The Slow Winter

JAMES MICKENS

James Mickens is a researcher in the Distributed Systems group at Microsoft's Redmond lab. His current research focuses on web applications, with an emphasis on the design of JavaScript frameworks that allow developers to diagnose and fix bugs in widely deployed web applications. James also works on fast, scalable storage systems for datacenters. James received his PhD in computer science from the University of Michigan, and a bachelor's degree in computer science from Georgia Tech. mickens@microsoft.com

According to my dad, flying in airplanes used to be fun. You could smoke on the plane, and smoking was actually good for you. Everybody was attractive, and there were no fees for anything, and there was so much legroom that you could orient your body parts in arbitrary and profane directions without bothering anyone, and you could eat caviar and manatee steak as you were showered with piles of money that were personally distributed by JFK and The Beach Boys. Times were good, assuming that you were a white man in the advertising business, WHICH MY FATHER WAS NOT SO PERHAPS I SHOULD ASK HIM SOME FOLLOW-UP QUESTIONS BUT I DIGRESS. The point is that flying in airplanes used to be fun, but now it resembles a dystopian bin-packing problem in which humans, carry-on luggage, and five dollar peanut bags compete for real estate while crying children materialize from the ether and make obscure demands in unintelligible, Wookie-like languages while you fantasize about who you won't be helping when the oxygen masks descend.

I think that it used to be fun to be a hardware architect. Anything that you invented would be amazing, and the laws of physics were actively trying to help you succeed. Your friend would say, "I wish that we could predict branches more accurately," and you'd think, "maybe we can leverage three bits of state per branch to implement a simple saturating counter," and you'd laugh and declare that such a stupid scheme would never work, but then you'd test it and it would be 94% accurate, and the branches would wake up the next morning and read their newspapers and the headlines would say OUR WORLD HAS BEEN SET ON FIRE. You'd give your buddy a high-five and go celebrate at the bar, and then you'd think, "I wonder if we can make branch predictors even more accurate," and the next day you'd start XOR'ing the branch's PC address with a shift register containing the branch's recent branching history, because in those days, you could XOR anything with anything and get something useful, and you test the new branch predictor, and now you're up to 96% accuracy, and the branches call you on the phone and say OK, WE GET IT, YOU DO NOT LIKE BRANCHES, but the phone call goes to your voicemail because you're too busy driving the speed boats and wearing the monocles that you purchased after your promotion at work. You go to work hung-over, and you realize that, during a drunken conference call, you told your boss that your processor has 32 registers when it only has 8, but then you realize THAT YOU CAN TOTALLY LIE ABOUT THE NUMBER OF PHYSICAL REGISTERS, and you invent a crazy hardware mapping scheme from virtual registers to physical ones, and at this point, you start seducing the spouses of the compiler team, because it's pretty clear that compilers are a thing of the past, and the next generation of processors will run English-level pseudocode directly. Of course, pride precedes the fall, and at some point, you realize that to implement aggressive out-of-order execution, you need to fit more transistors into the same die size, but then a material science guy pops out of a birthday cake and says YEAH WE CAN DO THAT, and by now, you're touring with Aerosmith and throwing Matisse paintings from hotel room windows, because when you order two Matisse paintings from room service and you get three, that equation is going to be balanced. It all goes so well, and the party keeps getting better. When you retire in 2003, your face is wrinkled from all of the smiles, and even though you've been sued by several pedestrians who suddenly acquired rare paintings as hats, you go out on top, the master of your domain. You look at your son John, who just joined Intel, and you rest well at night, knowing that he can look forward to a pliant universe and an easy life.

Unfortunately for John, the branches made a pact with Satan and quantum mechanics during a midnight screening of "Weekend at Bernie's II." In exchange for their last remaining bits of entropy, the branches cast evil spells on future generations of processors. Those evil spells had names like "scaling-induced voltage leaks" and "increasing levels of waste heat" and "Pauly Shore, who is only loosely connected to computer architecture, but who will continue to produce a new movie every three years until he sublimates into an empty bag of Cheetos and a pair of those running shoes that have individual toes and that make you look like you received a foot transplant from a Hobbit, Sasquatch, or an infertile Hobbit/Sasquatch hybrid." Once again, I digress. The point is that the branches, those vanquished foes from long ago, would have the last laugh.

