

考虑电池放电特性的动态电压调节策略研究

许 参, 胡 晨

(东南大学国家 ASIC 系统工程中心, 南京 210096)

[摘要] 电池寿命是移动设备的关键设计参数之一。由于电池容量消耗的非线性和与负载轮廓相关的特性, 最大化电池寿命是个相当困难的问题。研究了在考虑电池放电特性情况下动态电压调节策略(DVS, dynamic voltage scaling)最小化电池容量消耗的问题。针对现有考虑电池放电特性的 DVS 策略中空闲时间分配不合理的缺陷, 提出了空闲时间分布调整过程, 用于优化空闲时间分布从而减少电池容量消耗, 并分析了该过程对任务调度的影响。实验结果表明对现有考虑电池放电特性的 DVS 策略形成的任务调度进行空闲时间分布调整将进一步减少电池容量消耗。

[关键词] 电池优化; 动态电压调节; 空闲时间分布调整

[中图分类号] TP316 [文献标识码] A [文章编号] 1009-1742(2008)02-0079-07

1 前言

现代移动电子设备随着其性能不断提高而需要消耗更多的能量。但是, 在过去的十多年中电池技术的发展远跟不上移动设备对能量需求的增长速度。移动设备能量需求的快速增长和电池容量缓慢提高之间的矛盾日益突出。动态电压调节^[1](DVS, dynamic voltage scaling)技术是降低移动设备能量消耗的有效方法之一。但是对于把电池作为主要能量来源的移动设备, 仅仅通过 DVS 策略来减少设备自身的能量消耗是不够的, 若 DVS 策略产生的负载分布符合电池容量消耗的特点, 则消耗的电池容量将进一步减少。考虑电池放电特性的 DVS 策略(BADVS, battery-aware DVS)就是实现这个目标的方法。

为了实现电池容量消耗最小化, BADVS 策略采用的电池模型应该具有解析表达式。目前, 在众多的电池模型中, 具有解析表达式的电池模型主要有 Peukert 公式^[2]和 Rakhmaotv 等提出的电池模型^[3]。Luo 等^[4]和 Martin 等^[5, 6]分别利用 Peukert 公式提出了 BADVS 策略, 但 Peukert 公式对电池特性的建模不够精确, 这直接影响了 BADVS 策略的电池容量消

耗减少效果。而 Rakhmaotv 模型精确度很高, 但求解该模型表达式和 DVS 调度结合后的优化问题比较困难, 因此多个现有的利用该电池模型来考虑电池放电特性的 BADVS 策略^[7~11]根据由 Rakhmaotv 模型得到的放电原则来指导形成考虑电池特性的任务调度。但这些 BADVS 策略存在一个共同的缺点, 即仅仅根据由该模型得到的定性放电原则来形成任务调度, 这造成了现有的 BADVS 策略有较大的改进余地。笔者提出了空闲时间调整分布过程, 它对现有 BADVS 策略形成的任务调度中空闲时间的分布进行优化, 从而进一步减少电池容量消耗。

2 BADVS 策略分析

为了能够降低处理器的能量消耗, 很多处理器能在多个大小不同的供电电压下正常工作。当处理器电压下降时, 处理器功耗以平方关系减少, 而代价是处理器延时增加(处理器频率减少, 任务运行时间增加)^[1]。在保证任务时序要求前提下, 最小化处理器的能量消耗是 DVS 策略要解决的问题。由于处理器电压和任务运行时间存在对应关系, 则 DVS 问题等价于空闲时间分配问题。BADVS 策略要解决的问题是在保证任务时序要求前提下通过降低任务运

[收稿日期] 2007-05-12; [修回日期] 2007-06-14

[基金项目] 国家自然科学基金资助项目(90407009)

[作者简介] 许 参(1979-), 男, 浙江云和县人, 东南大学国家 ASIC 系统工程中心博士生, 研究方向为低功耗技术、专用集成电路设计

行电压来最小化电池容量消耗。

Rakhmatov 锂电池模型^[3]具有精确、能对主要的电池放电特性(速率容量效应和电荷恢复效应^[8])给出量化描述、具有解析表达式等突出优点,因而基于 Rakhmatov 电池模型的 BADVS 策略受到了较多关注。这些研究中,通常把电池容量消耗

$$\sigma(T) = \sum_{k=1}^n I_k F(T, t_k, t_k + \Delta_k, \beta)$$

最小化作为 BADVS 问题的优化目标,其中

$$F(T, t_k, t_k + \Delta_k, \beta) =$$

$$\Delta_k + 2 \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (T-t_k-\Delta_k)} - e^{-\beta^2 m^2 (T-t_k)}}{\beta^2 m^2}$$

