link for dynamic reconfiguration of it.

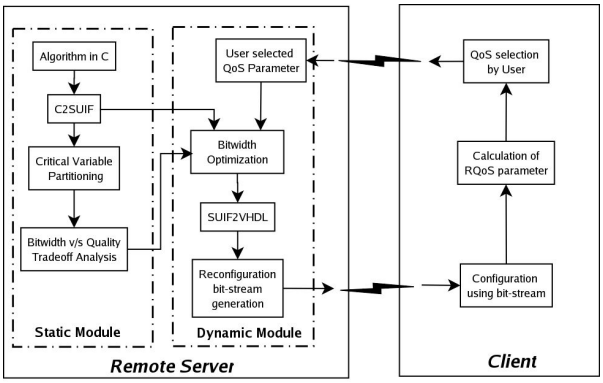

Figure. 1. Server and Client Role in Framework

At the client's end the user perception of QoS are mapped into technical parameters. Client model provides an easy interface to user for selecting QoS levels. The quality is divided into 8 levels based upon different quality aspects. This division is the result of extensive experimentation with multimedia system and their users. The user, whenever requires, selects a quality level in which he feels comfortable. Now client interface converts these quality levels into corresponding technical specifications such as image resolution, bandwidth, frame rate etc. The specifications are then transmitted to remote server, where it is incorporated as "user selected QoS parameter".

3. EXPERIMENTS AND RESULTS

For evaluation of methodology proposed, here we are presenting the experimental work and results. The experiments were solely on the functionality of MPEG2decoder. We are here describing the experiments on *idctrow* and *idctcol*, the core functions of mpeg decoder algorithm. Both the functions are computation intensive. The functions contain 9 integer variables (x0 to x8). We refined bitwidth of each of 9 variables firstly by applying Forward and Backward Bitwidth Analysis and then the approach of CVP and BQT. Thus each variable is now fine-tuned to contribute maximum in power saving. We calculated the bitwidth for 40 dB to 26 dB PSNR. (The PSNR was calculated with reference to image obtained by standard implementation of MPEG decoder). The results obtained shows that we can save 64.31% bitwidth for different algorithms at 40dB PSNR, while the saving is 73.95% for 26dB PSNR. (See fig.2 and fig.3) The framework supports every positive PSNR but our experiments showed that the quality of image degraded beyond the acceptable range for PSNR less than 26dB so here we are presenting only the saving between 40dB to 26dB PSNR.

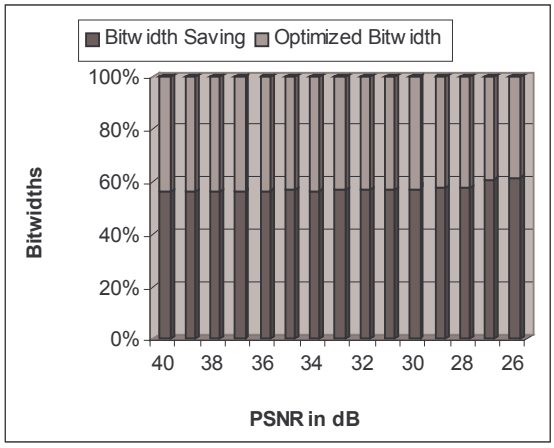

Figure. 2. Bitwidth Saving in IDCTCOL Algorithm

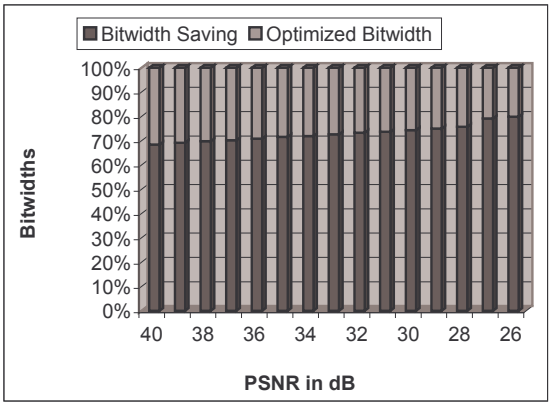

Figure. 3. Bitwidth Saving in IDCTROW Algorithm

4. CONCLUSION

The challenge of power optimization will remain as a subject to be investigated in the time ahead, as the next generation multimedia applications will demand more power. Therefore, it is very unlikely that the static, pre-tuned, solutions will succeed in the near future. Thus, adaptive and reconfigurable systems assume a role of extreme relevance in this context. The proposed framework at present, explores the power saving capability of run-time BWO with acceptable compromise in quality of multimedia. However, in near future, we will incorporate other high-level optimization in the framework.