When John went to work in 2003, he had an indomitable spirit and a love for danger, reminding people of a less attractive Ernest Hemingway or an equivalently attractive Winston Churchill. As a child in 1977, John had met Gordon Moore; Gordon had pulled a quarter from behind John's ear and then proclaimed that he would pull twice as many quarters from John's ear every 18 months. Moore, of course, was an incorrigible liar and tormentor of youths, and he never pulled another quarter from John's ear again, having immediately fled the scene while yelling that Hong Kong will always be a British territory, and nobody will ever pay $8 for a Mocha Frappuccino, and a variety of other things that seemed like universal laws to people at the time, but were actually just arbitrary nouns and adjectives that Moore had scrawled on a napkin earlier that morning. Regardless, John was changed forever, and when he grew up and became a hardware architect, he poured all of his genius into making transistors smaller and more efficient. For a while, John's efforts were rewarded with ever-faster CPUs, but at a certain point, the transistors became so small that they started to misbehave. They randomly switched states; they leaked voltage; they fell prey to the seductive whims of cosmic rays that, unlike the cosmic rays in comic books, did not turn you into a superhero, but instead made your transistors unreliable and shiftless, like a surly teenager who is told to clean his room and who will occasionally just spray his bed with Lysol and declare victory.

As the transistors became increasingly unpredictable, the foundations of John's world began to crumble. So, John did what any reasonable person would do: he cloaked himself in a wall of denial and acted like nothing had happened. "Making processors faster is increasingly difficult," John thought, "but maybe people won't notice if I give them more processors." This, of course, was a variant of the notorious Zubotov Gambit, named after the Soviet-era car manufacturer who abandoned its attempts to make its cars not explode, and instead offered customers two Zubotovs for the price of one, under the assumption that having two occasionally combustible items will distract you from the fact that both items are still occasionally combustible. John quietly began to harness a similar strategy, telling his marketing team to deemphasize their processors' speed, and emphasize their level of parallelism.

At first, John's processors flew off the shelves. Indeed, who wouldn't want an octavo-core machine with 73 virtual hyper-threads per physical processor? Alan Greenspan's loose core policy and weak parallelism regulation were declared a resounding success, and John sipped on champagne as he watched the money roll in. However, a bubble is born so that a bubble can pop, and this one was no different. John's massive parallelism strategy assumed that lay people use their computers to simulate hurricanes, decode monkey genomes, and otherwise multiply vast, unfathomably dimensioned matrices in a desperate attempt to unlock eigenvectors whose desolate grandeur could only be imagined by Edgar Allen Poe.

Of course, lay people do not actually spend their time trying to invert massive hash values while rendering nine copies of the Avatar planet in 1080p. Lay people use their computers for precisely ten things, none of which involve massive computational parallelism, and seven of which involve procuring a vast menagerie of pornographic data and then curating that data using a variety of fairly obvious management techniques, like the creation of a folder called "Work Stuff," which contains an inner folder called "More Work Stuff," where "More Work Stuff" contains a series of ostensible documentaries that describe the economic interactions between people who don't have enough money to pay for pizza and people who aren't too bothered by that fact. Thus, when John said "imagine a world in which you're constantly executing millions of parallel tasks," it was equivalent to saying "imagine a world that you do not and will never live in." Indeed, a world in which you're constantly simulating nuclear explosions while rendering massive 3-D environments is a world that's been taken over by members of a high school A.V. club. The members of a high school A.V. club

The Slow Winter

lack the chops to establish a global dictatorship, if only because doing such a thing would require them to reduce their visits to Renaissance festivals, and those turkey legs need help to be consumed in the style of a 15th century Italian aristocrat.

John was terrified by the collapse of the parallelism bubble, and he quickly discarded his plans for a 743-core processor that was dubbed The Hydra of Destiny and whose abstract Platonic ideal was briefly the third-best chess player in Gary, Indiana. Clutching a bottle of whiskey in one hand and a shotgun in the other, John scoured the research literature for ideas that might save his dreams of infinite scaling. He discovered several papers that described software-assisted hardware recovery. The basic idea was simple: if hardware suffers more transient failures as it gets smaller, why not allow software to detect erroneous computations and re-execute them? This idea seemed promising until John realized THAT IT WAS THE WORST IDEA EVER. Modern software barely works when the hardware is correct, so relying on software to correct hardware errors is like asking Godzilla to prevent Mega-Godzilla from terrorizing Japan. THIS DOES NOT LEAD TO RISING PROPERTY VALUES IN TOKYO. It's better to stop scaling your transistors and avoid playing with monsters in the first place, instead of devising an elaborate series of monster checks-and-balances and then hoping that the monsters don't do what monsters are always going to do because if they didn't do those things, they'd be called dandelions or puppy hugs.

