Software Development

Support for Energy Efficient Computing

CHRISTOPHER STONE

SEPTEMBER 2013

Building a community of innovators in the development and application of EEC technologies

tinyurl.com/eecsig

Special Interest Group

Energy Efficient Computing

Energy Efficient Computing Report 2013
Special Interest Group

1 Introduction

This paper provides an overview of some technologies and features provided by a selection of processor manufacturers and third-party tool vendors to assist embedded and application software developers to write energy-efficient code. The research was carried out using information provided by the companies and from internet sources. The processor companies reviewed were AMD, ARM, Imagination Technologies, Intel, NVIDIA and XMOS, while the third party companies were Allinea Software, Codeplay Software, Embecosm, Lauterbach, Mentor Graphics, Numerical Algorithms Group, Paralant and UltraSoC Technologies.

Energy efficient computing (EEC) is a rapidly-growing field. Across the whole spectrum of computing, from mobile devices through data centres and into high performance computing there is much activity occurring to improve the energy efficiency of computing and communication systems. One of the key technologies required to enable EEC is the provision of energy-aware and energy-efficient software. Historically, application software has not been associated with energy efficiency. The traditional software development process and layered design mean that application software is usually unaware of its impact on energy usage and a historical emphasis on performance as the ultimate benchmark has served to enforce this disassociation. However, for EEC to be successful, software developers must be aware of the impact of software on energy consumed and be proactive in optimising for energy efficiency.

This work was commissioned by the Energy Efficient Computing Special Interest Group[1] (EEC SIG) and was undertaken in June 2013. The EEC SIG is a partnership between the Technology Strategy Board (TSB)[2], the ESP KTN[3] and the ICT KTN[4]. It was formed to accelerate the development and commercial use of products, processes and services based on EEC technology in support of the TSB's Emerging Technologies & Industries programme.

2 Power Management

All of the processor manufacturers surveyed provide power management features accessible by software. These involve combinations of voltage and frequency scaling and the ability to power down unused portions of the hardware to some level of granularity. Similar features are also often provided for other devices in a system. However, whilst such features are ubiquitous, the exact feature sets and implementation are very much device specific. For example, Intel's recently introduced Haswell architecture[5] provides additional sleep states compared to its predecessors. In addition, the programming of such features is often non-trivial, involving time-energy trade-offs, sequences of hardware events, and dependencies between system components, all requiring specialised device knowledge. There may also be matters such as thermal management issues that must be considered when using these features.

For these reasons, the programming of hardware power management features is not appropriate for high-level application developers and is limited to bare-metal or low-level system developers and those developing embedded systems. Operating systems provide application software with access to power management abstractions through Application Programming Interfaces (APIs), with the operating system kernel and device drivers managing the hardware. Typically, these APIs only allow the developer to manipulate system policies and do not provide for micro-management of power settings. Operating systems also provide support for developers of device drivers to integrate their device driver into the operating system's power management framework.

Microsoft Windows has a comprehensive set of power management features available to application developers[6] and supporting developer documentation[7]. Microsoft offer various tools to diagnose issues with power management, such as

the Windows Performance Toolkit[8] and Pwrtest[9].

Linux has various subsystems that are accessible by application developers to manage such things as CPU voltage and frequency, sleep states when idling, the powering of CPU cores and scheduler parameters. The power management quality of service (PM QoS) infrastructure allows applications to tune certain aspects of power management[10]. PowerTOP[11] is a popular Linux utility for analysing power consumption and power management.

Android, though based on Linux and using its power management features, takes a somewhat different approach to power management, because it is designed for mobile devices. Here, the default power state is off and an application must request system devices to stay alive by acquiring wakelocks via the Android PowerManager service[12].

Mac OSX has a framework for developing device drivers (I/O Kit[13]) that may also be used by application programmers to gain access to hardware features, including power management. Apple's forthcoming OSX Mavericks release emphasises energy efficiency and contains several new technologies relevant to application developers[14].

Mentor Graphics Nucleus is a Real-time Operating System (RTOS) that claims to be the first RTOS to contain integrated power management features[15]. The operating system provides application developers with similar power management services as those found in general-purpose operating systems[16].