$\sigma(T)$ 表示到时间 T 电池消耗的电荷数, I_k 表示时间 $[t_k, t_k + \Delta_k]$ 内电池放电电流, β 表示电池容量消耗非线性程度。设电池的总容量为 α , 则电池剩余容量 $Q = \alpha - \sigma$ 。一个锂离子电池可以用参数对 (α, β) 表示。

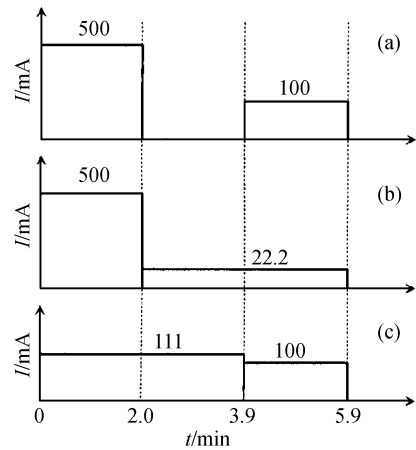
由于电池模型中函数 $F(T, t_1, t_2, \beta)$ 表达式比较复杂,要得到最小化 $\sigma(T)$ 的任务调度比较困难。为此,文献[2, 3]根据函数 σ 的特点提出了几个电池放电原则,其中最重要的一个原则是当给定 2 个完全相同的任务并要把一段空闲时间分配给其中一个任务时,分配给后一任务时的电池容量消耗要小于分配给前一任务时的电池容量消耗。现有的 BADVS 策略利用这个放电原则来指导任务调度形成过程。这些 BADVS 策略基本上都包含两个阶段^[6~9]:

第一阶段生成可行调度。BADVS 策略的可行调度有 2 个要求:a. 满足任务时序约束;b. 确保任务运行过程中不出现电池失效。文献[7~9]都采用 EDF(earliest deadline first)原则确定任务优先级,而 Rakhmatov 等^[10, 11]则根据任务放电电流大小来确定任务运行顺序。

第二阶段空闲时间分配。分配方法是把空闲时间优先分配给后面执行的任务,即按第一阶段形成的任务执行顺序从后到前逐个任务分配空闲时间,每个任务都将得到尽可能多的空闲时间。这个过程反复进行直到不存在可利用的空闲时间或任务电压不能被降得更低为止。

这些 BADVS 策略的电压调节主要集中在第二阶段,电池容量消耗的减少也主要在这个阶段实现,但是这些策略对空闲时间的分配方式是存在问题的。这些策略的空闲时间分配是以前述的放电原则

为依据的,但是该原则是有适用范围的,当用于相同任务时,该放电原则是正确的。然而,对于两个不相同的任务,仍然把空闲时间优先分配给后一任务并不总是最优的。例如,对于电池($40\ 375\ \text{mA} \cdot \text{min}$, $0.273\ \text{s}^{-1}$),当要把两个任务之间的空闲时间分配给两个任务时(如图 1 所示),把空闲时间分配给后一任务时电池剩余容量为 $36\ 914\ \text{mA} \cdot \text{min}$,而把空闲时间分配给前一任务时电池剩余容量为 $37\ 331\ \text{mA} \cdot \text{min}$,此时把空闲时间分配给前面任务更合适。而实际的 DVS 问题通常不在该放电原则的适用范围内,此时仍然把空闲时间优先分配给后面任务并不利于减少电池容量消耗。



(a) 初始状态, (b) 空闲时间分配给后一任务 $Q=36\ 914$,
(c) 空闲时间分配给前一任务 $Q=37\ 331$

图 1 空闲时间分配方式示例

Fig.1 Example for idle time distributions

由于电池模型表达式的复杂性,直接求得 BADVS 问题的最优解非常困难,而现有的 BADVS 策略的缺陷在于空闲时间在各个任务间的分布不合理,笔者求解 BADVS 问题是优化现有 BADVS 策略产生的任务调度中的空闲时间分布,以减少电池容量消耗。由空闲时间分布调整过程实现优化空闲时间分布。

