Achieving Utility Arbitrarily Close to the Optimal with Limited Energy

Gang Qu and Miodrag Potkonjak
Computer Science Department, University of California, Los Angeles, CA 90095

Abstract

Energy is one of the limited resources for modern systems, especially the battery-operated devices and personal digital assistants. The backlog in new technologies for more powerful battery is changing the traditional system design philosophies. For example, due to the limitation on battery life, it is more realistic to design for the optimal benefit from limited resource rather than design to meet all the applications' requirement. We consider the following problem: a system achieves a certain amount of utility from a set of applications by providing them certain levels of quality of service (QoS). We want to allocate the limited system resources to get the maximal system utility. We formulate this utility maximization problem, which is NP-hard in general, and propose heuristic algorithms that are capable of finding solutions provably arbitrarily close to the optimal. We have also derived explicit formulae to guide the allocation of resources to actually achieve such solutions. Simulation shows that our approach can use 99.9% of the given resource to achieve 25.6% and 32.17% more system utilities over two other heuristics, while providing QoS guarantees to the application program.

1. INTRODUCTION

The advances in the Internet and wireless communication make battery-operated devices much more versatile than ever. On one hand, such advances bring more and more new and complex applications (e.g., multimedia, distributed computing and simulation). On the other hand, they also challenge some of the traditional system design philosophies. For example, because of the limitation of battery life, it is more realistic to design for the optimal benefit from limited resources (such as energy and CPU time) rather than design to meet all the applications' requirements. However, little work has been reported at this end from the CAD society.

In this paper, we discuss how one can use system synthesis techniques to maximize the system utility with a fixed amount of resource. In particular, we consider the following problem: a system achieves certain amount of utility from a set of applications by providing them certain levels of quality of service (QoS), how to allocate the limited energy and CPU time to get the maximal system utility.

ISLPED '00, Rapallo, Italy.
Copyright 2000 ACM 1-58113-190-9/00/0007...$5.00

With the advantages provided by variable supply voltage processor cores [4], we can balance the processor's speed and CPU time from the given amount of energy by dynamically changing the supply voltage. In specific, high voltage yields high speed but consumes more energy while the same energy may support the system longer at a lower voltage level. In this paper, we show how one can optimally combine different levels of voltages to achieve better system utility, which can be arbitrarily close to the optimal.

The remainder of the paper is organized as follows. We define the utility maximization problem (UM) in Section 3. In Section 4, we propose several solutions to the UM problem for a single application system: the optimal approach for multi-voltage system, proof of a basic lemma for variable voltage system based on which we propose the partition and linear approximation heuristics (PLA). Section 5 highlights the error analysis for the PLA heuristics and a constructive proof of solutions arbitrarily close to optimal. In Section 6, a dynamic programming procedure (DP) is presented to solve multi-application system. The simulation results are reported in Section 7 before conclusions.

2. RELATED WORK

The essence of the UM problem is to allocate the limited resources, namely energy and CPU time, to multiple applications in such a way that the system achieves maximal utility. It belongs to a well-studied category of problem: the resource allocation problem. Ibaraki and Katoh's monograph [6] addresses the history, applications, and various approaches for this problem.

Quality of service is a concept that has been frequently discussed in networking community. As the new Internet-based, multimedia, and other applications becomes more and more popular, many research efforts have been put to deliver end-to-end QoS guarantees. Rajkumar et al. [10] present an analytical approach for satisfying multiple QoS dimensions in a resource-constraint environment and provide optimal and near-optimal resource allocation schemes for two special cases. Later on, [11] shows this problem is NP-hard and gives a polynomial algorithm which yields a solution within fixed short distance from the optimal solution as well as an approach by formulating the problem as a mixed integer programming problem. Cruz [3, 12] introduced the arrival curve and service curve in the context of packet-switch networks. From these curves, one can view QoS in terms of backlog, transmission delay and throughput. The problem of satisfying services guarantees becomes a scheduling problem to meet the backlog and latency constraints. Recently, Qu et al.[9] address system design methodology to minimize silicon area while providing synchronization guarantees for multimedia applications.

Power and energy have emerged as one of the most important concerns for system design, and many techniques have been proposed to guide the low-power system design. One of the most powerful methods of reducing power consumption is to lower supply voltage such that the system operates at the point of lowest power consumption [2, 7]. DC-DC converters and dynamic voltage-scaled microprocessor system have been implemented [1, 8], and variable voltage core-based system design methodology to cope with variable voltages has also been proposed [4]. In this paper, we use the variable voltage technique as a tool to dynamically allocate the energy and CPU time among the applications to achieve total system utility arbitrarily close to the optimal.

