Low power and QoS

Miodrag Potkonjak

University of California,
Los Angeles
miodrag@cs.ucla.edu

Quality of Service has recently emerged as one of most important research topics and engineering problems in a number of fields, including the Internet, multimedia, and wireless communication. However, the interest in QoS in these fields has not been matched by similar interest among system designers. For example, system-synthesis literature and system-on-a-chip research has not addressed QoS. Therefore, a large and growing gap exists between theoretical discussions of QoS in the multimedia and networking literature and the practical application of QoS paradigms in system synthesis. At the same time, researchers have pursued many aspects of low-power designs and have proposed a variety of modeling and optimization techniques. Also, there are hundreds of research papers on, and dozens of tools for, power minimization at essentially all stages of the design process. Nevertheless, no one has addressed the relationship between power optimization and QoS.

Therefore, in the very near future we will likely see a flurry of R&D to realize design methodologies and synthesis tools that incorporate QoS methodologies and optimize system power consumption. To build a basis and provide an impetus for this research, I will briefly survey the state-of-the-art QoS research and practice. I will not strive for completeness and mathematical accuracy. Instead, my goal is to provide a global picture and insights that can facilitate easier entry into low-power QoS research.

QoS BASICS

Generally, QoS is an optimization or guarantee (or both) of the user's perceived utility of service under constraints of the allocated or available resources. QoS has several dimensions. The most popular, at least from the research viewpoint, deals with *multiresolution*: providing service at several quality levels. A prime example is video (or image), where the resolution of the individual frames and the presented number of frames per second can vary.

In practice, several other QoS dimensions are often much more important. For example, *latency*, *throughput*, *synchronization*, and *resiliency to jitter* are commonly primary design goals or mandatory constraints. In addition to the objective QoS parameters, which can be uniquely quantified (for example, the *bit error rate*), there are subjective QoS parameters (for example, the quality of speech in audio compression coders and the subjective quality of music reproduction), which can be only statistically characterized by sampling a large number of subjects.

While the list of QoS dimensions is long and will evolve with the changing nature of applications, relatively few metrics dominate the current practice. Two traditional metrics of speed are latency and throughput. Latency indicates the delay between when the CPU receives an instance of a task and when the CPU processes that instance; throughput indicates the rate at which tasks can be processed.

In many multimedia applications, synchronization is of primary importance. It is the time discrepancy between two corresponding samples in two data streams. The presentation of a movie without proper synchronization is very annoying to viewers. Usually, synchronization is a part of operating system utility and scheduling functions. Because in many general-purpose systems this imposes significant overhead, several proposed DSP and communication operating systems delegate this functionality to the application program.

In wireless communication, the key QoS metric is often the bit error rate. BER is usually defined as the percentage of incorrectly received bits of data. Sometimes, it is normalized against power consumption. Another popular communication QoS metric is resiliency against jitter. This measure is of special interest when the process or transmitted data are periodic, as is the case for most audio and video data. In this case, jitter is the measure of irregularity in the period. Jitter, like other QoS metrics, particularly latency, is related most directly to the size of the buffer space required to ensure the smooth final presentation of data.

In many applications, consistency and fairness are important. Consistency measures how often the QoS changes. Fairness is an upper bound of the discrepancy in some other QoS metrics (for example, latency and jitter) experienced by any two users in a set of participants. For example, in many Internet games, fairness is of the highest importance.

You can quantify metrics in various ways. For example, you can be interested in the largest value (or some other metric measure) for a particular QoS dimension. Another alternative is a statistical quantification through the average (mean), median, or moment, or through more complex methods.

THE POWER-OPTIMIZATION ISSUE

In CMOS design, dynamic power dominates power consumption because of the discharge of logic and interconnect capacitance. The widely accepted power model has four components. Power depends linearly on *switched capacitance* (which can be approximated by the chip size) and *activity* (how often capacitance is charged). It also depends quadratically on the used *supply voltage*. Low voltage implies lower *operational speed*.

Semiconductor technology trends have greatly affected how power is optimized. For example, you can leverage lower power consumption by changing voltage as QoS requirements dictate. For instance, when you need low latency, you can increase voltage, and lower it when higher latency is acceptable.

Current QoS models

Two QoS models currently dominate: the *QoS resource-allocation model*[1] and the *demand-supply model*.[2–4] Q-RAM, proposed by Ragunathan Rajkumar, postulates the nonmonotonically increasing tradeoff between the amount of allocated resources and the utility to the user. A key advantage of this model is its generic nature. For example, as a resource you can consider diverse dimensions such as time, computational power (CPU time), amount of storage, and power. The model has at least one severe limitation: it imposes a constant demand for a resource over time. Because many of the most interesting applications are intrinsically bursty, the model's practical application domain is limited.