3 Instruction Set Architecture

The Instruction Set Architecture (ISA) of a processor provides potential for software developers to optimise their code for energy efficiency, as the choice and sequence of instructions used for a program affects the amount of energy used by the processor.

Embecosm[17] are a company that can provide compiler implementations for energy efficiency and are currently involved with the University of Bristol in a TSB-funded project to give all compilers the option to generate energy-efficient code. But even in the absence of specific compiler support for energy efficiency, benefits may still be gained as optimising for performance often also brings about savings in the energy consumption of software.

Most CPUs have a set of SIMD (Single Instruction Multiple Data) instructions that may be used to improve the energy efficiency of code. These instructions perform the same operation on multiple data items in parallel. Examples include Intel SSE and AVX[18] (also supported by AMD), ARM's NEON instructions[19] and Imagination Technologies' MIPS SIMD Architecture (MSA)[20]. Availability of these features varies across processor families. For example, there are multiple versions of SSE, and for Haswell Intel has introduced AVX2[21], currently not available on other architectures. Similarly, specialised instructions are sometimes provided targeting DSP (Digital Signal Processing) operations, e.g., the ARM DSP extensions[22] and Imagination Technologies' MIPS DSP ASE[23].

Instruction sets are not static and a few new instructions are usually added to the instruction set each time a new CPU is released. These additions usually target improved performance of particular operations, but may also aid energy efficiency. However, the processor-specific nature of ISAs and their changing landscape mean that this area is not best suited to application software.

In most cases the choice of instructions used will be decided by the compiler. This is the most convenient and practical way of exploiting these instructions as their use is essentially transparent to the programmer and allows source code to be independent of target hardware. Usually, the compiler will automatically use SIMD and other specialised instructions as appropriate. For example, Intel[24], AMD[25] and ARM[26] compilers all support loop-based parallelism.

There are other ways in which a developer can employ specific instructions from a high-level language without resorting to assembly language. In some cases compilers support language extensions and data types for vector arithmetic, but vendors also often provide intrinsic functions for hardware-specific functionality that are callable from standard high-level languages, e.g., Intel Intrinsics[27] and ARM's intrinsic functions[28]. The ARM intrinsics also provide functions for power management programming[29].

Numerical libraries are also available that can be linked into applications to provide optimised functions. These are usually for applications with specific performance requirements, such as scientific, image, video, and signal processing functions. AMD[30],[31], Intel[32],[33], NVIDIA[34] and the Numerical Algorithms Group[35] all provide such libraries.

Compilers may also support parallel operations across multiple cores. Languages for GPUs (Graphics Processing Units) such as those used with NVIDIA's CUDA platform[36] support extensions to provide a parallel programming model. Intel has an open-source compiler called ispc[37] that provides an SPMD (Single Program Multiple Data) programming environment that takes advantage of both vector instructions and multiple cores to provide a programming model closer to that used for GPUs.

Numerical precision is another area where instruction choice can yield energy efficiency benefits. ARM provides compiler extensions to support half-precision floating point[38]. Android's Renderscript[39] (see Section 4) allows for different levels of floating point precision via a compiler directive, whilst Filterscript[40] (Section 4) provides relaxed floating point precision to improve energy efficiency.

Other aspects of code generated by the compiler may also be optimised to improve energy efficiency. For example, memory alignment, register allocation and parameter ordering can all affect both performance and energy usage.

4 Heterogeneous Computing

Heterogeneous architectures are increasingly being used for reasons of performance and energy efficiency. This trend provides both opportunities and challenges for software developers. The availability of specialised architectures means that there are opportunities for developers to write software that is more energy efficient, but doing so increases the level of difficulty due to differences in underlying hardware architectures and programming models.

The ARM big.LITTLE architecture[41] provides the programmer with a straightforward approach to heterogeneity. Here, two processors with differing characteristics are used to allow tasks or CPU workloads to migrate between processors. This activity is usually controlled by the operating system in a manner similar to that used for power management, and the scenario relies on the processors being architecturally identical. However, this arrangement is atypical and normally in heterogeneous computing processors with different architectures are used, with their respective strengths being exploited by the type of code that each executes.