3. PROBLEM FORMULATION

There are n applications $\{\tau_1, \tau_2, \cdots, \tau_n\}$ available on a system with a fixed amount energy E^0 and m resources $\{\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_m\}$. Each application requires certain amount of resource and consumes energy. The system accrues a value, *utility* $U^i_{v_i}(\mathbf{R}^i)$, from serving application τ_i by allocating it resources $\mathbf{R}^i = (R_{i,1}, \cdots, R_{i,m})$ and supply voltage $v_i(t)$. The energy cost to achieve $U^i_{v_i}(\mathbf{R}^i)$ is given by

$$E_i(\mathbf{R}^i, v_i) = \int_0^T P(v_i(t))dt \qquad (1)$$

where T is the execution time and $P(v_i(t))$ is the power consumption at voltage level $v_i(t)$.

The utility maximization problem is a constrained optimization problem as shown in Figure 1. We want to maximize the system utility (i) within the capacity of each resource (ii) and under the energy budget (iii). Condition (iv) is optional, when it is enforced, each application's QoS guarantee, in terms of the system utility U^i will be provided.

Problem: Utility Maximization with limited resources

Maximize: $\sum_{i=1}^{n} U^i_{v_i}(\mathbf{R}^i)$ (i)

Subject to: $\sum_{i=1}^{n} R^i_j \leq \mathbf{R}_j$ for $1 \leq j \leq m$ (ii)

$\sum_{i=1}^{n} E_i(\mathbf{R}^i, v_i) \leq E^0$ (iii)

$U^i_{v_i}(\mathbf{R}^i) \geq U^i$ for $1 \leq i \leq n$ (iv)

Figure 1: Problem formulation.

If the system has sufficient resource to operate at the nominal voltage $v_{nominal}$ during the entire execution, then the system utility is maximized only if the supply voltage is fixed at $v_{nominal}$. When CPU time is the only resource besides energy, the UM problem simply reduces to the classic resource allocation problems with separable objective functions [6]:

$$\textbf{Maximize:} \quad \sum_{i=1}^{n} U^i_{v_{nominal}}(t_i)$$

$$\textbf{Subject to:} \quad \sum_{i=1}^{n} t_i = 1, \qquad t_i \geq 0$$

For both cases when t_i's are continuous or integer variables, algorithms have been proposed based on Kuhn-Tucker conditions and the Lagrangean relaxation ([6] Chapters 2 and 4). We consider an overloaded system where energy is not enough to support the system at nominal voltage. Now we have to make a choice between high voltage short service time and low voltage long service time.

4. SINGLE APPLICATION SYSTEM

In this section, we provide solutions to the UM problem for system with a single application under various occasions.

4.1 Multiple Voltages System

When there is only one application, the system will schedule, based on the utility function provided by the application, how to spend the limited energy E^0. In particular, how long should the system operate and with which voltage scheme to maximize the utility?

The system, as the service provider, will receive the utility function from the user as a function of CPU time at nominal voltage. When there are only multiple supply voltages available at the same time, the utility functions at different voltages can be obtained either by requiring from the user or by calculating the service time to accumulate the equivalent amount of computation if the utility is a single-variable function of computation.

Figure 2 illustrates the utility *vs.* service time curves at four different levels of supply voltage. Figure 3 plots the power dissipation over various supply voltages based on $P = \alpha C v_{dd}^2 f$. For a given amount of energy E^0, from the latter curve, the service time can be easily calculated from $t_i = \frac{E^0}{P_i}$. Then the utility U_i, achieved from E^0 with supply voltage v_i, can be located from the utility *vs.* service time curves. The voltage that makes the highest utility is chosen and the problem is solved.

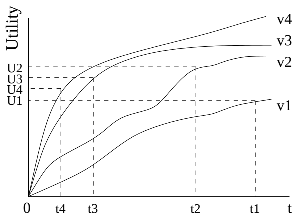

Figure 2: Utility *vs.* service time.

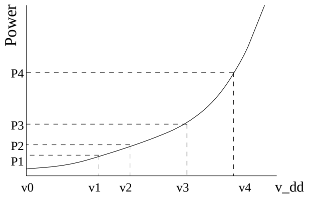

Figure 3: Power *vs.* voltage.

