

学习汇报

第八周

熊凯亚

April 28, 2018

本周主要看了高级软件工程课大作业布置的五篇论文。总结如下：

一、Energy Consumption Anatomy of Live Video Streaming from a Smartphone

介绍 本文的主要发现是，拍摄直播视频时消耗电池非常快，手机摄像头在对焦还没开始录制时，大部分电量已经被 camera 内部硬件及与视频处理相关的硬件所消耗。本文还研究了两种优化技术的有效性：使用帧绑定去优化 radio；以及更激进的频率和电压缩放以降低计算功耗。

功率分解分析显示，相机打开但尚未录制时（对焦模式），已经消耗了总功率的最大部分。对焦模式功率消耗不能随视频质量进行调整。功耗强度表明，除了摄像头的图像传感器之外，其他硬件组件在这种能源效率方面也发挥着作用。帧捆绑是减少无线通信所消耗功率部分的有效方法。选择更激进的 DVFS 策略也会降低一些功耗，但可能会降低用户体验。

智能手机电源管理

- WIFI：使用省电模式（Power Saving Mode）提升数据通讯中的电量利用效率，如果没有接收到数据包就进入睡眠模式关闭 radio。
- 计算：当条件允许的时候，使用动态电压频率缩放（DVFS）技术降低 CPU 操作频率。
- camera：影像传感器的能量消耗主要取决于它们的时钟频率。但是其时钟频率并不像 DVFS 那样可以缩放。

电源故障分析

- 分析方法：使用 Monsoon 电量监测器在不同平台（iOS、Android、Windows Phone、Sailfish）分别对显示屏、文件系统、无线网络、CPU/GPU 等电量消耗进行监测。
- 显示屏：显示屏的电量消耗有几个变量，屏幕亮度、屏幕尺寸。为了控制变量使用全黑或全白（好像不对吧，我记得 AMOLED 屏幕对于黑色的显示是肯定比 LCD 省电的）的屏幕并开启飞行模式。实验证明，AMOLED 屏幕比采用 IPS 的 LCD 屏幕明显省电。
- 相机：分析过程相机都聚焦于静态的物体。实验表明，拍摄模式与对焦模式相比，大部分电量消耗为文件 I/O；而且录制时不同的分辨率和不同的比特率不同的帧率对电量的消耗都差不多。

- 无线通讯：使用蜂窝移动数据比使用 WIFI 更加消耗电量。

优化电量消耗

- 从减小实时性限制方面考虑。使用帧绑定技术在多个帧 buffer 在一起后在发送，虽然降低了一点实时性但是可以减少电量的消耗。实验证明，当数据帧包内帧数增多时，耗电量会减少。
- 从降低计算方面考虑。安卓用内核级的管理者来控制 CPU 的频率，如 Performance Governor、Pegasysq Governor、Power Save Governor、On Demand Governor、User Space Governor。实验表明，如果在拍摄视频时使用 Power Save 模式将会带来很糟糕的用户体验。而 Performance 模式在带来最优秀的用户体验的同时却消耗了很多电量。

二、Statistical Power Consumption Analysis and Modeling for GPU-based Computing

介绍 GPU 内部集成了越来越多的晶体管，导致功耗需求稳步上升。本文利用功耗特性，通过运行时性能和动态工作负载之间的内在耦合，以统计分析主流 GPU (NVidia GeForce 8800gt) 的功耗。基于记录的运行时 GPU 工作负载信号，训练良好的统计模型能够强大而准确地预测目标 GPU 的功耗。

本文提出了一种分析和建模 GPU 功耗的新方案。基于记录的功耗，运行时工作负载信号和性能数据，构建了一个统计回归模型，能够根据选定的 GPU 工作负载信号子集动态估计运行时 GPU 的功耗。

数据的获取

- 功耗数据获取：测试计算机运行 benchmark 程序，另外使用特制的数据录制软件记录功耗数据。分别使用了四种 benchmark 程序：OpenGL Geometry Benchmark 1.0, Furmark, Jorik, 和 Parboil。为了提高后续分析和建模算法的效率和稳定性，本文通过平均和下采样操作处理原始功耗数据（每秒 1000 帧）。
- GPU 工作量信号记录：使用 NVidia PerfKit performance analysis tool 分析工具记录了 GPU 运行时的 39 个工作量信号，本文使用了其中五个：*pixel_shader_busy*, *texture_busy*, *geom_busy*, *rop_busy*。这 5 个信号变量表示 GPU 上主要管道阶段的运行时利用率，使用这些变量可以描述 GPU 的工作负载。

