GIGAOM

AI at the Edge: A GigaOm Research Byte

Credit: Jakarin2521

By Byron Reese

Sponsored by arm

GIGAOM

AI at the Edge: A GigaOm Research Byte

01/31/2019

Table of Contents

1 Summary

Artificial intelligence (AI), primarily in the form of machine learning (ML), is making increasing inroads into our lives. There are several primary reasons for this:

- The rapidly-increasing capability of computers used to build and train ML models.
- Greater data-capturing ability across the compute environment, often in the form of inexpensive sensors embedded in everyday consumer, business, and industrial products.
- The development of new algorithms and approaches that improve the accuracy of ML applications.
- The creation of software toolkits that make building and training ML applications substantially easier, and therefore less expensive.

In addition to these four ‘truths’, there are two other factors which are often overlooked that are equally as important in bringing AI into our lives. These factors are not about where AI’s are built and trained, but where they are deployed and used:

- A reduction in cost, and increase in performance, of chips doing AI inference “at the edge.”
- The development of middleware allowing a broader range of applications to run seamlessly on a wider variety of chips.

It is these final two developments that will allow AI to enhance our lives in countless new ways and enable AI in our pockets, cars, houses, and a host of other places. This report explores these latter two factors, ignoring how AI is built and trained while focusing on the methods by which AI impacts our lives. It explores the natural architectural migration of AI from central, powerful computers where an AI algorithm or application may have historically been built, trained, and used, to an edge model. In the edge model, the AI compute happens either on a user device or somewhere in the network stack beneath the traditional cloud, perhaps on an edge server.

This leads to a new AI model that is match-fit for what is to come: building and training, which will mainly continue on ever-more powerful (and power-hungry) cloud-based computers, and inference. Inference will be performed at the device edge, or close to it. It is where the AI will run on ever-more-powerful (but less power-hungry) chips. This foundational change in the AI architecture will be the single biggest driver in the advance of AI at scale.

This new architecture has several advantages over a highly centralized or cloud model, specifically:

- More scalable
- Faster
- Lower cost
- More secure
- Lower power

There are tradeoffs in this approach, including the fundamental constraints of the chipset and future upgradability. Further, there are still several outstanding questions about this shift that only time will answer:

- How far to the edge will AI compute finally be pushed to?
- Which chip design price/performance combinations will prove to be the most popular?
- How disposable will the chips of the future be?

It should be noted that there are use cases where this model of centralized training and edge inference will not be appropriate; cases where decision latency and power consumption are not factors. One can imagine, for instance, that a large and expensive medical device might ship data back to a central location to be processed and analyzed on a time scale (perhaps measured in seconds or minutes) that would be unacceptable in another application, such as a self-driving car. We discuss these exceptions as well.

The final part of this report briefly explores the societal impact of this change in architecture. Winston Churchill once said, “We shape our tools and then the tools shape us.” We are the generation that is shaping the digital tools of tomorrow, and it is worth reflecting on how they might shape us in return.

2 In the Beginning

In the summer of 1956, a group of academics met at Dartmouth to discuss what would become a new field in computing. One of them, a young mathematician named John McCarthy, gave the field a name, artificial intelligence. He would later come to regret the term, feeling it set the bar too high.

Marie des Jardins, Dean of the College of Organizational Computational and Information Sciences at Simmons University, picks up the story from there:

> “From the 1960s, when AI was first identified as a problem area in the field until probably the 1980s, AI was basically one CPU on one computer doing a particular task in the service of some larger application. A lot of progress was made in that time, particularly in formulating intelligent activity as searching through a space of solutions to problems.
>
> “Starting in the 1980s and coming up into the 2000s, robotic hardware became more advanced and we started to do more advanced things on the physical platform. AI became more embedded with sensors and actuators in the environment. So that meant two things: one is the algorithms became more complex and distributed and also more complex in their inputs, their outputs, and their sort of real-time nature. So now we’re not just trying to find a solution to a problem. We’re trying to do it in real time, in a complex environment, to control a complex system.
>
> “Then in the last 10 to 15 years, we’ve had this resurgence of interest in neural networks that are much more sophisticated mathematically, and much larger in scale than the neural networks of the ’60s and the ’80s.”