3 空闲时间分布调整过程

3.1 空闲时间分布调整过程原理

由于电池模型中函数 $F(T, t_1, t_2, \beta)$ 表达式比较复杂,要得到全局最优的空闲时间分布比较困难,为此笔者从局部优化入手。局部优化是优化空闲时间在相邻两个任务内的分布。而空闲时间分布的调整从最后的任务开始,从后到前逐步优化相邻

2 个任务中空闲时间在这 2 个任务内的分布。

2 个任务相邻是指前一任务结束后马上开始运行后一任务的情况。若连续的 2 个任务之间被处理器不能利用的时间隔开,则这 2 个任务不能称为相邻,也不可能调整空闲时间在这 2 个任务中的分布。

把由现有 BADVS 策略产生的任务调度记为 S_0 。设 S_0 中任务 i 和任务 j 相邻,任务 i 在任务 j 之前。调度中任务 i 的运行区间为 $[t_1, t_2]$,运行时间 $t_2 - t_1 = \Delta_i + s_i$,即运行时间由完成任务 i 所需的最少时间 Δ_i (此时处理器电压为最大值 V_{\max}) 和分配到的空闲时间 s_i 组成(如图 2 所示),电压为 V_{\max} 时任务 i 的电流为 I_i ,任务 j 的运行区间为 $[t_2, t_3]$,运行时间 $t_3 - t_2 = \Delta_j + s_j$,电压为 V_{\max} 时任务 j 的电流为 I_j 。

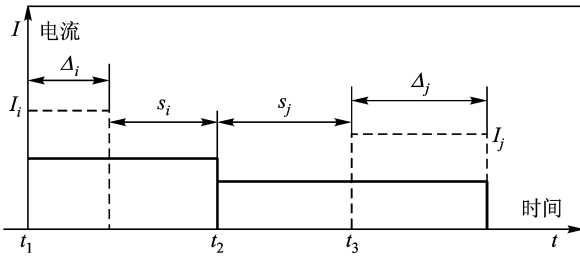


图 2 相邻任务间空闲时间优化分布示意图

Fig.2 Sketch map for the optimal idle time distribution between two nearby tasks

在调度 S_0 中,完成任务 i, j 所需的最少时间分别为 Δ_i 和 Δ_j ,那么只要为任务 i, j 分配运行时间 Δ_i 和 Δ_j 就可以保证 2 个任务能够完成。而由于调低处理器电压将会延长任务运行时间,总共 $s_i + s_j$ 的空闲时间可以分配给 2 个任务而允许其调低处理器电压,从而降低电流和电荷消耗。调度 S_0 中,对于这 $s_i + s_j$ 空闲时间的分布方式为任务 i, j 分配到的空闲时间分别为 s_i 和 s_j ,但从形成调度 S_0 的过程可知,把空闲时间 $s_i + s_j$ 在这 2 个任务内进行如此分布不一定是最优。而优化相邻任务的空闲时间分布就是要使空闲时间 $s_i + s_j$ 在 2 个相邻任务内的分布达到最优。

假设空闲时间重新分布后任务 i 的结束时间为 t ,同时 t 也是任务 j 的开始时间。既然 t 是任务 i 的结束时间,为了保证任务 i 的调度性,则 t 既不能大于任务 i 的完成期限 D_i ,也不能使任务 j 的运行时间小于 Δ_j ($t \leq t_3 - \Delta_j$),因此, $t \leq U_i = \min \{ D_i, t_3 -$

$\Delta_j \}$ 。而 t 也是任务 j 的开始时间,为了保证任务 j 的调度性,它既不能小于任务 j 的发布时间(release time) R_j ,也不能使任务 i 的运行时间小于 Δ_i ($t \leq t_1 + \Delta_i$),因此, $t \geq L_j = \max \{ R_j, t_1 + \Delta_i \}$ 。从而,空闲时间重新分布后 t 要满足 $t \in [L_j, U_i]$ 。可以证明,若任务 i 和任务 j 相邻,则有 $L_j \leq U_i$, t 的可行区间存在。