统计 GPU 功耗模型 在数学上，SVR（支持向量回归）的基本思想是通过优化公式 1 预测输入 x 的结果 y ，其中 w 和 b 用于描述 SVR 回归模型。LIBSVM 用于 SVR 实现。我们将选择的 SVR 模型的交叉验证结果与基于简单最小二乘的线性回归（SLR）模型进行了比较。图 2 显示了 SLR 和 SVR 之间的交叉验证比较结果。这里，平方误差的总和被用作度量预测质量（即，预测功率数据与原始功耗数据之间的差异）的度量。图 2 的顶部面板显示了 GPU 运行图形程序（OpenGL Geometry Benchmark 1.0）时交叉验证比较结果的一部分（200 秒）。在这种情况下，SLR 和 SVR 的平方和误差度量分别是 656.83 和 589.78。其底部显示了 GPU 运行 GPGPU Jorik 基准时交叉验证比较结果的一部分（80 秒），SLR 和 SVR 的平方和误差度量分别为 44.523 和 39.427。如图所示，无论我们使用图形计算还是 GPGPU 应用程序，选择的支持向量回归（SVR）模型在保留的交叉验证数据集上均优于传统 SLR。

评估和验证 如果 GPU 运行没有 benchmark 的程序，模型的准确性和鲁棒性如何呢？为此，选择图形类和 GPGPU 计算类共八种程序进行测试。使用已经建立的 GPU 功耗统计模型根据记录的 GPU 工作负载信号预

测所选定的八个程序的功耗。并比较预测结果与实际结果的功耗数据，另外还计算了均方误差作为客观度量来衡量预测的准确性。对于大多数时间间隔，模型的功耗预测误差很小且在各帧之间保持一致。但是，对于特定的时间间隔，预测的功耗误差很大。模型的另一个局限性（实际上是统计模型的局限性）是，很难提前预测需要多少训练数据。

总结 通过利用 GPU 的功耗，运行时性能和动态工作负载之间的固有耦合，我们提出了一种基于 GPU 的计算的新型统计功耗分析和建模方案。我们方案的核心部分是 GPU 运行时期动态功耗估计的统计模型。我们展示了我们的统计模型能够准确且可靠地估计 GPU 运行时的功耗，特别是对于图形应用程序。与 CPU 相比，GPU 具有相对简单的缓存层次结构，更多的并行性，更少的复杂控制要求以及更多的计算单元，这使得 GPU 功耗建模与通用处理单元不同。未来，需要依赖 GPU 的详细微观架构知识来提供更复杂和精确的建模方法。此外，在数据预处理步骤中，需要对 GPU 工作负载进行定量分析以及功耗相关工作负载的统计选择。尽管目前存在这些限制，但我们相信这项工作仍然可以作为未来为基于 GPU 的计算应用开发高能效工具的努力的一个有趣的起点。

三、Lightweight Measurement and Estimation of Mobile Ad Energy Consumption

介绍 移动广告是应用生态系统的重要组成部分，开发人员使用广告来产生收入，并且最终用户获得“免费”应用程序。然而，带有广告的应用程序实际上对最终用户在能源，网络使用 and 性能方面具有显著的隐藏成本。因此，开发人员需要平衡广告的使用与这些潜在的负面成本。但是，开发人员缺乏帮助他们衡量应用程序植入广告的潜在成本的技术。为了解决这个问题，本文提出并评估了几种用于衡量和预测广告相关能耗的轻量级统计方法。

