Why TOPS is Bottom of Metrics

Tony King-Smith

As the rush to provide hardware acceleration for AI continues to increase, a growing number of vendors are using TOPS as a key metric for characterizing their solution's performance capabilities. However, other metrics can provide much more meaningful information on the performance of NN accelerators. The white paper details AImotive's approach to benchmarking the performance and efficiency of aiWare. We give an in-depth explanation of why TMAC/S complemented by a range of realistic workload benchmarks are a more reliable measure and examine the unique challenges and limitations of benchmarking neural network accelerators.

Keywords: aiWare; neural network acceleration; benchmarking; TOPS; TMAC/S; hardware platforms; efficiency; performance

Overview

It seems everyone is developing not only new software but hardware for AI these days. Whether it's for the latest autonomous car or the next high-performance cloud data center, NN hardware acceleration is a hot topic.

However, history appears to be repeating itself in the early stages of this technology wave. Many analysts and companies are trying to characterize these blocks by one single parameter such as TOPS. And the benchmarks being used are proving to be increasingly inadequate, using out of date and overly simplistic input sizes and network depths.

This is proving a growing headache for companies who are relying on NN acceleration suppliers to not only provide a solution, but also to educate them on the key technical issues. What do the claimed figures really mean?

It also means that sometimes more innovative and efficient solutions are overlooked, because they don't claim the right headline figures compared to the irrivals, despite delivering superior results in real-world applications. Thus, if we're not careful, using the wrong benchmarks can stifle innovation, causing better solutions to appear worse than they are under real-world workloads. There are a few key reasons for the widespread usage of TOPS for example:

- Marketing and media people are forever drawn to big numbers;
- Much of CNN computation is dominated by MACs, however TOPS counts all computation elements – whether they're used or not;
- When it comes to use of silicon area and power consumption, CNNs are dominated by memory and data flow, not just computation. TOPS only tells you about peak computation capacity.

System designers and architects for SOCs, electronic subsystems and complex systems such as automotive AVs (Autonomous vehicles) need more data than TOPS in order to make the design decisions that impact long design cycles, and extend product life times. The purpose of this white paper is therefore to explain what the key characteristics of metrics being used for procuring NN acceleration solutions are, and how to relate these to your own application and key system criteria.

Why do people use TOPS?

When trying to describe complex engines like CPUs, GPUs or NN accelerators, semiconductor companies are inevitably drawn to using one or two simple headline numbers to characterizeperformance. For CPUs it was MIPS (Millions of Instructions Per Second). For GPUs it was MPix/s (Millions of Pixels per Second) or later thenumber of ALUs. In each case, end users of these devices found these numbers were a very poor indicator of how their end application would actually perform. This created demand for more realistic numbers based on credible, well-defined workload-based benchmarks recognizedby the broader industry as giving a much better indication of how the devices would perform under real-life, sustained operating conditions.

However, it appears we are doing it again, by quoting neural network (NN) acceleration in terms of TOPS: Tera (one million million, or one thousand billion) OPerations per Second.

Why? The quick answer is because people love a big number! And since it's hardware, it needs to be one we can justify. The simplest solution is to choose the smallest unit of performance, and say how many we have of it. Works great for marketing and the media – it just has a very poor correlation to doing the actual job, which ends up misleading application developers. That's not good for any of us – we need to do a better job, just as we have done for gaming, floating point computation and user interfaces.

When TOPS is quoted for a NN accelerator, all this figure tells you is the total number of computation elements implemented in hardware multiplied by their clock speed. What it doesn't tell you is whether the computation element is actually needed, or how much of the time it will have all the data it needs to do its task, or indeed do any real task. It is more a measure of what you could achieve if everything was perfect: data was always immediately available, infinite power was available, there were no memory constraints, the algorithm was perfectly mapped to the hardware etc. Also, it inevitably makes no allowance for the need for the hardware platform to do any tasks other than the computation itself.

Therefore, it tells you only the theoretical maximum performance that could be achieved. Problem is, that's rarely achieved in practice, as you must take into account at least which operations areactually being used most of the time, and the efficiency at which these are being performed.

While TOPS means something to a silicon implementation engineer, it can be a misleading figure for a software applications engineer or hardware systems engineer. That's the problem: these metrics give the application developer or systems designer a very poor indication of the available performance they can actually use in a real-world application. So, the result is that you specify a chip claiming a certain TOPS capability that you think delivers the performance you need, then find yourself spending months or even years scaling your application down to fit the real performance. Due to myriad practical limitations such as memory bandwidth, on-chip bus and memory subsystem priorities, software overheads, host CPU overheads, and interface bandwidth limitations.