根据 Rakhmatov 电池模型,空闲时间重新分布后任务 i 和任务 j 消耗的电荷为

$$C_{ij}(t) = H(I_i, \Delta_i, t - t_1) F(T, t_1, t, \beta) + H(I_j, \Delta_j, t_3 - t) F(T, t, t_3, \beta)。$$

函数 $H(I, \Delta, \Delta')$ 表示处理器电压调节时任务运行时间从 Δ 变化到 Δ' 时的电流(已知任务运行时间为 Δ 时的电流 I)。根据 CMOS 集成电路的相关理论,处理器电压调节时任务运行时间和电流的变化关系可以近似为,当处理器电压调节到原来的 s ($0 < s \leq 1$) 倍时,任务运行时间变化为原来的 $1/s$ 倍,电流为原来的 s^2 倍。根据上述关系有

$$H(I_i, \Delta_i, t - t_1) = I_i \Delta_i^2 (t - t_1)^{-2},$$

$$H(I_j, \Delta_j, t_3 - t) = I_j \Delta_j^2 (t_3 - t)^{-2}。$$

因此,局部优化的目标是

$$\min \{ I_i \Delta_i^2 (t - t_1)^{-2} F(T, t_1, t, \beta) + I_j \Delta_j^2 (t_3 - t)^{-2} F(T, t, t_3, \beta) \}, t \in [L_j, U_i]。$$

由于函数 F 的复杂性,用解析方法求解该最值问题几乎不可能,因此需要采用数值方法来求解,该问题是一个关于变量 t 的单变量函数极值问题,可以用平分法和 0.618 法等方法^[12] 进行求解。

如果局部优化的解 $t = t_2$,表明现有调度 S_0 中对于任务 i 和任务 j 的空闲时间分布已经达到最优,不用进行调整;如果 $t \neq t_2$,表明现有调度 S_0 中对于任务 i 和任务 j 的空闲时间分布需要调整,任务 i 的运行区间为 $[t_1, t_2] \rightarrow [t_1, t]$,任务 j 的运行区间为 $[t_2, t_3] \rightarrow [t, t_3]$ 。从上述分析可知,这样的空闲时间调整不会造成任务 i 和 j 违反时序约束,而且调整前后 2 个任务完成的工作量也没有发生变化。

当进行一次局部优化后,只有优化的 2 个相邻任务的运行状态发生变化,调度中其他任务的运行状态没发生任何变化,因此,每一次局部优化都会使电池容量消耗减少或保持不变。对调度 S_0 中所有相邻任务进行这种局部优化将能够进一步减少电池容量消耗。整个任务集一次完整的空闲时间分布调

整过程为:调整从最后的任务开始,从后向前逐步对相邻 2 个任务的空闲时间进行优化分布,一直到第一个任务为止。从后向前进行是由于调度 S_0 中的空闲时间是优先分配给后面的任务,从后向前逐步优化就可以使空闲时间逐渐分布到前面的任务。图 3 中虚线框内的流程给出了一次任务集空闲时间调整的实现过程。在任务集空闲时间调整过程中,若连续的 2 个任务不相邻,则将跳过对这 2 个任务的空闲时间调整,因为这 2 个任务的空闲时间不可能进行调整。