It is a testament to the relative youth of AI that this short introduction largely brings us up to the present. Indeed, it is developments that have happened over the last five years that have predominantly brought AI into its own. This rapid pace of change in our knowledge of AI, and what we can do with it, has grown so fast that if all development stopped immediately, it would still take decades for us to implement in all the places it could be leveraged today. But, needless to say, AI advances are not stopping or even slowing down; They are accelerating. AI, and specifically a single technique called machine learning (ML), have major implications. This single idea, this technique of studying data with computers and looking for patterns to predict the future, is creating trillions of dollars of new wealth.

How big will ML get? No one knows. No one even knows how big it is now. The best we can do is look at proxies that suggest its rate of growth. We see mid to high double-digit annual growth in ML patent applications, investments in projects and companies, hyper-scale data centers, and dozens of corollary indicators. Make no mistake, this is a technology juggernaut, a once-in-a-generation technical leap forward. Yes, AI and ML may be hyped, but the hype is warranted.

3 To the Cloud… and Beyond

The cloud has enabled a revolution in AI scalability; fueling the rise of ML by allowing model training to be done in an affordable and sustainable way. If you had a large data set and needed to study it, you could spin up a thousand machines to brute force your way through it. And when you were done, you let them go. No fuss, no muss, no bother.

Will ML live in the cloud? Yes and no.

First the “yes.” The training of all machine learning models we can imagine today will likely remain in the cloud because training requires massive storage, massive compute, and extended runtimes. In addition, data sets are growing dramatically, as is the compute used to process them. According to OpenAI, in the last six years the amount of compute utilized in the largest AI training models has increased exponentially, doubling every hundred days; meaning that, since 2012, the level of compute has increased 300,000-fold. Given the fact that this exponential growth is far from over, the cloud is an optimal platform for ML and is essential to delivering on its heady promises.

Next, the “no.” If the cloud is so powerful and scalable, why won’t ML live there forever? To answer this requires us to divide ML into two distinctly different parts: *Training* and *inference*.

Training is when a dataset is studied and the findings are built into an algorithm of some kind. For example, a million cat photos need to be studied to learn how to identify future images of cats. This process requires massive data and is compute intensive.

Inference is the application of a learned algorithm to a real-world problem. Training is about the past, inference is about the present. Teaching the model to identify a cat is training while taking a single photo and using that algorithm to discern whether it is a cat or not is inference. For an individual photo, this is not terribly computationally rigorous, but it can become so if, for instance, you wanted to count every photo on the web to see how many cats there are.

Stop for a moment and think about this. Where should you perform the calculations to do inference? The short answer is that for many – if not most – applications, inference in the future will be done at the edge, that is, where the data is collected. This will have a huge impact on how ML will develop.

4 Why the Edge?

So why will AI inference largely move to the edge? I put this question to Pete Warden, an engineer at Google and the technical lead of the TensorFlow mobile team, which is responsible for deep learning on mobile and embedded devices. His succinct answer hit the nail on the head:

“Fundamentally because that’s where all the data is.”

Steve Roddy, the VP of Special Projects inside Arm’s Machine Learning Group agreed when I spoke with him:

“The edge is the next stage of the evolution of AI technology because of the physical constraints, the cost constraints, and the practical constraints of running all AI applications in the cloud. It simply doesn’t make sense to send all the bits for things like video and audio streaming to the cloud and back down for every situation, every endpoint.

“Applications that people will engage within real-world products such as controlling home devices or providing driver assistance in a car, all of those applications are running on the edge and many will require real-time responses. Any delay from bouncing information to the cloud and back could be a problem.”

Jem Davies, a VP, Fellow, and the General Manager of the Machine Learning Group at Arm, underlined that:

“The inference from the machine learning is most useful when it’s done right next to the test data. If, for example, you’re trying to recognize things in video, the chances are that camera is out there in the wild. It’s not connected directly to a hyper-scale data center.

“So, if we see an explosion of machine learning inference moving to the edge, there are very sound reasons for that happening. Yes, it’s next to the data that you’re trying to test, but it’s also down to the laws of economics, the laws of physics, and the laws of the land.

“The physics argument is simple. There isn’t enough bandwidth in the world to transmit everyone’s video images to a cloud center in Seattle, have it interpreted, and the results sent back. You would break the internet. And there are economic implications with that scenario too. Google said if everybody used their Android Voice Assistant for three minutes per day they would have to double the number of data centers they owned. That would cost a fortune and it’s impractical. On top of the cost, there’s also latency issues. Say my new car is trying to identify a pedestrian in front of it. Should it be asking the cloud to interpret sensor data or making those decisions with its onboard systems? The answer is obvious – it’s best done at the edge. It’s fast and not prone to network interruptions.