One way of doing this is for the programmer to provide hints to the compiler, in the form of compiler directives, as to which parts of the code should be executed on accelerator hardware. The OpenACC standard, supported by NVIDIA[42] is an example of this approach.

All of the GPU manufacturers surveyed provide software development tools to support heterogeneous computing on their respective platforms, but for reasons of portability and developer productivity a common platform for software development across hardware from different vendors is desirable. OpenCL (Open Computing Language) is an industry effort to provide an open, cross-platform standard and is supported by AMD, Apple, ARM, Altera, Imagination Technologies, Intel, NVIDIA, Qualcomm, Samsung, TI, Vivante and others[43].

A more recent development is the HSA (Heterogeneous System Architecture) Foundation. This is an industry consortium founded by AMD, ARM, Imagination Technologies, MediaTek, Qualcomm, Samsung and TI to create a platform for heterogeneous computing across diverse hardware architectures[44], with an open source runtime and software tools.

Aparapi[45] is an AMD developed technology, now open source, that allows Java code to take advantage of hardware accelerators. It consists of an API and runtime that converts data parallel Java code fragments to OpenCL for execution on an accelerator. Bolt[46] is a C++ template library for heterogeneous computing also developed by AMD and open sourced. Both of these technologies are available as part of the AMD Accelerated Parallel Processing (APP) SDK[47].

The Android operating system provides Renderscript, which provides a hardware-independent platform for heterogeneous computing and Filterscript, a subset of Renderscript with stricter constraints, but improved optimisation.

Data transfers within heterogeneous memory systems can consume relatively large amounts of energy. Codeplay Software[48] produces compilers for GPUs and other specialised processors and has developed C++ language extensions that provide abstractions for such data movements. These allow existing C++ programs to be ported quickly to the target architecture. To assist in this effort there is also a tool, developed under the EU project *Low power GPU* (LPGPU)[49], for logging and visualising hidden and implicit data transfers.

5 Managed Runtime Environments

A Managed Runtime Environment (MRTE) such as an implementation of the Java Virtual Machine[51], the Android Dalvik Virtual Machine[50], or the Common Language Infrastructure[52] provides the facility to run hardware-independent code in a virtual environment.

As the MRTE has access to information about the runtime behaviour of the program it can provide adaptive optimisation as needed. Modern MRTEs provide Just in Time (JIT) compilation during runtime and offer dynamic optimisation, such as compiling code hot spots to native machine code as required.

In principle, the MRTE could target specific processor instructions or perform instruction scheduling to maximise efficiency,

but in this survey we have not found evidence of this type of optimisation in vendors' offerings.

One aspect of MRTEs that can be affected by the developer is the set of options used to launch the MRTE. Typically, these provide control over the memory management and garbage collection algorithms used, as well as other facets of the runtime that may influence energy efficiency.

6 Debugging and Profiling

Debugging and profiling tools are available from both processor hardware manufacturers and third party vendors. These tools all offer a similar range features for developers and by highlighting problem areas or performance bottlenecks, such as serial portions of GPU code, may contribute to improving energy efficiency.

Allinea Software specialise in debugging and profiling tools for parallel applications, including those running on GPUs from ARM, Intel and NVIDIA. One of these tools, Allinea MAP[53], an MPI profiler, enables developers to understand where software is spending its time burning processor cycles. It is able to profile costly processor instructions and operations to peripherals, such as disk access, and is particularly designed for multiple processes running together.

Apple's performance analysis tool, Instruments, supports the logging and analysis of energy usage events for iOS devices using the Energy Diagnostics Trace Template[54].

The ARM Development Studio 5 (DS-5) includes the ARM Streamline Performance Analyzer. This allows power consumption to be profiled using the ARM Energy Probe or National Instruments data acquisition devices[55].

The Intel VTune Amplifier XE performance profiling tool can be used to improve the power consumption of software[56].