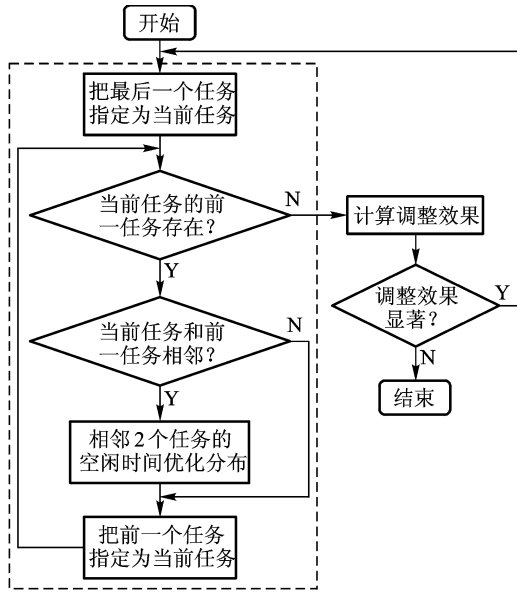


图 3 空闲时间分布调整过程流程图

Fig.3 Flow chart for the procedure of idle time distribution adjustment

设任务集在经过一次空闲时间分布调整后得到的调度为 S_1 , 相应的电池容量消耗为 $C(S_1)$, 由于每次相邻任务的空闲时间优化不会使电池容量消耗增加, 则 $C(S_1) \leq C(S_0)$ 。如果在经过一次空闲时间分布调整后电池容量消耗减少显著(可以用 $C(S_1) \leq \chi C(S_0)$ 表示, χ 为预先设置的常数, $0 < \chi < 1$), 表明调整过程中某些任务的运行时间变化较大, 继续对这些任务进行调整, 将可能进一步显著地减少电池容量消耗。因此, 如果调度 S_1 相对于 S_0 显著地减少了电池容量消耗, 有必要再次进行一次任务集空闲时间调整。如果再次空闲时间调整的效果显著, 则可以继续进行调整。如此反复, 直到某次任务集空闲时间分布调整的效果不显著为止。完整的空闲时间分布调整流程参考图 3。

3.2 空闲时间分布调整性质

设优化函数 $f(t, t_1, t_3) = I_1 \Delta_1^2 (t - t_1)^{-2} F(T, t_1, t, \beta) + I_2 \Delta_2^2 (t_3 - t)^{-2} F(T, t, t_3, \beta)$, 其中 $t \in (t_1, t_3)$ 。该函数有如下性质:

性质 1 函数 $f(t, t_1, t_3)$ 在区间 (t_1, t_3) 内有且仅有一个极点, 这个极点也是最小点。

证明 可以证明 $f'(t, t_1, t_3)$ 只有一个零点且 $f'(t, t_1, t_3) > 0$ 。证明过程略。

由于局部优化的变量 t 的变化区间 $[L_j, U_i] \subset (t_1, t_3)$, 则性质 1 表明优化相邻任务空闲时间分布时, 最优的空闲时间分布是唯一的, 而且用平分法等计算方法得到的极小点就是最优解。

性质 2 设 2 个相邻任务 i (前一个) 和任务 j (后一个) 进行局部优化后的运行区间分别为 (t_1, t_2) 和 (t_2, t_3) 。此时, 若在任务 i 开始时间变化到 t'_1 ($t'_1 > t_1$) 或者任务 j 结束时间变化到 t'_3 ($t'_3 > t_3$) 后再次对这 2 个任务重新进行局部优化, 设优化后 2 个任务的相交时间为 t'_2 , 则 $t'_2 \geq t_2$ 。

性质 2 分 3 种情况来解释: a. 设任务 i 开始时间变化到 t'_1 而任务 j 结束时间不变, 则相比于 (t_1, t_2) 和 (t_2, t_3) 状态, 任务 i 空闲时间减少而任务 j 空闲时间不变, 此时进行优化必然会把任务 j 空闲时间的一部分调整到任务 i , 任务 i 的结束时间将增加, 则 $t'_1 > t_1$; b. 假设任务 i 开始时间不变而任务 j 结束时间变化到 t'_3 , 则相比于 (t_1, t_2) 和 (t_2, t_3) 状态, 任务 i 空闲时间不变而任务 j 空闲时间变大, 进行优化必然会把任务 j 空闲时间的一部分调整到任务 i , 任务 i 结束时间变大, 则 $t'_2 > t_2$; c. 设任务 i 开始时间变化到 t'_1 同时 j 结束时间变化到 t'_3 , 这种情况可以分解为先进行情况 a 变化, 再进行情况 b 变化 2 个过程来实现, 分别对 2 个过程进行局部优化, 2 次优化都将使 2 个任务相交时间变大, $t'_2 > t_2$ 。再考虑到 2 个任务的发布时间或完成期限的限制以及必须完成最少工作量的限制, 重新优化后 2 个任务的相交时间至少不会变小。

对于相邻的任务, 前一任务的结束时间就是后一任务的开始时间, 因而这里仅给出任务结束时间的变化规律。根据性质 2, 空闲时间分布调整过程中任务结束时间的变化规律: 随着空闲时间分布调整的进行, 任务结束时间将不断变大, 或在 L_j 或 U_i 的限制下保持不变, 如图 4 所示。这意味着集中在后面任务的空闲时间逐步被分布到前面的任务。

