AI Accelerators: Transforming Scalability & Model Efficiency

*"AI Accelerators: Transforming Scalability & Model Efficiency"* in 10 key bullet points:

1. AI is rapidly evolving across industries, driven by real-world applications. **AI accelerators**, specialized hardware for AI inferencing, are central to enhancing speed and efficiency.
2. Just like cars evolved from carriages, **AI systems are diversifying**—there’s no one-size-fits-all model. Accelerators help tailor solutions to specific AI tasks and use cases.
3. The **AI stack** consists of hardware (infrastructure), models, and software (governance, security). Accelerators reside at the **hardware layer**, optimized for intensive matrix math (e.g., tensor ops).
4. AI accelerators leverage modern compute architectures (e.g., **parallelism, optimized memory bandwidth**) and can either have dedicated or shared memory with other processors.
5. Accelerators can be located **on-die, elsewhere in the system, or externally connected**. Each can be **customized for different workloads**, enabling heterogeneous AI systems.
6. There’s a distinction between **traditional AI (ML/DL)** and **generative AI (LLMs)**—the former is cheaper and faster, while the latter is more complex, powerful, and resource-intensive.
7. Selecting the **right-sized model** is key. Large models aren't always optimal—they may hinder system performance and sustainability. Matching model size to task ensures cost-effectiveness and efficiency.
8. **Model effectiveness** is a function of **accuracy, performance, and use case risk tolerance**. Different applications (e.g., fraud detection vs. document summarization) require differently sized models.
9. In a **credit card fraud detection** example, a fast, low-power ML model runs first using an on-die accelerator. If uncertain, the task escalates to a more powerful GenAI model on a different accelerator.
10. With millions of AI transactions per day, **hardware accelerators enable flexible, scalable, and efficient AI**, supporting **multi-model (ensemble) AI** at enterprise scale.

AI has captured the world's imagination. Ideas in the space for creating and improving applications are expanding in every direction, and at the heart of these ideas are AI accelerators, hardware accelerators, which is specific hardware designed for the inferencing and AI workloads.

(00:35) These enable transformation because they enable faster and more efficient processing.

(01:25) Hardware accelerators are really great at helping with the right size solutions for AI and to really explain the intersection between hardware acceleration and AI, let's take a simplified look at the AI stack., at its fundamental level, at the bottom, we have infrastructure that's hard to spell. It's hardware.

(01:55) On top of that, of course, we have our models. You've all heard of these. And then at the top left, but most definitely not least, is software, management of the tasks, but also governance, and governance is making sure that the AI continues to behave correctly, without bias, ethically, as it evolves and develops and goes forward.

(02:23) In addition, the software manages security, making sure that the AI model data and the data that's fed in and out of those models remains private and secure. Where do the accelerators fit in this stack? Well, you might have guessed already, they are part of the hardware. So the accelerators are here. What specifically is an AI accelerator? Again, this is purpose-built hardware, silicon, that's built and optimized for the high-compute matrix mathematics that is necessary to do AI transactions.

(03:03) So the linear algebra, the tensor calculations, this hardware is laid out specifically, the silicon is designed to optimize and do that faster so that the inferencing answers can happen faster, with reduced power consumption and a smaller physical footprint compared to a general purpose piece of hardware that's designed to do everything and therefore has to cover all bases.

(03:26) Accelerators do use all of the computer architecture, modern techniques like parallel processing and optimize memory bandwidths and they interact with memory, they have a large memory component. It's a specific hardware engine that does the compute, and that either has its own memory or it shares memory with other processors in the system.

(03:50) Let's draw a picture of a system to really get a sense of where these accelerators might exist in an actual piece of hardware. So this is a computer system. It could be a server or a mainframe, and like most today, let's make it an SMP so it has a variety of processor chips on it. These AI accelerators, they could be located anywhere within the system.

(04:19) For example, you might wanna have an on-die accelerator close to your processing compute. You could have that on one or more of your processor compute engines You could have a AI engine somewhere else in the system, not necessarily on a processor chip, but still within the box. Additionally, since servers interact with the external world through industry standard IO protocols, you could actually attach an AI accelerator externally to the box, and those accelerators could come with their own compute as well as their own memory.

(04:57) The other thing to note about these hardware accelerators, and this is what really provides that flexibility and scale that we really need in computing today to enable all of these use cases and these ideas that people are coming up with in the AI industry, is the... concept that each one of these AI engines doesn't have to be the exact same design.

(05:19) Your on-chip accelerator, for example, could be optimized in a different way for a different type of workloads than the one that's inside the system, which may or may not be different from the ones that you attach to the box. So there's definitely a heterogeneous system of hardware and within hardware, and heterogenous system of accelerators that are becoming available so that we can meet these growing AI needs.

(05:45) To really see how these accelerators now interact with the rest of the stack, let's take a couple of minutes to just think about models for a second, and I think, you know, we all I think are familiar now with our traditional AI models, right? Machine learning, you know, deep learning. These are relatively fast.

(06:09) They're relatively small in terms of model size, and they're pretty good at coming up with, I'll say, suggestions. They can't give me life advice necessarily, but they can offer suggestions. Gen AI coming out in recent years, more complex, right? Another level, right, they're generative, right, versus predictive, right.

(06:39) These are things like our LLMs and simulators and things, right these are much larger models in general, but they have, you know, they take more energy to run, they are more costly, but they're also more powerful in terms of delivering accuracy and doing more complex things. They can... dare I say it, maybe not give life advice, but in some cases give, you know, advice.