对于用户来说，广告需要大量的 CPU 处理时间，这会降低移动应用的响应速度；产生大量的网络流量，通常是应用程序使用的数据量的两倍；并消耗大量的能量，缩短移动设备的电池寿命。对于开发者来说，这些隐藏的成本可能会导致负面评论和最终用户的差评。与能源成本不同，性能和网络成本消耗都可以通过现成的软件或工具（例如，top 和 tcpdump）查询得到。这使开发人员无法轻松确定能源成本的规模或影响。为了解决这个问题，我们提出并评估了两种用于确定最终用户的广告能量成本的轻量级统计方法。这些技术的第一个旨在应用开发生命周期的早期提供反馈。为此，本文构建了一个统计模型，该模型使用来自广告元配置文件的信息，这是发布者从广告网络请求的高级配置信息（例如，广告类型和刷新率）。这种模式允许开发人员获得及早的反馈，即使没有工作实施，他们的应用程序在用户能源成本方面的代价也会很高。这可以使开发人员有机会在软件生命周期的早期比较他们的应用程序，并可能在投资开发工作之前更改其计划的广告使用情况。虽然这种技术可以提供有用的指导，但是一旦功能代码写完，就可以提供更准确的估计值。这是因为能源成本与根据实施结构而变化的运行时信息直接相关。因此，本文引入了第二种技术，可以捕获关键指标并使用这些指标为开发人员提供关于应用广告相关能耗的更准确信息。评估侧重于展示本文提出的用于衡量和预测用户广告相关能源成本的技术的能力。特别是，本文的技术可以使用广告单元配置文件将移动广告的能耗预测控制在 31% 以内，使用真实测量的运行时间度量可以将 14% 以内的移动广告的能耗预测在 14% 以内总体而言，这些结果表明，本文的技术可以提供可操作的信息，并且可以形成指导开发者在他们的应用中使用广告的技术的基础。

为了建立静态模型，本文测量了不同配置下广告的能耗，并对收集的数据进行了统计分析。具体而言，修改了其中一个特定配置（例如 RRATE），同时其他参数保持固定，并且在一天中的不同时间段内在应用程序上运行多次相同的工作负载。实验发现当 SIZE 和 TYPE 固定时，移动广告的平均能耗与 RRATE 的值成线性关系。随着 RRATE 的增加（即广告更新频率较低），广告的平均能耗呈线性下降。在本文的方法中，能源模型用于计算移动广告的能源成本。它将广告相关指标（包括不同频率的 CPU 时间，网络使用率和与时间戳相关的屏幕截图）作为输入，并返回一个数值作为执行期间广告的能耗。

四、Online Estimation of the Remaining Energy Capacity in Mobile Systems Considering System-Wide Power Consumption and Battery Characteristics

介绍 新兴的移动系统通过可充电电池形式的小型能源将许多功能集成到小型设备中。这种情况需要准确估计电池中的剩余能量，以使用户应用程序可以明智地了解它们如何消耗这些稀缺和宝贵的资源。因此，本文着重于评估基于 Android OS 的移动系统的剩余电池能量。本文提出测试 Android 内核，以便根据运行应用程序的实时报告收集和报告准确的子系统活动值。活动信息以及针对智能手机中主要子系统的基于回归的功率宏模型构建的整体系统的功耗评估结果。接下来，在考虑电池的速率 - 容量效应的同时，将总功耗数据转换为电池的能量消耗速率，并且随后基于其当前充电状态信息来计算电池的剩余寿命。最后，本文描述了一个新颖的应用程序设计框架，该框架考虑了目标应用程序的电池状态 (SOC)，电池能量消耗率和服务质量。设计框架的好处通过检查一个原型案例来说明，涉及 Android OS 中基于 GPS 的应用程序的设计空间探索和优化。

精确的功耗建模和估计是任何功耗感知设计方法和低功耗设计工具的关键要求。我们需要系统级功耗信息来开发功耗感知型应用程序，因为现代智能手机应用程序同时使用包括应用程序处理器，显示器，音频，无线通信等在内的许多系统资源。因此必须开发一个电池容量损耗（耗尽）速率估算器，而不是简单的移动系统功耗估算器。前者捕获与电池化学成分有关的重要影响，以及转换和分配损失。从功耗到电池容量损失率的转换不是一个常数因素，而是取决于电池的 SOC，负载功耗和速率容量效应以及环境温度等等的高度非线性函数。所提出的方法使得 Android 内核能够产生准确的子系统活动。功耗数据以及有关电池 SOC 的信息用于估算剩余电池寿命。本文还介绍了一个基于 Android 的移动系统的能量感知应用程序设计框架，其中包含系统级功耗模型，电池状态模型和服务质量模型。通过考虑案例研究中与电池寿命相关的定位分辨率和行程覆盖，可以最大化 GPS 应用的服务质量。一般来说，更高的性能提供更高的服务质量，但也会导致更大的功耗和更短的服务时间。因此，应该设计应用程序，平衡性能和服务时间，以最大化服务质量为目标。