For example, with the four different voltages v_1, v_2, v_3 and v_4 in Figure 3 and E^0, we locate their respective power consumption and find their service time t_1, t_2, t_3 and t_4. Then it becomes clear (in Figure 2) that the utility achieves the maximal U_2 with supply voltage v_2 and service time t_2.

4.2 Variable Voltage System

However, it becomes impractical to obtain all the utility functions for the variable voltage system, where the supply voltages can be changed continuously. We adopt the following approach to approximate the utility function at any level of supply voltage: we first partition the time interval $[0, 1] : 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$, and on each subinterval $[t_{i-1}, t_i]$ we approximate the utility function $U(S_{nominal}, t)$ at the nominal voltage by the linear function of service time t (for $t \in [t_{i-1}, t_i]$):

$$U(S_{nominal}, t) = U(t_{i-1}) + \frac{U(t_i) - U(t_{i-1})}{t_i - t_{i-1}}(t - t_{i-1}) \quad (2)$$

we further assume that U is proportional to the processor's speed and rewrite this as:

$$U(S_{nominal}, t) = a_i S_{nominal} t + b_i S_{nominal}$$

where $S_{nominal}$ is the speed at nominal voltage. Finally, we approximate the utility at voltage v (with the associated speed S):

$$U(S, t) = a_i St + b_i S \quad (3)$$

Figure 4 shows a given utility function at nominal voltage, its linear approximation, and the linear approximation for the utility function with a lower supply voltage.

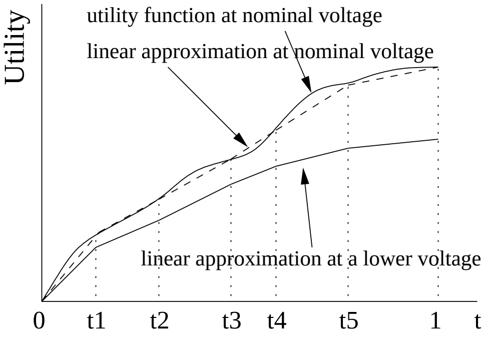

Figure 4: A utility function at nominal voltage and two linear approximations.

Suppose the system services an application whose utility function is given by $U(S, t) = aSt + bS$ with energy E^0 and the finish time is restricted to $[t_1, t_2]$. Let v_i be the voltage such that E^0 can support the system up to time t_i. Denote v_{min} and v_{max} as the minimal and maximal physical possible voltages. Assume the power consumption is $P = k_1 v(v - v_t)^2$ with a supply voltage v and v_t is the threshold voltage. Let v_{opt} be the voltage at which the system runs to gain the maximal utility with E^0, then we have:

Lemma 4.2.1

$$v_{opt} = \begin{cases} \max\{v_{min}, v_2\} & \text{if } b \leq 0; \\ \max\{v_{min}, v_2\} \text{ or } \min\{v_{max}, v_1\} & \text{if } b > 0 \end{cases}$$

[*Proof:*] We know that, to gain a given amount of utility, the energy is minimized when running at a constant voltage [5]. From the duality, with fixed energy, the utility cannot be maximized unless a constant voltage is applied.

Suppose we are able to run the processor at v, consume the given energy E^0 after time t, and let $P = k_1 v(v - v_t)^2$ and $S = k_2 \frac{(v-v_t)^2}{v}$ be the power consumption and speed at v. The linear approximation of utility function at v is:

$$\begin{aligned} U(S, t) &= aSt + bS \\ &= S(\frac{aE^0}{P} + b) \\ &= k_2 \frac{(v - v_t)^2}{v}(\frac{aE^0}{k_1 v(v - v_t)^2} + b) \\ &= \frac{ak_2 E^0}{k_1 v^2} + \frac{bk_2(v - v_t)^2}{v} \end{aligned}$$

The first order condition of the last equation results in the following equation:

$$\frac{dU}{dv} = \frac{1}{k_1 v^3}[bk_1 v(v - v_t)(v + v_t) - 2aE^0] = 0 \quad (4)$$

where a is non-negative from the assumption that utility function is non-decreasing in terms of service time. Recall that any feasible voltage v has to be larger than v_t. When $b \leq 0$, (4) has no solution and $\frac{dU}{dv} < 0$, hence the lower the voltage is, the more the utility. When $b > 0$, (4) has at most one solution $v' > v_t$ and it is easy to verify that a local minimum is achieved at v'. When $v > v'$, we want to use the highest possible voltage $\min\{v_{max}, v_1\}$ and when $v < v'$ utility is maximized at the lowest possible voltage $\max\{v_{min}, v_2\}$, therefore the optimal solution is the one that gains more utility. □