“And finally, there’s privacy and security – the laws of the land. People are increasingly worried about their personal data being spread all over the internet, and rightfully so. So, if I can have my personal data interpreted on my device and not channeled to a public cloud, I’d feel much more comfortable.”

If you distill the elements Davies discusses down, we are led to five major advantages of performing AI inference at the edge:

1. Scalability across a vast number of edge devices
2. The speed of local decision making
3. Lower power consumption at the edge
4. Security, retaining data on the device
5. Cost, minimizing data center and transport

In contrast to the five edge inference benefits, there are two inherent tradeoffs:

1. Upgradability, keeping up with firmware updates and new learning
2. Feedback data will not be available or as frequent

Let us look at each one of these in more detail, starting with the benefits.

4.1 Scalability

No one knows how many computers there are in the world, but it is estimated that they consume approximately 10% of all global power. By extension, we also do not know how many processors there are running in chips. Doing some rough counting though, we can include the billions of smartphones currently in use, each with eight or so processors. Throw in smart appliances, cameras, cars, buildings, and all the rest, and we get to at least 100 billion CPUs capable of doing inference. The number is, of course, likely to be far higher. But with edge-based AI inferencing, it does not matter. You can take that CPU number to 200B, 500B, a trillion, 10T, or 100T. Everything scales accordingly as the AI compute is mainly being performed on each individual device. If each device had to communicate back to a data center for all AI-based decisions, the planet would be engulfed in racks, wires, and air-conditioned data centers.

4.2 Speed

Even in a world where data can be sent to giant supercomputers at the speed of light, inference on the edge is much, much faster. Why? Because electricity is so slow. If a self-driving car is trying to decide whether the object ahead of it is a deer or a stray grocery bag, its systems do not have the luxury of waiting for a break or avoid decision from a data center many miles, even continents, away. The inference must occur at the edge.

Sri Chandrasekaran of IEEE elaborates:

“We can’t overlook the importance of latency. AI at the edge will allow for faster data transfer, which will, in turn, benefit the many industries AI touches, especially industrial IoT and automotive. These industries benefit from AI at the edge because the machines and automobiles must be able to understand many different aspects at once. Sending data to the cloud and back is not only inefficient, but it is also less secure and much slower, ultimately leading to a decrease in productivity and reliability. In addition, it is important to consider power requirements for AI at the edge, especially when it comes to industrial IoT. In order to successfully accomplish AI at the edge, these devices must have the appropriate computing power.”

4.3 Power Consumption

Pete Warden of Google explains the power consumption advantages of edge inference:

“The underlying issue is that it takes massive amounts of power to send data over the air. Even over a few meters like Bluetooth or Wi-Fi, and it takes a very small amount of power – orders of magnitude less- to do arithmetic on that data when it’s sitting on a device. So, for most places where you can have devices, there is a very strict power constraint. Either they’re not wired in or they’re on battery power and they must last for years. You don’t want people changing batteries all the time, or relying on energy harvesting like solar or thermal.”

4.4 Security

The advantages of the edge for security are clear. If data has to travel hundreds of miles to feed an inference decision and then ship the answer-back, there are too many ways for that data to be compromised.

And there is an additional wrinkle to this. Warden explains:

“One of the things I like about doing things on the edge is I like objects that don’t have the ability to easily talk on the network, because then essentially everything is an air gap. So, if you don't have the ability to connect via a network, then at least you removed multiple ways that these devices can be misused."

4.5 Cost

The cost advantage is the net result of the prior four benefits. Building inexpensive chips to do inference at the edge costs less than building the apparatus and infrastructure needed to be in constant contact with the cloud.

However, not all AI will happen at the edge for the simple reason that the edge, by necessity, is often a power-constrained environment. A small, low-power-consuming processor simply is not as beefy as the CPUs at a data center. This constrained environment results in drawbacks in upgradability and feedback data.

