



How posits are revolutionizing computing

Over the hundred years that they have been used, floating-point numbers have become so entrenched that replacing them would be like "boiling the ocean." But Dr. John Gustafson believes he has done just that.

By Rebecca Tan

For the first 30 seconds, all seemed to be going well with the first test launch of Ariane 5, a 700-million-dollar rocket developed by the European Space Agency. The boosters fired up right on cue, sending up plumes of smoke as the rocket followed its planned trajectory skyward and engineers on the ground heaved a collective sigh of relief. Just ten seconds later, Ariane 5 self-destructed, scattering smoldering debris across the launch site.

The costly error was eventually traced to a flaw in the on-board guidance system: a bug in the code applied by a calculation error. This error, according to applied physicist and mathematician Dr. John Gustafson, could have been completely avoided in a new system of arithmetic that he proposes.

WHEN FLOATING-POINTS FLOP

To understand what went wrong, you'll first need to understand how computers deal with numbers. Floating-point arithmetic is a system, developed over a hundred years ago, that represents real numbers by converting them into three component parts: sign, exponent and significant digits. According to the rules determined by the Institute of Electrical and Electronics Engineers (IEEE), floating-point numbers are typically 32 or 64 bits long, with one bit representing the sign, eight or 11 bits giving the exponent, and the remaining bits for recording the significant digits (significand).

The advantage of floating-point numbers is that they allow users to deal with both very large and very small numbers by simply varying the exponent component. However, there is an inherent trade-off: The more bits you use to increase the range of numbers that can be represented (dynamic range), the fewer bits there are left to record the number accurately.

For the first 30 seconds, all seemed to be going well with the first test launch of Ariane 5, a 700-million-dollar rocket developed by the European Space Agency. The boosters fired up right on cue, sending up plumes of smoke as the rocket followed its planned trajectory skyward and engineers on the ground heaved a collective sigh of relief. Just ten seconds later, Ariane 5 self-destructed, scattering smoldering debris across the launch site.

34756488; 7936274; 8925892; 2863882; 9397192; 9327967; 7235014; 2848822; 4574392; 1455977; 0150330; 4403742; 2093762; 9795915; 9908658; 7463667; 2378705; 1782966; 5474673; 3149568; 5654528; 5970309; 3269049; 7799140; 6695010; 4205685; 9204199; 1602876; 1646776; 1504452; 11071285; 7229295; 4587687; 3228824; 4426666; 8855909; 5722208; 5020454; 5187445; 9495630; 2799007; 3668501; 3932512; 3932512; 3341988; 0183484; 6092861; 3261193;



Dr. John Gustafson
Visiting Professor
A*STAR Computational Resource Centre

Together around this, programmers use more bits to represent floating-point numbers to get more precision for applications that require it, even if it's more computationally expensive to do so.

"In the case of Ariane 5, they were measuring speed with a 64-bit number but feeding it into a guidance system that used 16-bit numbers," explained Dr. John Gustafson, a visiting professor at Singapore's Agency for Science, Technology and Research (A*STAR)'s Computational Resource Centre, who also holds a joint appointment with the National University of Singapore.

"Ariane's programmers specified the dynamic range and chose poorly. What I'm pitching is to let computers manage their own accuracy and dynamic range to automatically avoid that kind of mistake."

THE POWER OF POSIT THINKING

The solution Gustafson has proposed involves entities named *numms*, which are the equivalent of adding a bit at the end of a number to indicate that a result is or is not exact, like writing the value of π as 3.14... indicating that there are more digits. Instead of making the mistake of saying that π is 3.14 exactly, "This allows computers to distinguish between when an infinite value is the correct answer and when it is in fact an overflow error like the case of Ariane 5. Recently, he created a form of *numm* called a *posit* that is designed as a drop-in replacement for floating-point numbers.

What are posits?

Posits are a new type of data that, like floating-point numbers, round if a number is inexact, but do so in a way that is simpler and more accurate.

By introducing a component which Gustafson calls the "regime," posits allow the exponent and significant to vary in size, where bits saved by having a smaller exponent can be used to describe the significant with more precision. This unique feature gives posits accuracy where it is needed, at everyday numbers close to one where most calculations are likely to be done.