4.3 PLA Heuristic

Figure 5 shows the (**P**artition and **L**inear **A**pproximation) heuristic for finding the best voltage to serve an application with a given amount of energy E^0 in time $[0, 1]$.

```
Input:   $U(t)$, the utility vs. service time at $v_{nominal}$.
         $E^0$, the system's total energy
Output:  The maximal utility that can be gained in $[0, 1]$ with
         $E^0$ and a voltage level $v$ to achieve this.

Procedure PLA: Partition and Linear Approximation
  1)  Partition $[0, 1] : 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$.
  2)  For each $i = 1, 2, \cdots, n$
        find $U(t_i)$ from the utility vs. service time curve $U(t)$
        $a_i = \frac{U(t_i) - U(t_{i-1})}{t_i - t_{i-1}} \cdot \frac{1}{S}$
        $b_i = \frac{U(t_{i-1})}{S} - a_i \cdot t_{i-1}$
      where $S$ is the speed at nominal voltage and $U(0) = 0$.
  3)  In each $[t_{i-1}, t_i]$, apply Lemma 4.2.1, find the optimal
      voltage $v_i^{opt}$ and calculate the corresponding utility $U_i$.
  4)  Choose $i_0$ (may not be unique) such that $U_{i_0} = \max_{i=1}^{n} U_i$.
  5)  Fine-tune the solution by applying either of the followings:
      · recompute the coefficients $a_i, b_i$ from $v_{i_0}^{opt}$, goto step (3);
      · repeat this procedure on interval $[t_{i_0 - 1}, t_{i_0}]$
  6)  Report $U_{i_0}$ and $v_{i_0}$.
```

Figure 5: PLA heuristic for finding the best voltage scheme(s).

We start from partitioning the time interval $[0, 1]$, which determines the quality of the solution. A good partition introduces less error in the linear approximation and results in an accurate solution. Strategies on partitioning will be discussed in the next section. In step 2), we compute, for each subinterval, the coefficients of the linear approximation in equation (3). We then enforce the service time to be in each subinterval and apply Lemma 4.2.1 to find the voltage v_i^{opt} that achieves the (local) maximal utility U_i. An overall best is selected in step 4) and this completes one iteration. The solution is optimal up to the partition in step 1) and the coefficients in step 2). The accuracy of the solution can be iteratively improved as illustrated in step 5). The first one increases the accuracy of the linear approximation, while the other one reduces the error from partitioning.

With the assumption that the utility function is continuous and non-decreasing with respect to the supply voltage and service time, it is known that the best strategy is to keep the voltage constant [5]. Lemma 4.2.1 uses a (uniformly) linear approximation to predict suboptimal solutions to the UM problem which results in the PLA heuristic in Figure

strategy	description	finish time	supply voltage
S_1	the overall optimal strategy	$T_1 \in [t_{k-1}, t_k]$	$v(t) = c_1$
S_2	PLA's best strategy in $[0, 1]$	$0 < T_2 \leq 1$	$v(t) = c_2$
S_3	PLA's best strategy in $[t_{k-1}, t_k]$	$T_3 \in [t_{k-1}, t_k]$	$v(t) = c_3$
S_4	strategy that finishes at t_k with fixed voltage t_k	$T_4 = t_k$	$v(t) = c_4$
S_5	strategy that finishes at t_k with variable voltages	$T_4 = t_k$	$v(t) = v_5(t)$

Table 1: Five different strategies to consume E^0.

5. Without complete knowledge of the utility functions *vs.* service time at all (continuous) supply voltages, we cannot find the optimal voltage that maximizes the utility with the given amount of energy consumption.

5. ERROR ESTIMATION

We analyze PLA procedure's accuracy and answer the question "how close can a PLA-obtained solution be from the optimal?". We claim that, by carefully partitioning, PLA is capable of providing suboptimal solutions within any given bounds in either absolute or relative errors[1].