Step 1: TMAC/s – a better measure, at least for Convolution-centric tasks

With many computation-heavy tasks using CNNs, the workloads are actually dominated by convolution – as much as 95% of some AV workloads for example. Therefore, AImotive approaches the problem by first focusing on how to measure the efficiency of just the convolution workload by using TMAC/s: Tera Multiply Accumulates Per Second.

Since all the other functions used to complete the CNN workload (e.g. pooling, activation, concatenation) are orders of magnitude less significant in terms of overall workload, we ignore them for the purposes of benchmarking a CNN. That's why we quote our Convolution LAMs in TMAC/s at a certain clock frequency.

TMAC/s is defined by the native mathematical resolution used in the computation itself – often INT8 (8-bit fixed point integers) for many automotive applications. However, the unit can also be used for FP16 or FP32 (16-bit or 32-bit Floating Point) resolutions – it just needs to be identified in the metric itself.

Step 2: Evaluate CNN Efficiency

TMAC/s is an important measure for any semiconductor design team, as these are the crucial engines of the computation that must work efficiently. But how they are connected, and how data flows between them, defines how well the engine works as a whole. And since the convolution engines will dominate any CNN implementation, their underlying performance is what defines the area and power consumption as well as efficiency under sustained workloads.

Therefore, we need to identify how well the data flows through the NN accelerator under sustained computation, to see how much of the engine is being used. This is where we measure "Efficiency":

the percentage of TMAC/s theoretical performance that is used under a real workload. We aim for higher than 90%–95% sustained efficiency, i.e. continuous operation, not just a few isolated frames of data.

For example, in AImotive's aiWare v1 Test Chip, performance is quoted as 1.6 INT8 TMAC/s at 400MHz. This tells the application developer exactly how much convolution processing capacity is potentially available from the core if everything is working perfectly. We then aim to get the efficiency of the core under a wide range of real workloads as close to 90%–95% of that theoretical maximum TMAC/s figure.

This, however, is still only telling you the performance of convolution operations of a single network. So, we need to do more.

Step 3: Use Realistic Workload Benchmarks

In a real AV application, a number of different NNs are used to do different aspects of the computation, from front-end perception and classification through to back-end trajectory planning and control. What really matters is how well a NN accelerator executes such a set of real NNs under real-life operating conditions.

For AImotive, we use our own complete aiDrive software stack to help us understand efficiency on our aiWare hardware IP, complemented by more representative NNs such as ResNet50, GoogleNet etc. on real data. By constantly evaluating the efficiency of our aiWare-based hardware in this way, we can ensure we invest our engineering effort into optimizing the right things for the right reasons.

As workloads grow ever-more complex, we can see how our growing set of aiDrive NN workloads impacts the architecture of aiWare. We are constantly re-evaluating aiWare's ability to cope with an ever-expanding range of use cases, identifying any bottlenecks or

performance barriers, and refining our architecture accordingly. As a result, or latest generation of aiWare NN hardware accelerators can handle a wider range of input sizes, with more network depth and greater range of layer configurations than ever before.

Public networks help broaden applicability

When comparing NN accelerators from various vendors, it's important to utilize networks created by a variety of sources. That's why we have derived our customized benchmarks from a variety of sources that are regularly reviewed and updated, including some popular publicly available networks such as Googlenet, Resnet and vgg16, alongside some of our own aiDrive workloads. AImotive's performance evaluation methodology is published regularly by AImotive on its website (see https://aimotive.com/share/benchmark/aiWare.pdf).

We run these benchmarks using a range of input resolutions from VGA to 8Mpixels, i.e. those needed for typical multi-HD camera scenarios. Since our aiDrive AV software solution follows a vision-first approach – using heterogeneous sensor fusion to augment camera data with high resolution Lidar, radar and other sensor data – we anticipate ever higher demands for a growing number of 2k x 2k and higher resolution sensor inputs.