4.6 Upgradability

In the cloud model, algorithms can be constantly iterated and improved. Every day the system can get better, and therefore any device connected to it constantly improves. In the edge model, this is less often the case. If your TV remote can understand simple voice commands or your digital camera automatically adjusts its exposure settings automatically, those capabilities are usually set in stone (or in this case silicon). Perhaps firmware updates can provide occasional enhancements, but that is about it. The counter to this is if the device performs adequately when it is shipped, presumably it will perform adequately for its expected life.

Arm's Steve Roddy addresses this further:

> "One management overhead that we need to tackle is the management of the *updating* of those ML applications. The training and the retraining of the networks. So, for example, if your home automation system has security cameras that unlock the door when you walk up, it needs to learn family members' faces and IDs. It needs to get corrected and improved over time so it becomes more reliable in accepting the right people and rejecting strangers. It must do this whatever people are wearing too – thick winter coats, hats, sporting a few days' beard growth.
>
> "There's a reinforcement learning, incremental learning, that needs to occur by transferring that data safely and securely from the endpoint up to a cloud for updating, for retraining networks, for updating the machine learning implementations. Whether it's the individual with the home security system or a retailer who has a thousand stores and 10,000 security cameras, you want to update the protocols for what they're identifying and analyzing in the cameras.
>
> "That whole system of routinely updating the firmware in embedded devices is something

> that needs to be fine-tuned. We know how that works with our phones and our PCs. But managing those devices in an IoT world is much more distributed, much more varied – it's a system management challenge that needs to be solved in the industry."

4.7 Feedback Data

What is different about the cloud-based inference model is there are constant streams of new potential training data coming back to the cloud to be used to refine the inference model. This data has the added benefit of being real-world data about how the system is used. In the edge model, this feedback loop does not happen. I brought this up when chatting with Pete Warden:

> "Yes. That's totally fair. But that must be hard anyway, just because the physical power costs of sending data back are so high. Meanwhile, try to think about gathering data for training as a very separate thing from running this in production. So, we try to be very explicit about when we're gathering data for training, to make sure it's very clear that it is a specialized thing that's happening."

5 Exceptions to the Edge Model

If inference at the edge is to be the overwhelming norm, what are the exceptions? One can imagine several. Steve Roddy explains:

> “When looking at the difference between doing inference on the edge and doing inference in the cloud there are going to be many applications that fall into both categories. Certainly, there are dozens – if not hundreds – of applications that *do* require real-time response on-device, at the edge, whether that be safety-critical applications in cars or simple things like asking your home assistant to turn the lights on when you walk into a room. You don’t want to be dependent upon a link back to the cloud.
>
> “However, there are applications that do better in the cloud because they require massive compute and there aren’t latency or time-sensitivity issues. Take the example of a medical X-ray that must be read today, but whether the radiologist gets to it in one minute or five minutes does not matter. If it goes to the cloud for pre-processing before a human looks at it, that is perfectly fine. You might also imagine a farmer taking aerial drone pictures of their crops to analyze moisture and growth rate indicators analyzed. That data could be dealt with through a cloud application, as adjusting the watering system doesn’t have to happen in split seconds, it just has to happen in a reasonable timeframe.
>
> “So, we foresee growing numbers of ML applications that fit in both categories. Some are obviously time, cost, or power-run on the edge, but there are many others that can be done in the cloud. The AI world will be built on both scenarios and the explosion of ML techniques across a wide range of applications means there’s plenty of room for growth for each.”

6 Where Does the Edge Begin and End?

This is a point of contention. Consider a self-driving car. Is it the car as a whole? The electronics control unit? The embedded sensors in the bumpers? How far down can meaningful edge AI be pushed? I raised this question to Pete Warden:

> "Wherever the data is being gathered – if you don't have to then send that data somewhere else, and if instead you can kind of turn that data into something actionable and useful – then that's where the processing should happen.
>
> "Looking at the accelerometer data that's on a bumper, if you want to understand whether there's been a damaging collision or just a piece of grit bouncing off the road, you shouldn't need to send that data over to the car's central processor to figure that out. You should just have a sensor on the bumper that knows."

Arm's Steve Roddy weighed in as well:

> "There are still many unanswered questions about how the compute capability necessary to run edge AI applications will actually be deployed in the real world. Whether that be the automobile, where you may have multiple distributed systems or one centralized system; or a factory, where you could have wireless connections from every smart device back to a centralized CPU or you could have distributed computing. The same thing with cities: how dense will the compute be in the 5G networks, versus how centralized? So, there's a lot of uncertainty as to where that compute will live, both short-term and long-term. One of the goals that Arm has is to create the systems in the middleware layers that will allow applications to move seamlessly from different compute models as those models evolve over time."