Let $\mathcal{T} : 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$ be the initial partition, and $\Delta U = \max_{i=1}^{n}\{U(t_i) - U(t_{i-1})\}$, where $U(t)$ is the utility function at nominal voltage $v_{nominal}$. Suppose that with the same amount of energy, the optimal strategy brings a utility U^{opt} and the strategy from the PLA heuristics under the partition $\mathcal{T}$ without any refinement has U^{opt}_{PLA}, then we have:

Lemma 5.1 $U^{opt} - U^{opt}_{PLA} \leq \Delta U$

[*Proof:*] As we pointed out earlier, the best strategy to consume the given energy E^0 is to operate at a constant supply voltage. Consider the following five strategies in Table 1: S_1 is the strategy that consumes E^0 and gets the maximal utility by running a constant supply voltage c_1, supposing the finish time $T_1 \in [t_{k-1}, t_k]$ is in the k-th subinterval of partition $\mathcal{T}$. S_2 is the one reported by PLA at step 4) in Figure 5, which is the overall best solutions from all subintervals $[t_{i-1}, t_i]$. S_3 is the best solution from PLA with the additional constraint that the service has to be completed in $[t_{k-1}, t_k]$. S_4 selects the constant voltage c_4 to consume E_0 at the time t_k. Finally we construct S_5, which also uses all energy at exactly t_k by a variable voltage scheme:

$$v_5(t) = \begin{cases} c_1, & 0 \leq t \leq t_{k-1} \\ v_5(t), & t_{k-1} < t \leq t_k \end{cases}$$

Let U_i be the utility accumulated by $S_i, (i = 1, 2, \cdots, 5)$. Obviously we have:

$$U_1 \geq U_2 \geq U_3 \geq U_4 \geq U_5$$

Comparing S_5 with the optimal strategy S_1, which are identical by time t_{k-1}, notice that S_1 finishes no later than S_5 ($T_1 \leq t_k$), hence there exists voltage profile $v_5(t)$ for $t_{k-1} < t \leq t_k$, such that $v_5(t) \leq v_{max}$ and S_5 can consume the same amount of energy on $[t_{k-1}, t_k]$ as S_1 does on $[t_{k-1}, T_1]$.

Recall that S_1 is the optimal strategy and S_2 is the one PLA gets at step (4) without any refinement, their utility difference is:

$$\begin{aligned} U^{opt} - U^{opt}_{PLA} &= U_1 - U_2 \\ &\leq U_1 - U_5 \\ &= (U_1 - U_5)|_{[t_{k-1}, t_k]} \\ &\leq U_{c_1}(t_k) - U_{c_1}(t_{k-1}) \\ &\leq U(t_k) - U(t_{k-1}) \\ &\leq \Delta U \end{aligned}$$

□

Theorem 5.2 There exists a partition such that PLA (without any refinement) gives a solution with at most an absolute error in the amount of C.

[*Proof:*] It is clear that the theorem holds with partition $\mathcal{T} : 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$, where t_i is selected such that $U(t_i) = i \cdot C$ □

Theorem 5.3 There exists a partition such that PLA (without any refinement) gives a solution with at most a relative error in the amount of c.

[*Proof:*] Let U_0 be the utility achieved by a random strategy, for example, the one that runs at nominal voltage until all the energy is consumed. Define the partition $\mathcal{T} : 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = 1$, with t_i satisfying $U(t_i) = i \cdot c \cdot U_0$. Since $U_0 \leq U^{opt}$, the relative error is:

$$\frac{U^{opt} - U^{opt}_{PLA}}{U^{opt}} \leq \frac{\Delta U}{U_0} = c$$

□

6. MULTI-APPLICATION SYSTEMS

We have showed how to find the optimal voltage for a single application system based on partitioning and linear approximation. In this section, we apply the dynamic programming technique to solve the UM problem for the system with n applications. Consider n independent applications $\tau_1, \tau_2, \cdots, \tau_n$ and their utility *vs.* service time functions. We want to distribute the CPU time among them and determine the voltage level for each τ_i such that the system gets maximal utility with a given amount of energy E^0.

A solution to the n application UM problem is n 4-turples:

$$S = \{(t_i, v_i, U^i_{v_i}(t_i), E^i_{v_i}(t_i)) : i = 1, 2, \cdots, n\}$$

A turple $S_i = (t_i, v_i, U^i_{v_i}(t_i), E^i_{v_i}(t_i))$ represents that the system serves application τ_i for time t_i with supply voltage v_i, achieves utility in the amount of $U^i_{v_i}(t_i)$ and consumes energy $E^i_{v_i}(t_i)$.

