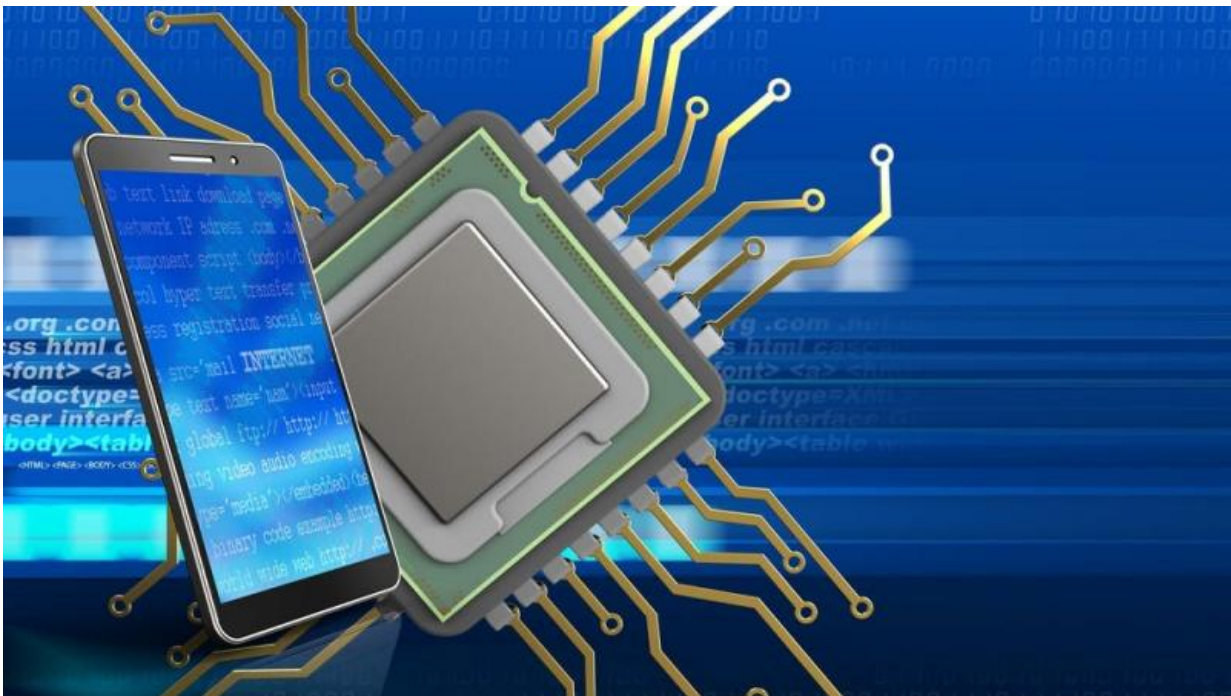# Never mind the iPhone X, battery life could soon take a great leap forward

September 14 2017, by Alexander Brownlee And Jerry Swan

Credit: Mmaxer

Another suite of Apple iPhones, another media frenzy. Much has been written about the $999/£999 iPhone X, the demise of the home button, the "face ID" function, wireless charging and so on. Somewhere down the list of improvements was extra battery life, at least for the iPhone X, thanks to its new souped up A11 bionic processor.

Apple says the new device will charge for up to two hours more than the iPhone 7, suggesting 14 hours of internet use, for instance. Battery life on the iPhone 8, on the other hand, appears to be about comparable with its predecessor. Wireless charging, to which Apple is late to the party, makes no difference to the amount of power devices can store.

Improvements to batteries are usually a key part of smartphone launches, as you would expect for one of the major specifications on which consumers judge new devices. Samsung had much to say on this subject when it launched the Galaxy Note 8 last month – albeit less about extending battery life than ensuring no repeats of the flaws in Note 7s that made them prone to catch fire.

Yet several decades into the mobile computing revolution, even the best products are still relatively limited in how long they can function on a single charge. The original iPhone was good for eight hours of internet browsing, for example, so Apple's devices have only advanced modestly in ten years.

So far, manufacturers have tended to focus on improving battery technology, packing more and more energy into less and less space. Those fiery Galaxy Note 7s were a cautionary tale of what can go wrong when this energy gets released as heat.

Manufacturers also look to improve other mobile hardware that consumes energy – including the display, WiFi, GPS and the central processing unit (CPU). The new iPhones' improved CPUs and OLED screens have made them more battery friendly, for example.

But one area that has received surprisingly little attention is the energy consumed by software, or rather the energy consumed by the CPU when running particular software. Neither Samsung nor Apple seemed to make any noises in this direction with their latest launches, but this emerging

field could make a major difference to how long we need to charge our devices in future.

Software sap

Decades ago, when computers were thousands of times slower, developers would hand-tune code to near perfection to squeeze out every last drop of performance. But as software has become more complicated – thanks to new features, improved user experience and so forth – this stopped being possible.

Software development is now several layers removed from the raw binary machine code that the CPU deals in. Developers also rely on libraries of existing code because it would take too long to build each instruction from scratch every time. Both changes reduce duplicated effort and greatly speed up development time. But the final code often contains parts that are redundant in a particular app, or it could be improved with more efficient tailor-made segments.

Developers often try to mitigate these disadvantages by making their code run as fast as possible, which in theory reduces energy consumption. Yet this doesn't always work in practice, since some instructions are more power-hungry than others and can end up neutralising the benefits.

The net result is that the energy consumption of software has increased considerably over the years. Nobody much cared until the last decade or so, since most software ran on machines that were mains powered. This has changed with the rise of mobile devices – while mounting concerns about the links between electricity consumption and climate change have added extra urgency.

The AIs have it

There is another reason why developers were slow to address this problem, which is that the energy consumption from each piece of software was very difficult to measure. This is because each device's configuration is different. Energy use can change depending on whether a program has run before, or whether other programs are running.

Lately, however, there have been advances. They involve using machine learning to estimate energy use by analysing particular lines of code or software components, and referencing energy data from other programs running on many other devices. Do this well and you can get the computer to do the hard part: search for alternative software designs that make the software more efficient.

Welcome to search-based software engineering. It can be as simple as finding redundant code that can be skipped or fine-tuning the configuration, or it can extend to making changes to existing source code. Our own work has looked at both choosing alternative software components from existing libraries and generating new parts of code from scratch. We even managed to find and repair several hundred bugs in Hadoop, a very widely used software framework.

Our vision is that these search-based methods for improving energy efficiency will be incorporated into what is known as the "compiler" stage, when human-readable computer code is converted into the zeroes and ones the machine understands. These searches would happen automatically and developers wouldn't need to think about them – their code would be efficient out of the box.

There is still a long way to go, it should be said. The main difficulty is getting the estimates of software energy consumption right, especially for lots of different devices at the same time. But the potential over the next five years looks exciting. We were able to show a 40% to 70% reduction in CPU energy use for a couple of specific tasks, and it's not

inconceivable that this could be replicated over all running software.

Combined with better batteries and more improvements to hardware performance, such as [zero energy screens](#), we could be talking about serious gains on battery life. In future, the leading manufacturers may no longer be talking up incremental improvements to battery life capabilities – instead they could be adding many hours and maybe even days.

This article was originally published on [The Conversation](#). Read the [original article](#).

Provided by The Conversation