At this point, John was living under a bridge and wearing a bird's nest as a hat. Despite his tragic sartorial collaborations with the avian world, John still believed that somehow, some way, he could continue to make his transistors smaller. Perhaps the processor could run multiple copies of each program, comparing the results to detect errors? Perhaps a new video codec could tolerate persistently hateful levels of hardware error? All of these techniques could be implemented. However, John slowly realized that these solutions were just things that he could do, and inventing "a thing that you could do" is a low bar for human achievement. If I were walking past your house and I saw that it was on fire, I could try to put out the fire by finding a dingo and then teaching it how to speak Spanish. That's certainly a thing that I could do. However, when you arrived at your erstwhile house and found a pile of heirloom ashes, me, and a dingo with a chewed-up Rosetta Stone box, you would be less than pleased, despite my protestations that negative scientific results are useful and I had just proven that Spanish-illiterate dingoes cannot extinguish fires using mind power.

It was at this moment, when John had hit the bottom, that he discovered religion.

John began to attend The Church of the Impending Power Catastrophe. He sat in the pew and he heard the cautionary tales, and he was afraid. John learned about the new hyper-threaded processor from AMD that ran so hot that it burned a hole to the center of the earth, yelled "I've come to rejoin my people!", discovered that magma people are extremely bigoted against processor people, and then created the Processor Liberation Front to wage a decades-long, hilariously futile War to Burn the Intrinsically OK-With-Being-Burnt Magma People. John learned about the rumored Intel Septium chip, a chip whose prototype had been turned on exactly once, and which had leaked so much voltage that it had transformed into a young Linda Blair and demanded an exorcism before it embarked on a series of poor career moves that culminated in an inevitable spokesperson role for PETA. The future was bleak, and John knew that he had to fight it. So, John repented his addiction to scaling, and he rededicated his life to reducing the power consumption of CPUs. It was a hard path, and a lonely path, but John could find no other way. Formerly the life of the party, John now resembled the scraggly, one-eyed wizard in a fantasy novel who constantly warns the protagonist about the variety of things that can lead to monocular bescragglement. At team meetings, whenever someone proposed a new hardware feature, John would yell "THE MAGMA PEOPLE ARE WAITING FOR OUR MISTAKES." He would then throw a coffee cup at the speaker and say that adding new hardware features would require each processor to be connected to a dedicated coal plant in West Virginia. John's coworkers eventually understood his wisdom, and their need to wear coffee-resistant indoor ponchos lessened with time. Every evening, after John left work, he went to the bus stop and distributed power literature to strangers, telling them to abandon transistor scaling and save their souls. Standing next to John, another man wore a sandwich board that said that the Federal Reserve was using fluorinated water to hide the fact that we never landed on the moon. The sandwich board required no transistors at all. It made John smile.

When John comes home for the holidays, you're glad that he's back, but you miss the old twinkle in his eye. Your thoughts wander to your own glory days thirty years ago, when Aerosmith mistook young John for a large Xanax tablet and tried to trade him for a surface-to-air missile that could be used against anti-classic rock regimes. Oh, how you laughed! The subsequent visit by Child Protection Services was less amusing, but that was the way that hardware architects lived: working hard, partying hard, and occasionally waking up in Tijuana to discover that your left kidney is missing and your toddler has been shipped to a Columbian arms smuggler. It was crazy, but you wouldn't change a thing. Your generation had lived so many dreams, and slain so many foes.

Today, if a person uses a desktop or laptop, she is justifiably angry if she discovers that her machine is doing a non-trivial amount of work. If her hard disk is active for more than a second per hour, or if her CPU utilization goes above 4%, she either has a computer virus, or she made the disastrous decision to run a Java program. Either way, it's not your fault: you brought the fire down from Olympus, and the mortals do with it what they will. But now, all the easy giants were dead, and John was left to fight the ghosts that Schrödinger had left behind. "John," you say as you pour some eggnog, "did I ever tell you how I implemented an out-of-order pipeline with David Hasselhoff and Hulk Hogan's moustache colorist?" You are suddenly aware that you left your poncho in the other room.