As a result, posits have a wider dynamic range than floating-point numbers of the same bit length and yet are more accurate. Gustafson has also found that in every case tested so far, posits can even get more accurate answers with fewer bits.

"That's what really motivated me—making the operands smaller and faster instead of making the transistors smaller. In effect, that would help us to continue Moore's Law for free, and more importantly, help us get to exascale computing without breaking the power budget," he said.

"If you put my arithmetic on a chip, it will take up less space in silicon and use less energy. And because things like our mobile devices and video games are constrained by how much you can get out of a watt, changing the arithmetic could have a big impact on battery life and miniaturization."

The best part about posits, however, might be the fact that they work like floating-point numbers and therefore can immediately replace them without needing to change the way things are currently done.

"Think about LED light bulbs: you can go to the store to get a light bulb that you can actually screw into the same socket as an incandescent light bulb and it immediately starts saving you energy, producing more light with less power. I believe posits can do the same for computation," Gustafson added.

"But why are we walking when we could run so much faster?"

YOU CAN BOIL THE OCEAN

This ability to seamlessly replace floating-point numbers was the final piece of the puzzle that Gustafson has been working on for the last 35 years. When he presented an older version of the posit format to his colleagues while still a director at Intel Labs, he was told that it was futile to attempt to displace floating-point operations, that it was like trying to "boil the ocean."

"Essentially, they told me, 'Yes, there are better ideas out there, but look at all this existing infrastructure: we're stuck with it,'" shared Gustafson, an inaugural Gordon Bell Prize winner who is no stranger to revolutions in computing, having pioneered parallel processing at a time when it was dismissed as impossible.

"I relish these situations. I've often played the maverick and been a disruptive influence on the computing business, and posits have all the same feel that parallel processing went through in the 1980s," he said. "The ocean is starting to boil!"

Although the idea for posits was 35 years in the making, 32 of those years were false starts. Gustafson said, "Everything I tried

had something that broke. It's only in the last three years while I've been here in Singapore that I've made the breakthrough, realizing that I could do a drop-in replacement for floating-point arithmetic."

RUN, DON'T WALK

In fact, Gustafson believes that posits will radically change the way we think about supercomputers. Currently, the world's most powerful computers ranked on the TOP500 list are judged based on how many floating-point operations per second (FLOPS) they are able to perform. Even though double precision 64-bit numbers are used, none of the machines are actually able to give the exact answer to the benchmark problem.

"They might get 0.999999 or 1.0001 instead of 1 and they'll say, 'Close enough.' With posits, I'm able to use one-fourth as many bits and get the exact answer," Gustafson said. "If accuracy were a criteria, using posits would destroy the rules of the LINPACK benchmark used to evaluate the TOP500."

"Using floating-point numbers is like taking part in a walking race, where they impose rules that prevent people from getting both feet off the ground. But why are we walking when we could run so much faster? Posits would let us do it the right way and go really fast."

Interestingly, posits are also particularly suitable for training neural networks, one of the hottest topics in artificial intelligence research. While typical computers take over a hundred clock cycles to calculate the sigmoidal curves used in neural network training, using posits could make the calculations a hundred times faster and use just eight bits.

VIVA LA REVOLUCIÓN

So while you may not even be aware when your computer switches from floating-point to posit arithmetic, you will be bound to feel the difference in terms of speed and battery life.

"I once calculated that people have put over 100,000 man-years of time into playing Angry Birds. If we converted that game from 32-bit floats to 16-bit posits, it would save thousands of barrels of oil!" Gustafson said, laughing.

But far beyond games, he predicts that posits will make a profound impact wherever calculations are necessary. The revolution, however, might take some time.

"It takes about ten years for an idea to get used everywhere in practice. This is the pattern and it doesn't change no matter whether it is 1950 or 1980 or 2017," Gustafson contended.

"Singapore has been a wonderful place for me to develop these ideas, and I'd love to see Singapore become known as the place where arithmetic changed forever and revolutionized the world," he said. "And it will be." ■

COMING SOON

Gustafson is leading the formation of the Centre for Next-Generation Arithmetic at the Agency for Science, Technology and Research (A*STAR) in Singapore. Look out for our follow-up feature in the next issue!