本文介绍了基于 Android OS 的移动系统的精确功率估算，剩余电池电量估算和功耗感知应用设计。考虑到他们的数字和模拟状态，例如显示器亮度，音量等，开发了包括 CPU，Wi-Fi，显示器，音频，GPS，振动电机等主要子组件的功率模型，以及它们的/关闭状态。由于以前没有任何工作将功耗引导至剩余电池寿命，因此根据智能手机活动推导出电池充电状态模型。所有的设备型号都基于商业广泛使用的元件的测量结果。对这些模型进行验证，并与实际平台测量结果进行比较，结果显示更精确的全容量放电结果持续时间，还通过 GPS 应用实践演示了功率感知应用设计。本文进行了设计空间探索，并介绍了质量和电池寿命的折衷。结果表明，通过提出的 poweraware 设计框架，达到了最大的服务质量价值。

五、Design and Implementation of DU Battery Saver based on Android

本文首先从用户的角度提取客户端、服务端功能需求，对其进行功能设计以及实现。点心省电的客户端是基于的开发，使用软件的三层架构的设计理念。点心省电的服务端使用框架为客户端提供不同请求的接口，为了快速的响应客户端的请求设计索引和多级缓存机制。服务端的配置管理平台使用的是设计模式，使得系统可移植性高，易于维护。服务端将不同的功能模块划分成不同的子系统，减小系统的耦合性，不同子系统之间通过消息队列进行通讯。然后从收益的角度考虑，根据服务端下发、客户端上报的大量的有效的数据，利用分析计算的结果，按照一定的策略进行下发广告，使点心省电成为一个闭环的系统，能够及时的发现问题并不影响系统的正常工作。

因为手机是事件触发式的进程管理机制，比如当手机开机时、开机后、网络连接更改、电量充足、电量不足、插上电源、时间更改、断开电源等这些状态发生改变，都会当作是触发事件，从而后台会自动运行相关的进程。这些后台进程会偷偷的使用手机的电量，造成手机电量的使用时间变短。

手机不同硬件的设置对手机的耗电量也有明显的影响，例如对手机的屏幕开关、亮度设置、网络设置等，对手机进行合理的设置会节约一定的电量。

首先会先判断省电的逻辑与硬件设备的逻辑是否一致，若一致，则逻辑结束。若满足智能省电的逻辑，它会根据客户端的具体的场景，来进行省电配置。可以对不同的省电模式进行自动的切换，低电量模式切换、按时切换模式、等其他情形进行智能省电设置。

智能省电会判断当前操作是否为高危操作，若不是高危操作，直接切换，消息通知即可。若是高危操作，则需要判断操作的类型是打还是关闭，若是打开，直接操作即可。若为打开通话，直接切换消息通知即可；若是打开网络，则要判断打开网络的类型，直接打开即可，消息通知，若是打开移动网络，则要进行询问。若是关闭操作，则需要判断是关闭网络或是关闭通话，分别进行不同的操作。若是不可用的网络，直接关闭即可，消息通知用户；若是可用的网络，则要对用户进行询问，根据用户的选择再进行操作。若是在通话中对其进行关闭，关闭后要向用户询问。询问的时候，判断手机屏幕若为亮，若用户操作导致的用户可以自己进行选择，若非用户操作导致的，进行切换，并向用户发送消息。发送消息时，若为夜间，发送消息则无提示音；若不是夜间，则可发送有声音的提示，使得用户得到最好的体验。用户可以查看手机电池的续航时间，点击全面检测即可检测手机前后台耗电的软件，并可以进行相应的省电操作，设置自己的省电模式，以及用户可以使用健康充电功能，最有效的利用现有的手机电量，延长手机电池的使用时间，减慢电池的衰减率，点心省电客户端系统基本实现了一个省电工具类应用这一目标。