Crowdsharing Idle Processor Time

M. Brian Blake
Drexel University

Editor-in-Chief M. Brian Blake ponders the idea of “AirBnB for Available Computing Cycles.”

I was sitting on a commercial airplane when I experienced a now regular occurrence. My laptop was spinning ferociously for no apparent reason and completely ignoring the ten different operations and applications that I had invoked. At the same time, I realized my wife was using her mobile device while allowing her laptop to sit idle… Wouldn’t it be great if her processor could augment my own computer’s capacity? I wondered how many other computers on the airplane had spare processor cycles that I could exploit. But I was sure that none of them would allow me to use those cycles for free. All of a sudden, I realized that what I really needed is an “AirBnB for Available Computing Cycles.” For those who don’t know, AirBnB provides a network platform where homeowners can offer extra bedrooms and entire houses to consumers for a fee, much like hotels. “AirBnB for Computing”—what a brilliant idea! But wait, let me curb my enthusiasm. I turned to Google to see if the idea had already been conceived and implemented.

As I suspected, The Golem Project recently announced an extension to their current platform to allow consumers to share idle processor time specifically for rendering 3D computer graphics. Although they plan to extend beyond this single-use case, the first step is to allow users to distribute the CGI processing of any Blender (www.blender.org) and LuxRender (www.luxrender.net) scene over their network. They hope to extend their platform to address use cases in the areas of scientific calculations, machine learning, and other process-intensive tasks.

Although this is a potentially impactful idea with significant promise, I imagine that the implementation would be quite difficult. It is commendable that the Golem Network is attempting to constrain the complexity by focusing in one domain. Just the initiation of this idea presents the need for several new areas of investigation. It might be interesting to explore challenges with the current AirBnB process for houses and see if similar challenges extend to a future AirBnB-type approach to computing, particularly Internet computing.

Interestingly enough, my wife and I have a second home in Miami, where we have been AirBnB hosts for the past two years. Our home is a small, two-bedroom cottage in a quaint location in South Miami. For Miami, the peak rental hotel market is from December to March. At this time, single hotel rooms rent for close to $500/night. As such, AirBnB becomes popular for families looking for a less expensive option with more space, a kitchen, washer/dryer, and beach equipment. We offer our home at a relatively high price and a minimum three-night stay, but offer a gated yard, off-street parking, and many amenities such as beverages, fruit, and pastries. Although we mainly get great clientele, there are several generic AirBnB challenges that might serve as a reasonable starting point for Internet computing challenge areas.
• **Consumers who exceed the guest limit.** To reduce the probability of damage, most AirBnB hosts limit the number of guests. To enforce the rule, some homes have cameras installed at doors to see who comes and goes. Unfortunately, it is difficult to enforce guest limits during the stay. Regarding AirBnB for Computing, what if multiple users spoof their identities and overextend the provider’s resources? Researchers and engineers must create network security protocols to detect multiple users and deploy mechanisms that gracefully limit their use during the rented processor time. Moreover, when it is advantageous to have multiple users in one time slot, should the platform manage multiple users while guaranteeing their availability to computational power?

• **Consumers who damage the property.** Broken windows, scratches to walls, and other damage can discourage hosts from sharing their property with consumers. In the computing scenario, can consumers invoke operations that might damage host computers? Can they intentionally conduct denial-of-service attacks on hosts? Considering a peer-to-peer network, how do infrastructure providers, like the Golem Network, ensure appropriate operations for the entire community?

• **Consumers who cannot operate the key lockbox, electric gate, or check-in too late.** There could be challenges to the guest experience because each home is different. User interfaces must be detailed enough to allow hosts to customize their offerings but generalized enough so that guests can move from one host to the other seamlessly. Human computer-interaction will be important, particularly considering the business aspects. The movement of data is also important and would perhaps be most effective with an intermediary to store and marshal information between the consumer and host.

• **Guests who scope out the property to later burglarize it.** There are certainly challenges when opening your home to strangers. Consistent with malicious operations such as denial of service, it is also important that hosts guarantee a level of privacy when opening their hardware to unknown consumers. There certainly should be a trust network helps to certify both hosts and consumers.

• **The market.** AirBnB computing hosts want to drive maximum traffic to their offerings. As such, prices and amenities will naturally vary and fluctuate. Reputation management services further animate price. Are all processors created equal? How do you really evaluate the effectiveness of processing time from one computer to another?

I believe that the more Internet computing researchers and engineers can create new approaches in the context of existing approaches in synergistic domains, the more effective initial offerings will be. This issue on The Evolution of Rack-Scale Systems is a perfect complement to the ideas around The Golem Network, though the papers here represent more traditional notions of stackable processing. I would like to thank our guest editors, Fred Douglis and Dejan Milojicic, along with our authors and reviewers for producing this special issue. I hope you enjoy it.

**REFERENCES**


**ABOUT THE AUTHOR**

**M. Brian Blake** is the Nina Henderson provost and executive vice president of academic affairs at Drexel University. As a professor of computer science and electrical engineering, his research interests are in service-oriented computing, adaptive distributed systems, and Web-based software engineering. Blake has a PhD in information and software engineering from George Mason University. Contact him at mbrian.blake@drexel.edu.