To apply the dynamic programming technique, we first discretize both the time and energy domain by dividing the time interval $[0, 1]$ and energy E^0 into L and M equal pieces

[1] Suppose the optimal solution achieves 10 units of utility and another solution yields 8 with the same amount of energy, we define the *absolute error* as $10 - 8 = 2$, and the *relative error* as $\frac{10-8}{10} = 0.2$.

	I: Equal energy				II: Equal execution time				DP's strategy				DP's strategy for 0.1 utility guarantee			
	time	voltage	utility	energy	time	voltage	utility	energy	time	voltage	utility	energy	time	voltage	utility	energy
τ_1	0.45	2.61	0.332	0.188	0.25	3.05	0.088	0.188	0.45	2.85	0.390	0.262	0.45	2.96	0.415	0.300
τ_2	0.05	3.30	0.110	0.050	0.25	3.21	0.143	0.225	0.05	3.05	0.097	0.037	0.05	3.30	0.110	0.050
τ_3	0.20	3.24	0.434	0.188	0.25	3.30	0.500	0.250	0.45	3.22	0.645	0.413	0.15	3.30	0.387	0.150
τ_4	0.30	2.91	0.073	0.188	0.25	2.40	0.035	0.075	0.05	3.05	0.002	0.037	0.35	2.93	0.100	0.225
total	1.00	N/A	**0.949**	0.613	1.00	N/A	**0.766**	0.738	1.00	N/A	**1.134**	0.750	1.00	N/A	**1.013**	0.725

Table 2: Total system utility for the strategy given by the dynamic programming and heuristics I and II (energy is normalized to the amount consumed at $v_{nominal}$ in unit time).

respectively. Let

$$U^k(i,j) = \max\{\sum_{p=1}^{k} U^p_{v_p}(t_p) : \sum_{p=1}^{k} t_p \le \frac{i}{L}, \sum_{p=1}^{k} E^p_{v_p}(t_p) \le \frac{j}{M}E^0\}$$

for $k = 1, 2, \cdots n, 0 \le i \le L$, and $0 \le j \le M$. $U^k(i,j)$ is the maximal utility from applications $\tau_1, \cdots, \tau_k$ in time $[0, \frac{i}{L}]$ with the amount of energy $\frac{j}{M}E^0$. Let

$$S^k(i,j) = \{(t'_p, v'_p) : p = 1, 2, \cdots, k\}$$

be a strategy such that $U^k(i,j)$ is achieved, that is:

$$\sum_{p=1}^{k} t'_p \le \frac{i}{L}, \sum_{p=1}^{k} E^p_{v'_p}(t'_p) \le \frac{j}{M}E^0, \text{ and } \sum_{p=1}^{k} U^p_{v'_p}(t'_p) = U^k(i,j)$$

Notice that $U^n(L,M)$ gives the amount of maximal utility to the original problem and $S^n(L,M)$ provides one of the solutions to the UM problem.

$U^1(i,j)$ and $S^1(i,j)$ can be calculated by the procedure call PLA$(\tau_1, \frac{i}{L}, \frac{j}{M}E^0)$, and generally, the following recurrence formula enables us to compute $U^k(i,j)$ from $U^{k-1}(p,q)$:

$$\begin{aligned} U^k(i,j) = \quad & \max\{\ U^{k-1}(p,q) + U^k_{v_k}(t_k) : 0 \le p \le i, 0 \le q \le j, \\ & t_k \le \frac{p}{L}, E^k_{v_k}(t_k) \le \frac{q}{M}E^0\} \end{aligned}$$

where v_k, t_k, and $U^k_{v_k}(t_k)$ are from procedure PLA$(\tau_k, \frac{p}{L}, \frac{q}{M}E^0)$. $S^k(i,j)$ can also be constructed easily from:

$$S^k(i,j) = S^{k-1}(p,q) \cup (t'_k, v'_k)$$

such that $U^{k-1}(p,q) + U^k_{v'_k}(t'_k) = U^k(i,j), E^k_{v'_k}(t'_k) \le \frac{j-q}{M}E^0$, and $t'_k \le \frac{i-p}{L}$.