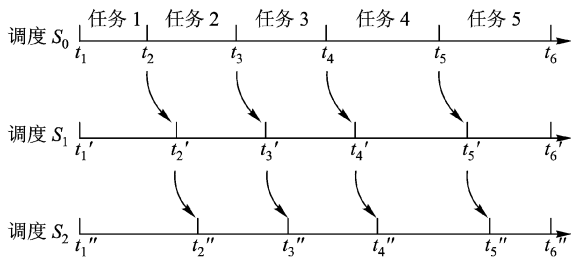


图4 空闲时间分布调整过程中任务运行区间变化示意图

Fig.4 Task running districts changes during the procedure of idle time distribution adjustment

4 BADVS 策略

提出的空闲时间分布调整过程可以用于进一步优化现有的 BADVS 策略。这里给出空闲时间调整过程和 Chowdhury 提出的 BADVS 策略^[7]结合的新策略,它包括 3 个阶段:

1) 根据如下原则生成可行调度:a. 使用 EDF 原则确定任务优先级;b. 尽可能使任务电流按非增顺序排列;c. 确保电池放电过程中不出现电池失效,如果出现失效,则最大程度地调低造成电池失效的任务的电压。

2) 空闲时间分配:把空闲时间优先分配给后面执行的任务,按第一阶段得到的任务执行顺序从后到前逐个任务分配空闲时间,每个任务都将得到尽可能多空闲时间。这个过程反复进行直到不存在可利用的空闲时间或任务电压不能被降得更低为止。

3) 空闲时间分布调整:调整从最后的任务开始,向前逐步优化相邻 2 个任务的空闲时间分布,一直到第一个任务为止。经历一次空闲时间分布调整后,如果电池容量消耗减少显著,则继续进行空闲时间分布调整。空闲时间调整过程的流程可以参考图 3。

其中第一、二阶段为 Chowdhury 提出的 BADVS 策略,第三阶段空闲时间分布调整则是在该策略的调度结果上优化空闲时间分布,进一步减少电池容量消耗。

5 实验验证

为了验证空闲时间分布调整的有效性,笔者对提出的 BADVS 策略进行了一系列的实验。由于 Rakhmatov 模型的精确度比较高,实验中利用该模型来计算电池容量消耗。电池参数为电池 B1(35 220

$\text{mA}\cdot\text{min}, 0.637 \text{ s}^{-1}$)和电池 B2(40 375 $\text{mA}\cdot\text{min}, 0.273 \text{ s}^{-1}$),其中电池 B1 的参数通过对一 Tallwise 型号为 7 260 的锂离子电池的测量数据分析得到,而电池 B2 来自文献[3]。

表 1 给出了一个任务集,它包含 3 个任务,3 个任务的最少工作时间(处理器电压为 V_{\max} 时任务工作时间)都是 2 min,而周期均为 12 min,处理器电压为 V_{\max} 时工作电流分别为 500, 250, 100 mA。3 个任务分别在电池 B1 和电池 B2 上运行 2 个周期。在经过第一、二阶段得到调度 S_0 后,分别根据 2 个电池的参数进行 4 次空闲时间调整,产生的相应调度分别为 S_1 , S_2 , S_3 和 S_4 。图 5 分别给出了调度 S_0 , 在电池 B1 和 B2 上经过 4 次空闲时间调整得到的调度(分别记为 S_1 (B1)和 S_1 (B2))中任务状态图。调度 S_0 根据后面任务优先利用空闲时间的原则把所有空闲时间都分配给了任务 3,而调度 S_1 (B1)和 S_1 (B2)则通过空闲时间分布调整过程把空闲时间分布到 3 个任务。

表 1 初始状态下任务参数

Table 1 Initial task specifications

任务	最少工作时间/min	周期/min	电流/mA
1	2	12	500
2	2	12	250
3	2	12	100

表 2 给出这几种调度下电池容量消耗 σ 以及剩余容量 Q 。可以看出,对于这 2 个电池,4 次空闲时间分布调整后电池容量消耗分别降低到调度 S_0 下电池容量消耗的 64.2%和 57.9%,空闲时间分布调整对于在调度 S_0 基础上减少电池电荷消耗的效果非常显著。