7 How Far Can it All Go?

Where does the edge end up? A car today has hundreds of smart sensors, a building, tens of thousands. Will your pen be an AI device on the edge? Your smart toothbrush? Your coffee cup? The answer is an unequivocal yes. As Marie des Jardins of Simmons University explains:

> “I think everything’s going to become a little bit smart, and I think very fast, we’re going to stop thinking of it as smart. I’m looking at my coffee pot [right now]. Your coffee pot is just going to know that it should make coffee for you; we’re not going to think of that as being smart anymore. It’s just going to be what coffee pots do. Just the way thermostats just keep the temperature where you want it. At one point that was amazing.”

Pete Warden weighs in:

> “I think sensors that are able to do smart things, like voice interfaces or accelerometers, are going to become so low-power and so cheap that they’re going to be everywhere. For example, with your pen… if you actually have an accelerometer in it and you are able to record … and then upload that to some other device …it might be a potentially useful application of machine learning.”

8 Compatibility and Middleware

In the introduction to this report, I mentioned two oft-overlooked factors driving AI and ML:

- The decrease in cost and increase in power of chips capable of performing AI inferencing “at the edge.”
- The development of middleware allowing a broader range of applications to run seamlessly on a wider variety of chips.

The future will be a world with countless chipsets, produced by dozens or hundreds of manufacturers. Each may be used in different devices and in different combinations. There will be dozens, or hundreds, of development environments for creating applications to run on those chips, in those applications.

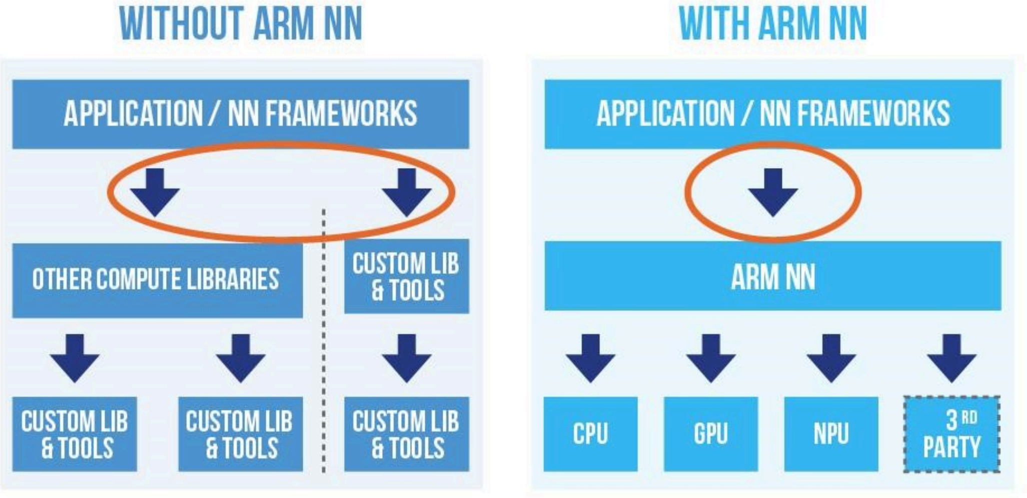
In fact, it is not so much the world of tomorrow, as the world of today. It is already like the wild west in terms of edge ML. First, there are a number of neural net development frameworks in the cloud, including Caffe, Caffe2, MXNet, ONNX, PyTorch, and TensorFlowLite. And there are a number of multiple, intermediate forms, such as FlatBuffers, NNEF, ONNX, and Protobuf. In addition, there are a variety of inference engine interfaces such as Android NNAPI, MACE, and PaddlePaddle.