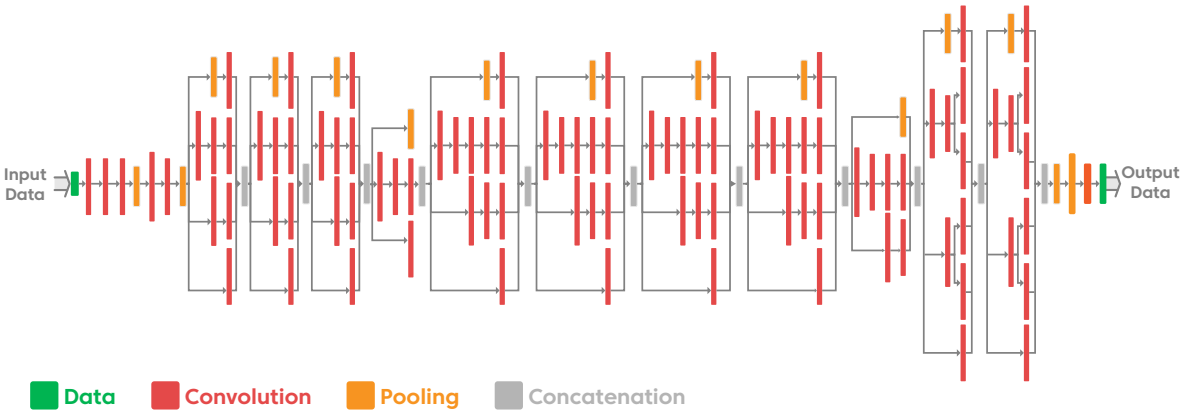


Figure 1: Our own networks and popular publicly available networks offer a variety of complexity and depth. We use them with much higher resolution inputs than commonly used today; reflecting what we believe the crucial parameters for the future generation AVs will be.

Not only cameras but also Lidars and other sensors, such as TOF. That's why we believe so strongly that it is essential to use higher resolution inputs when benchmarking NNs for AV applications.

Since the most demanding computation challenges lie around achieving high efficiencies for higher resolution inputs, it's also important for a NN accelerator to handle the widest possible range of workloads, as every mm2 on an SoC must be as useful as possible. We then measure the actual efficiency of our engine and quote that figure, using a combination of simulation, FPGA and (from Q4 2018) real test silicon results. In this way, our customers can be confident that our claimed performance can be achieved under sustained workloads in a real "always-on" automotive environment.

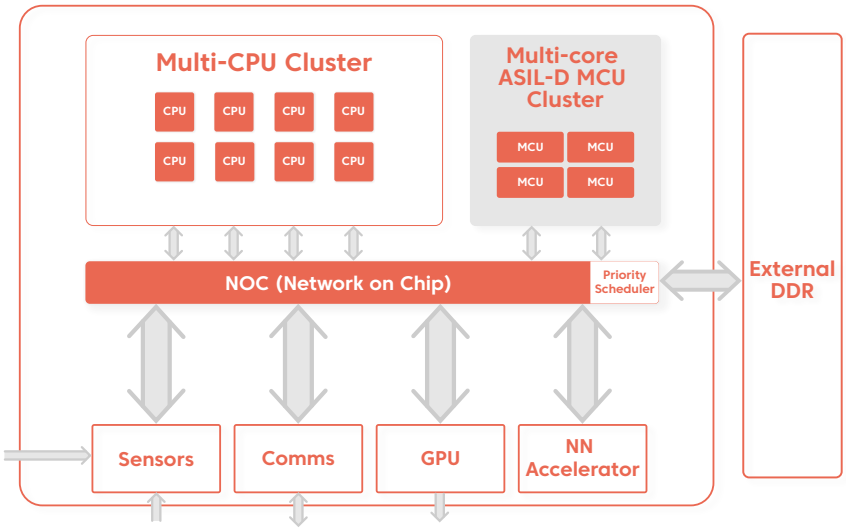
This approach is key when developing NN acceleration technology for vehicles targeting production in 4–8 years' time, not the test vehicles being deployed today.

Older NNs use older resolutions

In our experience, many people tend to use publicly available benchmarks as a starting point. The problem with this for automotive workloads is that many of these come from an era defined by the winners of the enormously influential DARPA competitions 6–12 years ago. These tended to use lower resolution Lidars with input sizes of 224 x 224 or less, or other low-resolution sensors.

The compute power available to these teams in the days of the DARPA Grand Challenge was an order of magnitude smaller than what we are using today, meaning networks had to be less complex than the ones we use today. And the sensors they used were relatively simple Lidars – nothing like the more advanced cameras or higher resolution radars and Lidars rapidly appearing on the market.

Figure 2: Tomorrow's automotive SoCs already place massive demands on memory subsystems and NOCs from multiple high-bandwidth CPU, GPU and other subsystems – before adding higher bandwidth NN acceleration.



Thus, not surprisingly, evaluations relying on these older benchmarks can be misleading when used to scope future generation solutions targeting future generations of vehicles using higher resolution sensors, and more of them.