在表 2 中,第二次和第三次空闲时间分布调整的效果已经非常接近,此时继续进行空闲时间调整的效果不显著,第三次调整完成后可以结束空闲时间分布调整过程。

从图 5 和表 2 可以看出电池参数 β (表示电池非线性程度)对 BADVS 策略的影响。在图 5 中,相比电池 B1 上的调度,电池 B2 上的调度中后面的任务得到的空闲时间更多,这是由于 β 越小,电池非线性程度越大,函数 $F(T, t_1, t_2, \beta)$ 值越大,而函数 $F(T, t_1, t_2, \beta)$ 的特点是越接近时间 T 其数值越大,此时把空闲时间分配给后面的任务(更接近时间 T 的任务)更利于降低电池容量消耗。与此同

表 2 空闲时间调整过程中电池容量消耗和
剩余容量变化

Table 2 Consumed battery capacity and residual capacity
changes during the procedure of idle time
distribution adjustment

调度	电池 B1		电池 B2	
	容量消耗	剩余容量	容量消耗	剩余容量
S_0	5 413(100%)	34 962	3 197(100%)	32 023
S_1	3 615(66.8%)	36 760	1 945(60.8%)	33 275
S_2	3 477(64.3%)	36 898	1 856(58.1%)	33 364
S_3	3 468(64.2%)	36 907	1 850(57.9%)	33 370
S_4	3 467(64.2%)	36 908	1 849(57.9%)	33 371

Rakhmatov 模型的 BADVS 策略采用定性的放电原则来指导任务调度,这样的方法不能充分体现电池非线性(即参数 β)对于任务调度的影响。

表 3 给出了空闲时间分布调整过程中任务运行时间的变化,其中的时间表示任务开始时间或结束时间,这些任务运行区间对应于图 5 中的任务。可以看出,随着调整过程的进行,任务结束时间(或开始时间)在不断增加或保持不变,例如,任务 3 第二个周期(电池 B1 下)的开始时间变化为 16→18.59→19.46→19.69→19.74 min,而结束时间保持在其完成期限 24 min。这验证了笔者对空闲时间分布调整过程中任务结束时间变化规律的分析。

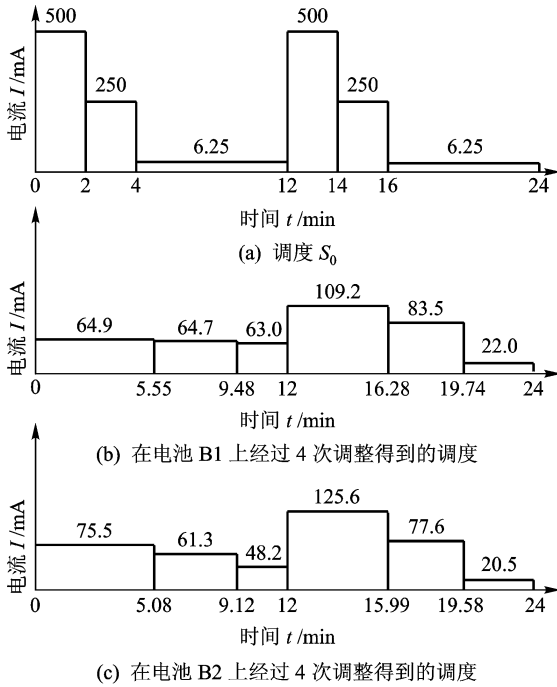


图 5 BADVS 策略产生的任务状态图

Fig.5 Task profiles generated by proposed BADVS policy

时,电压调节也能有效地减少能量消耗。电池的容量消耗是电池非线性放电和电压调节共同作用的结果,这要求 BADVS 策略能够对两者的具体效果进行量化分析,空闲时间分布调整过程则是对两者效果进行量化优化的一种实现方式。而现有基于

表 3 空闲时间调整过程中任务运行时间变化

Table 3 Task running time changes during the procedure of idle time distribution adjustment

调度	B1 下任务运行时间/min	B2 下任务运行时间/min
S_0	(0, 2, 4, 12, 14, 16, 24)	(0, 2, 4, 12, 14, 16, 24)
S_1	(0, 4.65, 8.11, 12, 15.72, 18.59, 24)	(0, 4.37, 7.11, 12, 15.49, 18.48, 24)
S_2	(0, 5.37, 9.18, 12, 16.15, 19.46, 24)	(0, 4.93, 8.82, 12, 15.87, 19.32, 24)
S_3	(0, 5.51, 9.42, 12, 16.25, 19.69, 24)	(0, 5.05, 9.06, 12, 15.97, 19.53, 24)
S_4	(0, 5.55, 9.48, 12, 16.28, 19.74, 24)	(0, 5.08, 9.12, 12, 15.99, 19.58, 24)