The correctness of the above recurrence formulae is based on the observation that any optimal strategy S is comprised of n optimal solutions $S_i (i = 1, 2, \cdots, n)$ to n single application problems: *finding the voltage scheme for application τ_i such that the utility from serving τ_i by time t_i with energy E^i is maximized.*

we propose in Figure 6 the dynamic programming approach and summarize this section by the following theorem:

Theorem 6.1 Procedure DP solves the UM problem with n applications in $O(nL^2M^2)$ time provided that time $[0,1]$ and energy are partitioned into L and M pieces respectively.

```
Input:   $U^i(t)$, the application $\tau_i$'s utility vs. service time
         at nominal voltage $E^0$, the system's total energy
Output:  $(t_i, v_i)$, service time and voltage level for each $\tau_i$
         such that the total system utility gained in $[0,1]$
         with $E^0$ is maximized.
Procedure DP: Dynamic Programming
 1)  Divide time into $L$ and energy into $M$ equal pieces;
 2)  For each $0 \le i \le L, 0 \le j \le M$
       Calculate $U^1(i,j)$ and $(U^1(i,j)$ from PLA$(\tau_1, \frac{i}{L}, \frac{j}{M}E^0)$
 3)  For each $2 \le k \le n$
 4)    For each $0 \le i \le L, 0 \le j \le M$
 5)      Compute $U^k(i,j)$ and $S^k(i,j)$ from $U^{k-1}(.,.)$,
           $S^{k-1}(.,.)$ and PLA procedure calls
 6)  Report $S^n(L,M)$
```

Figure 6: Dynamic Programming approach for the n application UM problem.

7. SIMULATION RESULTS

To demonstrate the effectiveness of PLA heuristics and DP procedure in solving the utility maximization problem, we consider a system with four applications whose utility *vs.* service time functions at the nominal voltage $v_{nominal}$ are given as:

$$U_1(t) = \begin{cases} 0, & \text{if } t < t_1; \\ u_1, & \text{if } t_1 \le t < t_2; \\ u_2, & \text{if } t_2 \le t < t_3; \\ u_3, & \text{if } t \ge t_3. \end{cases} \qquad \begin{aligned} U_2(t) &= a \cdot t + b \\ U_3(t) &= \sqrt{t} \\ U_4(t) &= t^2 \end{aligned}$$

Suppose we have a total execution time 1 and a fixed amount of energy, we want to distribute time and energy among these four applications such that system's utility $\sum_{i=1}^4 U_i(t_i)$ is maximized. We compare the strategy given by the DP procedure with two suboptimal heuristics: (I) assign equal amount of energy to each application and find the best execution partition. (II) assign equal amount of execution time to each application and find the best way to divide energy.

Table 2 shows the system utility achieved by strategies I, II and the one from DP procedure with the following parameter settings: $t_1 = 0.15, t_2 = 0.35, t_3 = 0.45; u_1 = 0.1, u_2 = 0.3, u_3 = 0.5; a = 0.2, b = 0.1; v_{nominal} = 3.3V, v_t = 0.8V$. The given energy can support the system at $v_{nominal}$ for 0.75 unit time. In the DP procedure, both the unit time and given energy are divided into 20 equal pieces.

Heuristics I (or II) does an exhaustive search over all the combinations of execution time (or energy) assignment when energy (or execution time) is evenly assigned to the applications. Variable voltages have also been applied in both heuristics. However, under their respective constraints, the given energy is not completely consumed because we assume the supply voltage can not be higher than $v_{nominal}$. They achieve a total system utility of 0.9491 and 0.7660 respectively. The DP procedure finds a strategy that guides to a total system utility of 1.1337, an improvement of **19%** over heuristics I and **48%** over heuristics II. Moreover, it uses more than **99.9%** of the given amount of energy.

Another benefit from the proposed DP procedure is that it delivers QoS guarantees. For example, if the system wants to achieve at least 0.1 utility from each individual application by meeting their QoS guarantees. Heuristics I can gain this by assigning 0.35, 0.05, 0.20 and 0.40 unit of time to the four applications, but the total utility drops to 0.8817. Heuristics II fails to meet this QoS guarantee because the system cannot gain a 0.1 utility within 0.25 unit time from application τ_4. However, DP procedure finds a solution, as shown in the last four columns of Table 2, which not only provides the QoS guarantees, but also has a total utility 1.0126, better than both heuristics without any QoS guarantees.