Lauterbach[57] produce hardware-assisted debug, trace and logic analysis tools for a large number of processor architectures. Their TRACE32 product supports energy profiling using a hardware probe allowing real-time measurement and triggering of current, voltage and power. A cache analysis tool is also available that may be used to optimise cache usage to reduce power consumption.

UltraSoC Technologies has recently been awarded TSB funding to develop, with the University of Cambridge, software tools based on its hardware-assisted debug technology to aid developers in optimising energy usage[58].

7 Training

There is little training available to assist developers in writing energy-efficient code. Many companies offer training courses covering product architecture (which may touch on hardware power management) and procedural information in the use of development tools, (which may include a section on performance optimisation), but we did not find any courses specifically targeted towards EEC.

Allinea Software provides training in optimization and performance improvement.

ARM software development courses cover power management and ARM has a training course for NEON programming.

Codeplay Software are planning to introduce training in the near future.

Embecosm provides bespoke engineer-to-engineer training packages for all of its technology.

Lauterbach offer free training courses on all aspects of their tools to any group of design engineers within the UK.

Numerical Algorithms Group provides training on its numerical routines and on parallel programming techniques to enable programmers to write more efficient code.

Apple provides some tips in the iOS App Programming Guide on how to reduce power consumption[59] and videos are available online of sessions from the 2013 Apple Worldwide Developers Conference (WWDC) relating to the energy efficiency features of OSX Mavericks[60].

A section of Intel's website is dedicated to energy-efficient software development[61] and contains information and resources relevant to their products. Intel also has a power efficiency forum[62] where developers can exchange information. The Intel white papers *Energy-Efficient Software Guidelines*[63] and *Energy-Efficient Platforms – Considerations for Application Software and Services*[64] contain much useful information for developers.

8 Conclusion

The use of hardware-specific features by software developers to implement EEC is generally only appropriate for low-level software such as operating systems, device drivers, runtime environments, compilers and embedded applications. The use of hardware features requires specialised knowledge on the part of the programmer and a skill set unlikely to be possessed by application developers. Moreover, good design principles are that application software should be as portable and hardware independent as possible, meaning that hardware-specific energy-efficiency measures should be performed, where possible, below the level of the application software. This is particularly important given the constant evolution of computer hardware and the industry's move towards heterogeneous computing.

By their nature, some specialised applications and embedded systems will require hand-tuning using low-level hardware-specific features to achieve best results. However, in many cases automatic optimisation for energy efficiency by compilers, operating systems and runtime environments has the potential to deliver benefits to applications with little or no impact to the applications themselves. In this way, investments in energy efficiency made by system software are available to all components running higher up the stack.

Ideally, software developers should be able to specify energy-efficiency intent rather than its implementation, for example in the form of hints and options to the compiler and runtime systems. More support for energy efficiency in the software tool chain is undoubtedly needed, but much can still be achieved through a hardware-agnostic, common sense approach to design and coding for energy efficiency. An important factor in achieving energy efficiency is a good choice of algorithms and data structures. There is no substitute for sound software engineering practices and software developers have to understand the importance of taking energy considerations into account in their designs and implementations. There is a need for developer education in energy-aware design and coding practices to improve the energy-efficiency credentials of future software.