(07:04) Do some advisement. Because we have this new, more complicated one, it doesn't mean that we should just abandon the traditional AI because it does have a lot of value. Most notably, it's best and it's small and it is less expensive, right? So if we can get a good answer out of that, it is worthwhile to use it.

(07:29) Really, right sizing your model is where you want to be when you are doing AI. If you have a AI task that's this size and your model is little, you run the risk of, you might get an answer fast, but that answer might be wrong, or it might come back with more risk than your application can tolerate. So you might say, okay, I'll err on the side of caution, and I'll just always use a huge model, but you're not being cost effective and you're being sustainable when you do this.

(08:05) In addition, you run the risk, if you happen to be attempting to do something in the same compute space as your running system, you run the risk of kneecapping other simultaneously running workloads on your box if you use too large of a model. So ideally, what we wanna go for and try to do is to right size the model to the task that we're doing.

(08:32) Ok, how do hardware accelerators help with that? To understand that, it helps to think about the model the way hardware thinks about the model, which is that the model really, at the end of the day, is data, and when you think about hardware and how you optimize hardware, what we think about generally is performance, a common metric of performance in the hardware world, familiar to me, I'm a hardware person, If you couldn't tell already, is latency or we call it response time, right? How fast do you respond?

(09:09) And when you're measuring response time as a function of size, right, the size of your working set, the size of your data, the of your model, basically, generally, the bigger that model gets, the worse your response time gets, right. It takes longer and longer to get the answer out there, right? If we try to use a similar graph to measure performance of AI, it's not quite as linear, because what we're trying to do really when we're measuring how well AI performs is we really say, well, how effective is it?

(09:46) And if you try to draw a graph like this, it would end up looking a little bit more confusing, and you kind of have to think, well what's going on there? What's going on there is response time, hardware performance is not the only factor when you're talking about measuring the success or the effectiveness of the AI.

(10:09) It's really accuracy as well as performance. And the variable that's hidden here is a third dimension. Bear with me as I try to draw a third-dimension. Which we touched on before I hinted at it. It's real your use case, and that green line is really coming out in the third dimension here. So the questions you want to be asking when you're trying to decide what model to use are, how much risk do I need? If I'm a credit card fraud detection system, you know, I have a different risk profile, I have different inputs, I have levels of

(10:49) tolerance than I do if I'm batch document summarization process, right? So those factors drive the need for models that align with specific use cases. So at the end of the day, what's happening is your use case is dictating your model size, and if you have knowledge of your model size, and you have access to these hardware accelerators, then you can leverage these hardware accelerator and run the model that you need on the optimized hardware accelerator for the task at hand.

(11:22) Let me illustrate all this with an example. It's a common example, but it's one of my favorites because it's very personal and it resonates, I think with everybody, it's credit card fraud detect, right? I wanna use my credit card online all the time, and I want to hit pay now and I wanted to go through as long as it's me.

(11:39) If it's not me, I want it to stop. So that task is fairly complex. It also has a high dependence on response time because I'm gonna navigate away if it takes too long to use this site, I'll go someplace else, right. So let's illustrate how we might use AI models and hardware accelerators to optimize for that use case.

(12:08) So we have a task here, nd really, that task is, is it fraud? All right.I want to get the answer as fast as possible. So maybe I'll start by using a. ML DL model. This will be fast, it'll have a low response time, and it'll have a good forecast of whether or not this is fraud or not. Maybe it'll be 80% I'm making these numbers up, successful.

(12:41) That model, if I'm doing something like this, I will probably, if have it, want to use a hardware accelerator that's on the same processor chip as the transaction workload that's running. The reason is that transaction workload already has the data that needs to go into that model, and the model's a relatively small size, so you can pull that data into the cache right there in the processor chip and enable in real time actually getting, you know, your suggestion of whether or not this is fraud, yery quickly.

(13:12) Then I'd say 80% of the time, that answer comes back. They're like, yeah, this is you. This is a valid purchase. I'm gonna approve it. And then you get your toy that you bought and nobody loses any money and everything is good, but maybe 20% of time, the model says, hmm, I'm not sure. Some things look legit, but some things look suspicious.

(13:36) We need to take a deeper look at this. At that point, then, you could decide that now you know it's worth the investment of going into a more expensive gen AI model, and that model requires a lot more memory because it's a much bigger model. So you wouldn't be using the AI engine in this case that you had on your die.

(13:58) You might want to use an AI engine that's on a different die that's not running a performance critical workload, or maybe one that's in a different place on the system, or maybe one attached and sharded through the IO peripherals that has its own memory. Based off of that, if you have that flexibility, then you can decide which to use and the hardware can use that, and then at the end of that transaction, it can come out and say, no, it was good.

(14:23) It's okay, it turns out that that was the right one. So the idea here is using more than one model, this concept of multi-model AI, or ensemble AI, can really be supported and enabled and optimized by having these hardware accelerators that are optimized for different model types, and then, when you think about when you're doing AI, the last thing to think about here is, you know, we're not doing one of these a day, right? We're doing millions of these day, many of them at the same time, right.

(14:55) So you want to have all of these processing AI engines running all at the time, doing different things, and then the next second or millisecond, It's doing a completely different model with a completely difference task. So hardware accelerators give you the flexibility to do that at scale, and because of that, they intersect well with the AI stack as we know it today and are really starting to really be used to ensure that we're continuing to expand and meet business need by deploying scalable, efficient and secure AI to meet today's business problems and tomorrow's.