The DS model, developed by Rene Cruz, addresses burstiness quite well. It assumes periodic segmentation of time. During each period, each process receives a task of generally varying complexity. A process's cumulative sum of tasks can be depicted as a demand curve imposed on the system. The system serves the task sequentially by allocating resources during each time period to one of the processes. The cumulative sum of the processed data forms a supply curve. The model models several dimensions of QoS exceptionally well. For example, latency is the horizontal distance between the demand curve and the supply curve for a process. Synchronization is the discrepancy between two demand curves at a given moment of time. Therefore, synchronization corresponds to the current difference between the arrival moments of currently served tasks for the processes. Nevertheless, the DS model has several limitations. One key limitation is the representation's complexity. A typical movie, for example, requires storing, analyzing, and processing tens of thousands of time slots for each curve.

The future of QoS and power consumption

Researchers will likely develop new QoS models to overcome the limitations of these models. For example, I envision a *statistical DS model*. The SDS model starts from the DS and acknowledges the importance of capturing a process's timing behavior. However, instead of using the complete record of the process, this model would use a set of statistical parameters that suitably and compactly describe the process. These parameters include features such as average value, variance, and correlation. In particular, I propose using powerful statistical nonparametric modeling and validation techniques as modeling tools. Applying statistical features drastically reduces the complexity of a process's description. It also provides suitable representation for reasoning about, and the optimization of, the implementation of QoS applications. Incorporating Q-RAM into the SDS model is straightforward.

Once the new QoS statistical models are developed and mathematically analyzed, they will provide a basis for low-power QoS research at several levels of the design process, from system synthesis, to architecture definition and algorithm selection, to development of real-time operating systems (RTOS) and compilers. The goal will be to develop modular yet easy-to-coordinate methodologies and tools for low-power QoS system synthesis. The OS and compilation steps will be emphasized.

At the algorithm level, I foresee tools that analyze structurally different yet functionally equivalent algorithms to identify those most suitable for providing the requested QoS level while minimizing power. For example, for the compression task, researchers could develop algorithms that produce data of different average levels of compression and different variance in compression effectiveness. For a system that supports variable voltage, the key is to select an algorithm with a low variance. On the other hand, if the system supports shutdown, an algorithm with as bursty an output as possible is preferable.

At the architecture level, I see a strong need to develop new architectures of general-purpose and network processing elements that support QoS requirements. Our preliminary study indicates that altering the voltage supply in the system pipeline of network processors can reduce power by an order of magnitude.[5] During compilation, we can use code-generation and transformation techniques to minimize power while satisfying the required QoS metrics. We can select scheduling and transformation techniques that support latency, synchronization, or the preferable level of burstiness.

RTOS research will include the development of mechanisms for CPU time, memory, and I/O allocation to maximize QoS for a given power. Power-optimization degrees of freedom could include predictive shutdown, access control for new processes, and variable-voltage techniques. Other preliminary studies indicate that judiciously simultaneously employing statistical shutdown techniques and variable voltage mechanisms can reduce power by an order of magnitude while preserving the QoS utility level.[6,7]

As with any prediction, some or even many of my predictions might not be realized. However, in the near future, QoS will doubtlessly receive a great deal of attention from system designers, system software researchers, and CAD tool developers.

References

1. R. Rajkumar et al., "A Resource Allocation Model for QoS Management," *Proc. 18th IEEE Real-Time Systems Symp.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp. 298–307.
2. R.L. Cruz, "A Calculus for Network Delay, I: Network Elements in Isolation," *IEEE Trans. Information Theory*, Vol. 37, No. 1, Jan. 1991, pp. 114–131.
3. R.L. Cruz, "A Calculus for Network Delay, II: Network Analysis," *IEEE Trans. Information Theory*, Vol. 37, No. 1, Jan. 1991, pp. 132–141.
4. R. Agrawal et al., "A Framework for Adaptive Service Guarantees," *Proc. 36th Ann. Allerton Conf. Communication, Control, and Computing*, Univ. of Illinois, Urbana-Champaign, Ill., 1998, pp. 693–702.
5. G. Qu and M. Potkonjak, "Energy Minimization of Communication Pipelines on Variable Voltage Processor," *Proc. ICCAD '98: Int'l Conf. Computer-Aided Design*, ACM Press, New York, 1998, pp. 597–600.
6. I. Hong et al., "Synthesis Techniques for Low-Power Hard Real-Time Systems on Variable Voltage Processor," *Proc. 1998 Real-Time Systems Symp.*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 178–187.
7. K.T. Kornegay, G. Qu, and M. Potkonjak, "Quality of Service and System Design," *Proc. IEEE Workshop on VLSI '99*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1999, pp. 112–117.

Miodrag Potkonjak is an associate professor in the Computer Science Department at the University of California, Los Angeles. He received his PhD in electrical engineering and computer science from the University of California, Berkeley. Contact him at the Computer Science Dept., 3532G Boelter Hall, UCLA, Los Angeles, CA 90095-1596; miodrag@cs.ucla.edu.