CNNs are dominated by memory and data

When designing today's complex SoCs, many high-performance blocks are competing for on-chip and off-chip memory resources. The latest mobile phone SoCs are excellent examples of the state of the art, combining 4–8 core 64-bit CPU clusters, GPUs with hundreds of ALUs, plus multi-gigabit modems and peripherals. There is a huge amount of data already flowing through these SoCs – just consider how much external memory bandwidth is required to support the continuous operation of an 8-core 64-bit CPU cluster alone clocking at 2GHz.

The crucial design challenge here is the on-chip bus structure – these days usually implemented as a "NOC" (Network on Chip). This is a complex, multi-lane high bandwidth subsystem where multiple channels of high-bandwidth data can be moved from place to place on chip. These are complemented by increasingly sophisticated

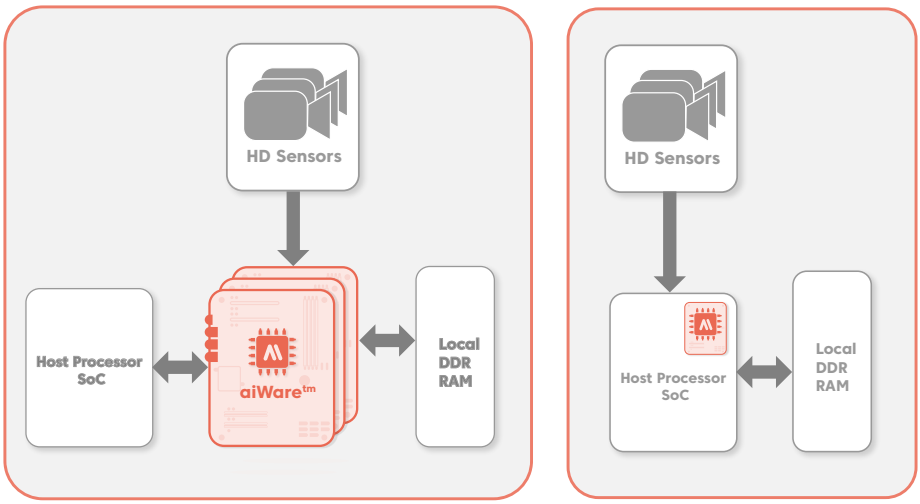
cache structures within the CPUs, GPUs and elsewhere, all aimed at maximizing the efficient use of every single memory read or write cycle.

Therefore, SoC architects are already struggling with adding NN acceleration to these chips, which is a memory-centric, high data rate dependent function. Putting such additional demands on a shared memory subsystem will either demand extremely complex, high-power memory subsystems, or will risk making the overall SoC seriously inefficient due to data bottlenecks on the SoC's NOC and external memory subsystem.

Efficient embedded NN inference needs dedicated architectures

We expect the demands of NN acceleration to grow ever higher, as input sizes increase, sensor fusion demands more high-resolution sensor inputs to be combined at various levels of abstraction. The overall reliability of AV must grow an order of magnitude above that of humans. At the moment, these numbers are extremely difficult to reconcile for SoC architects and system designers.

Figure 3: The aiWare NN accelerator is designed for use in either on-chip or separate chip configurations, supports scalability by using multiple chips in parallel, and supports direct data streaming from sensors.



That's why we've developed a new hardware architecture for our aiWare dedicated hardware engine for acceleration of CNNs. And that's why we've also developed new benchmarking methodologies to make sure we're solving the right problems, not just coming up with impressive numbers.

The aiWare architecture is highly scalable not only on-chip, but also using one or more dedicated accelerator chips, to create extremely high performance yet extremely efficient solutions. When tackling the challenge of such a broad range of workloads, it was vital that AImotive's designers understood in great detail how NN data flows and execution cycles operate, and crucially how data moves from one calculation to the next.

The result is an architecture that can be highly optimized according to key parameters such as depth of networks, size of inputs, and available area. Since the area is dominated by memory (not computation elements), aiWare allows designers to trade off on-chip memory for efficiency, using off-chip memory to help minimize the impact of reducing memory size.

By having a benchmarking strategy based on real-life workloads, we can help our customers to make the best choices for optimal hardware platforms, delivering the power, cost and performance targets needed.

Conclusions

Application developers must be wary of theoretical measures such as TOPS – they need much more information to make informed decisions. We all need to learn from the mistakes of the past by realizing the biggest number does not necessarily deliver the best performance. We need to ensure that the right set of metrics are available to properly evaluate the ability of hardware solutions to deliver sustained real-time performance in real production ECUs for hour after hour. Designers deserve more than just one big figure!

It is vital that we all provide system architects and applications engineers throughout the engineering ecosystem and supply chains with the tools they need to make the right decisions for their stakeholders. If we do this right, we will build trusted, long-term relationships between us and our customers that are essential for a successful AV future.