6 结语

针对现有 BADVS 策略中空闲时间分布不合理的缺陷,笔者提出了通过优化空闲时间分布来减少电池容量消耗的空闲时间分布调整过程,空闲时间分布调整过程从局部优化入手,从后到前不断优化相邻任务的空闲时间分布。空闲时间分布调整过程中任务运行区间的变化规律为任务结束时间(或开始时间)在不断增加或保持不变。实验结果表明,空闲时间分布调整过程能在现有 BADVS 策略基础上

显著地进一步减少电池容消耗。

参考文献

[1] Kim W, Shin D, Yun H S, et al. Performance comparison of dynamic voltage scaling algorithms for hard real-time systems [J]. IEEE Design and Test of Computers, 2001, 18(2): 10~19

[2] Rao R, Vrudhula S, Rakhmatov D. Battery modeling for energy-aware systems design [J]. IEEE Computer, 2003, 36(12): 77~87

[3] Rakhmatov D N, Vrudhula S, Wallach D A. An model for battery lifetime analysis for organizing applications on pocket computer [J]. IEEE Transaction on VLSI Systems, 2003, 11(6): 1019~1030

- [4] Luo J, Jha N K. Battery-aware static scheduling for distributed real-time embedded systems [A]. Design Automation Conference [C]. Las Vegas, USA, 2001. 444~449
- [5] Martin T L, Siewiorek D P. A power metric for mobile systems [A]. International Symposium on Low Power Electronics and Design [C]. Monterey, USA, 1996. 37~42
- [6] Martin T L. Balancing Batteries, Power and Performance: System Issues in CPU Speed-setting for Mobile Computing [D]. Doctoral Dissertation, Electrical and Computer Engineering Department, Carnegie Mellon University, 1999
- [7] Chowdhury P, Chakrabarti C. Static task-scheduling algorithms for battery-powered DVS systems [J]. IEEE Transactions on VLSI Systems, 2005, 13(2): 226~237
- [8] Zhuo J, Chakrabarti C. An efficient dynamic task scheduling algorithm for battery powered DVS systems [A]. Asia and South Pacific Design Automation Conference [C]. Shanghai, China, 2005. 846~849
- [9] Ahmed J, Chakrabarti C. A dynamic task scheduling algorithm for battery powered DVS systems [A]. International Symposium on Circuits and Systems [C]. Vancouver, Canada, 2004. 813~816
- [10] Rakhmatov D, Sarma V, Chakrabarti C. Battery-conscious task sequencing for portable devices including voltage/clock scaling [A]. Design Automation Conference [C]. New Orleans, USA, 2002. 189~194
- [11] Rakhmatov D. Modeling and Optimization of Energy Supply and Demand for Portable Reconfigurable Electronic Systems [D]. Doctor Thesis, Electrical and Computer Engineering Department, Arizona University, 2002
- [12] 唐焕文,秦学志. 最优化方法[M]. 大连:大连理工大学出版社,1994. 101~104

The Research on Battery-aware Dynamic Voltage Scaling Policy

Xu Shen, Hu Chen

(National ASIC System Engineering Center, Southeast University, Nanjing 210096, China)

[**Abstract**] Battery lifetime is one of the critical design parameters for mobile computing devices. Maximizing the battery lifetime is a particularly difficult problem due to the nonlinearity of the battery discharge behavior and its dependence on the discharge profile. In this paper, the problem of task scheduling with dynamic voltage scaling such that the maximum consume battery capacity is addressed. To deal with the shortcoming of the existing battery-aware DVS policy, the idle time distribution adjustment procedure is proposed, which optimizes idle time distribution and then reduce the battery capacity consumption, and the change of task scheduling under the procedure is analyzed. The experiment results show that the proposed procedure can save more battery capacity consumption evidently than the existing battery-aware DVS policy.

[**Key words**] battery optimization; dynamic voltage scaling; adjustment of idle time distribution