References

1. https://connect.innovateuk.org/web/eec/overview
2. https://www.innovateuk.org/
3. https://connect.innovateuk.org/web/espktn
4. https://connect.innovateuk.org/web/ictktn
5. http://www.intel.com/content/www/us/en/processors/core/core-processor-family.html
6. http://msdn.microsoft.com/en-us/library/windows/desktop/bb968807%28v=vs.85%29.aspx
7. http://msdn.microsoft.com/en-us/library/windows/hardware/gg463220.aspx
8. http://msdn.microsoft.com/en-us/library/hh162945.aspx
9. http://msdn.microsoft.com/en-us/library/windows/hardware/ff550682%28v=vs.85%29.aspx
10. https://www.kernel.org/doc/
11. https://01.org/powertop/
12. http://developer.android.com/reference/android/os/PowerManager.html
13. https://developer.apple.com/library/mac/#documentation/devicedrivers/conceptual/IOKitFundamentals/PowerMgmt/PowerMgmt.html#//apple_ref/doc/uid/TP0000020-BABCCBIJ
14. http://images.apple.com/osx/preview/docs/OSX_Mavericks_Core_Technology_Overview.pdf
15. http://www.mentor.com/embedded-software/nucleus/
16. http://www.mentor.com/embedded-software/nucleus/power-management
17. http://www.embecosm.com/
18. http://software.intel.com/en-us/avx
19. http://www.arm.com/products/processors/technologies/neon.php
20. http://www.imgtec.com/mips/mips-simd.asp
21. http://download-software.intel.com/sites/default/files/m/8/a/1/8/4/36945-319433-011.pdf
22. http://www.arm.com/products/processors/technologies/dsp-simd.php
23. http://www.mips.com/products/architectures/application-specific-extensions/mips-dsp-ase/
24. http://software.intel.com/en-us/articles/a-guide-to-auto-vectorization-with-intel-c-compilers/
25. http://developer.amd.com/tools-and-sdks/cpu-development/amd-open64-software-development-kit/
26. http://www.arm.com/products/tools/software-tools/ds-5/arm-compiler.php
27. http://software.intel.com/en-us/articles/intel-intrinsics-guide
28. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0491i/CHDFIJAE.html
29. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/BABHGFDF.html
30. http://developer.amd.com/tools-and-sdks/cpu-development/amd-core-math-library-acml/
31. http://developer.amd.com/tools-and-sdks/cpu-development/libm/
32. http://software.intel.com/en-us/intel-ipp/
33. http://software.intel.com/en-us/intel-mkl
34. https://developer.nvidia.com/gpu-accelerated-libraries
35. http://www.nag.co.uk/numeric/numerical_libraries.asp
36. https://developer.nvidia.com/category/zone/cuda-zone
37. http://ispc.github.io/
38. http://www.arm.com/products/processors/technologies/vector-floating-point.php
39. http://developer.android.com/guide/topics/renderscript/index.html
40. http://developer.android.com/guide/topics/renderscript/compute.html#filterscript
41. http://www.arm.com/products/processors/technologies/biglittleprocessing.php
42. https://developer.nvidia.com/openacc
43. http://www.khronos.org/opencl/
44. http://hsafoundation.com/
45. http://code.google.com/p/aparapi/
46. http://developer.amd.com/tools-and-sdks/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/bolt-c-template-library/
47. http://developer.amd.com/tools-and-sdks/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/
48. http://www.codeplay.com
49. http://lpgpu.org
50. http://docs.oracle.com/javase/specs/jvms/se7/html/
51. https://source.android.com/tech/dalvik/
52. http://www.ecma-international.org/publications/standards/Ecma-335.htm
53. http://www.allinea.com/products/map/
54. https://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/AnalysingCPUUsageinYourOSXApp/AnalysingCPUUsageinYourOSXApp.html#//apple_ref/doc/uid/TP40004652-CH16-SW1
55. http://www.arm.com/products/tools/software-tools/ds-5/streamline.php
56. http://software.intel.com/en-us/finding-embedded-problems-using-vtune-amplifier-for-systems
57. http://www.lauterbach.com/frames.html?country=gb%3fhome_g.html
58. http://www.ultrasoc.com/?p=1002
59. https://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/PerformanceTuning/PerformanceTuning.html
60. https://developer.apple.com/wwdc/videos/
61. http://software.intel.com/en-us/energy-efficient-software
62. http://software.intel.com/en-us/forums/power-efficiency
63. http://software.intel.com/sites/default/files/m/d/4/1/d/8/Energy_Efficient_Software_Guidelines.pdf
64. http://www.intel.com/content/dam/doc/white-paper/energy-efficient-platforms-2011-white-paper.pdf

Building a community of innovators in the development and application of EEC technologies

tinyurl.com/eecsig

Special Interest Group

Energy Efficient Computing

Energy Efficient Computing Special Interest Group

Bailey House
4-10 Barttelot Road
Horsham
RH12 1DQ

+44 (1403) 251 354

tinyurl.com/eecsig

@eecsig