Table 3 reports our experiments with other parameter settings and different utility *vs.* service time curves. For eight different tests, we apply the heuristics I, II, and the DP procedure. n represents the number of applications in each test, E is the given energy, and the next three columns show the total system utilities achieved by the three different strategies. The last two columns illustrate DP procedure's improvement over the other two. In average, solutions from the DP procedure guide the system to achieve a utility 25.60% and 32.17% better than heuristics I and II respectively. Notice that both heuristics have exploited the variable voltage technique which enables better performance than traditional fixed supply voltage.

n	E	I	II	DP	DP vs. I	DP vs. II
4	0.75	0.95	0.77	1.13	19.45%	48.00%
4	0.90	1.16	0.97	1.49	28.34%	52.95%
8	0.75	1.90	2.25	2.56	34.67%	13.60%
8	0.90	2.28	1.89	2.70	18.52%	42.69%
10	0.75	2.53	2.38	3.22	27.02%	34.86%
10	0.90	3.13	3.10	3.81	21.62%	23.13%
15	0.75	4.03	4.21	5.43	34.66%	28.91%
15	0.90	5.00	5.32	6.02	20.56%	13.27%
average improvement					25.60%	32.17%

Table 3: More comparison of the system utilities provided by heuristics I, II, and the DP procedure.

8. CONCLUSIONS

As the wireless and battery-operated devices grow popular and powerful at a much faster pace than new technologies for battery, we are facing the problem of designing systems to achieve the optimal benefit from a set of application programs. The traditional system design targets at optimizing the design metrics, such as speed, area, and power, while meeting all the applications' requirement. In this paper, we fill the gap between the new system design requirements and the design philosophies. We show the concept of design for QoS guarantees. Specifically, we consider a system with limited resource and apply the variable voltage design methodology to select voltage profile that guides to the optimal system utility with a given amount of energy.

We formulate the utility maximization problem (UM) which is NP-hard in general. We propose a partition and linear approximation (PLA) heuristics and a dynamic programming (DP) procedure, both of which can guarantee solutions to the UM problem arbitrarily close the optimal. Furthermore, we derive explicit formulae to guide the selection of partition to achieve such solutions. Simulation shows that our approach can use 99.9% of the given resource to achieve 25.60% and 32.17% more system utilities over two other heuristics, while providing QoS guarantees to the application program.

9. REFERENCES

[1] T.D. Burd, T. Pering, A. Stratakos, and R. Brodersen. *A Dynamic Voltage-Scaled Microprocessor System.* IEEE International Solid-State Circuits Conference, February, 2000.

[2] A. Chandrakasan, V. Gutnik, T. Xanthopoulos. *Data driven signal processing: an approach for energy efficient computing.* International Symposium on Low Power Electronics and Design, pp. 374-352, August 1996.

[3] R.L. Cruz. *Quality of Service Guarantees in Virtual Circuit Switched Networks.* IEEE Journal on Selected Areas in Communications, Vol.13, No.6, pp. 1048-1056, August 1995.

[4] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M.B. Srivastava. *Power Optimization of Variable Voltage Core-Based Systems.* 35th ACM/IEEE Design and Automation Conference, pp. 176-181, June 1998.

[5] I. Hong, G. Qu, M. Potkonjak, and M.B. Srivastava. *Synthesis Techniques for Low-Power Hard Real-Time Systems on Variable Voltage Processor.* The 19th IEEE Real-Time Systems Symposium, pp. 178-187, December 1998.

[6] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches.* The MIT Press, 1988.

[7] E. Macii, M. Pedram, and F. Somenzi. *High-Level Power Modeling, Estimation, and Optimization.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.17, No.11, pp. 1061-1079, November 1998.

[8] W. Namgoong, M. Yu, T. Meng. *A high-efficiency variable-voltage CMOS dynamic dc-dc switching regulator.* IEEE International Solid-State Circuits Conference Digest of Technical Papers, pp. 380-381, 489, February 1997.

[9] G. Qu, M. Mesarina, and M. Potkonjak. *System Synthesis of Synchronous Multimedia Applications.* 12th IEEE/ACM International Symposium of System Synthesis. pp. 128-133, November 1999.

[10] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. *A resource allocation model for QoS management.* Proceedings. The 18th IEEE Real-Time Systems Symposium, pp. 298-307, December 1997.

[11] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. *Practical Solutions for QoS-based Resource Allocation Problems.* Proceedings. The 19th IEEE Real-Time Systems Symposium, pp. 296-306, December 1998.

[12] H. Sariowan, R.L. Cruz, and G.C. Polyzos. *Scheduling for quality of service guarantees via service curves.* The 4th International Conference on Computer Communications and Networks, pp. 512-520, 1995.