These examples only touch the tip of the AI foundation iceberg. And the situation will become vastly more complex. ML applications are so specialized that a single, uber toolkit that is all things to all people will almost certainly not emerge. So, how can we prevent this from descending into a chaos of technical incompatibility and inoperability? The answer is in standards. What is needed is a widely accepted, and used, middle layer acting as a common platform between software and hardware. Developers could code to the middleware layer, and it would handle operating in the wide variety of environments. Linux-based Arm NN (neural net) is such a solution. Arm’s Steve Roddy explains:

> “One of the challenges of designing AI, or an ML application for devices on the edge, is that application developers want to build their application once and port it to different end products, for different variations of products – and they simply want it to work. If, for example, you’re building an app for an Android phone, you want to build it once and put it on the Android store and not worry about which phone it’s being deployed on.
>
> “So, the challenge for widely deploying AI applications is to build an abstraction layer – a middleware layer – to simplify things for app developers, allowing an app to run many implementations. Whether you’re looking at an IoS or Android phone, there are millions of applications, and millions of developers wanting to run their apps on billions of phones. Simplifying that and creating an abstraction layer as Arm has done with Arm NN, addresses a key market pain point. If every one of those million developers had to optimize

> implementations for each platform, it would quickly become an insurmountable amount of work."

Here is how it works:



Partitioning and targeting of NN workloads falls to application developer / system developer

Arm´s heterogeneous SW platform sees all available processors as compute targets, under a single, unified interface. Tools work across the platform as a whole

What makes the Arm solution so viable is it is open-source after it was donated to Linaro so that it could be freely used and modified by other organizations. Qualcomm and Google are just two of the many companies now integrating Arm NN into their workflows.

9 Conclusion

As the price of computing continues to fall, the number of computation devices will continue to increase exponentially. While that cannot persist indefinitely, one thing seems certain: Computational devices powered by artificial intelligence will touch our lives in almost every conceivable way. The power, security, and speed requirements of these devices necessitate inference be performed at the edge, where the data is collected. This will enable an ever-more-common way of scaling the digital devices that will come to play a role in our lives.

AI has promised many things for many years. At the root of it all is a simple belief that if data drives decision-making, that is good for everyone. AI provides the way to make data decision-making better and, if it is applied across all compute, there may be nothing out of our reach. We are the first generation to glimpse these possibilities and know they can become real. To quote a famous Star Trek Captain, we now just have to “Make it so.”

10 About Byron Reese

Byron Reese is the CEO and Publisher of GigaOm. He is also the author of a recent book on AI called *The Fourth Age: Smart robots, conscious computers, and the future of humanity*. He hosts two AI podcasts, *Voices in AI* and *The AI Minute*. He resides in Austin, where GigaOm is headquartered.

11 Contributors

The following individuals were interviewed for this report and quoted extensively throughout.

Steve Roddy, Arm – Vice President, Special Projects, Machine Learning Group.

Pete Warden, Google – Technical Lead of the TensorFlow Mobile team, responsible for deep learning on mobile and embedded devices.

Marie des Jardins, Simmons University – Dean of the College of Organizational Computational and Information Sciences.

Sri Chandrasekaran, IEEE Standards Association – Senior Director of Standards and Technology in India.

In addition, I also reference notable quotes from an episode of Voices in AI that I recorded with Jem Davies, a Fellow at Arm.

I want to thank Arm for supporting this report and giving me access to the folks included herein. Arm is a natural partner, as they are helping to drive the future of AI. Their chips provide the intelligence in approximately 85% of all mobile computing devices and their architecture forms the bedrock of the internet of things (IoT). More recently Amazon also announced it was using Arm IP in new server chips that will handle increasing network data workloads. To date, Arm's partners have shipped 130 billion processors, with a current deployment rate of about 20 billion chips a year.

12 About GigaOm

GigaOm provides technical, operational, and business advice for IT's strategic digital enterprise and business initiatives. Enterprise business leaders, CIOs, and technology organizations partner with GigaOm for practical, actionable, strategic, and visionary advice for modernizing and transforming their business. GigaOm's advice empowers enterprises to successfully compete in an increasingly complicated business atmosphere that requires a solid understanding of constantly changing customer demands.

GigaOm works directly with enterprises both inside and outside of the IT organization to apply proven research and methodologies designed to avoid pitfalls and roadblocks while balancing risk and innovation. Research methodologies include but are not limited to adoption and benchmarking surveys, use cases, interviews, ROI/TCO, market landscapes, strategic trends, and technical benchmarks. Our analysts possess 20+ years of experience advising a spectrum of clients from early adopters to mainstream enterprises.

GigaOm's perspective is that of the unbiased enterprise practitioner. Through this perspective, GigaOm connects with engaged and loyal subscribers on a deep and meaningful level.