Distributed protocol underpinning cloud computing automatically determined safe and secure

October 25 2021

Concept illustration of a block chain. Credit: U-M Engineering

In an important step toward ensuring the protocols that dictate how our networked services operate are safe, secure and running as expected, University of Michigan researchers have automated a technique called formal verification.

Their system proves, without any human effort, that one of the most foundational distributed computing protocols—known as Paxos—meets

its specifications. The achievement refutes a common assumption that the Paxos protocol and others like it are too complex to be proven secure without hours of manual work.

"Paxos is one of the first and most celebrated ideas that laid the foundation for how different things come to an agreement asynchronously," said Aman Goel, a doctoral student in computer science and engineering, who presented the work at the Formal Methods in Computer-Aided Design Conference Oct. 20.

The dominance of cloud computing and rising technologies like blockchain applications have changed how organizations and individuals engage with computing, creating a world powered by networked machines under a constantly growing load.

As a consequence, our critical infrastructure is more susceptible than ever to widespread fallout from server outages, hackers and buggy network behavior. Airtight distributed protocols are needed to ensure that software systems can effectively run on machines spread across the world.

These protocols are extremely complex algorithms that define how machines in a network can work collaboratively as a single system. Paxos is one of the most important examples of the category, describing an approach called consensus that has been put to use in nearly all critical distributed systems, including all of the applications supported by cloud computing.

Most recently, consensus has garnered widespread attention for enabling blockchain applications like cryptocurrencies. Such protocols form the backbone of a blockchain by helping all nodes in the network verify transactions as they happen.

"Most—if not all—consensus algorithms fundamentally derive concepts from Paxos," Goel said.

Formal verification is a class of techniques used to demonstrate that something is correct and reliable with the elegance of a logical proof. The process is very useful for software and hardware alike, providing a certificate that a certain algorithm, working piece of software or computer chip will always operate the way its specifications say it should. Theoretically, it would enable software to be released with substantially less testing than currently needed.

"Having a foolproof system that says: You develop it, you check it automatically and you get a certificate of correctness, that's what gives you confidence that you can deploy a program without issue," said Karem Sakallah, professor of computer science and engineering.

Unfortunately, proving the correctness of a program with many complex behaviors ranges from tedious to impossible—making burgeoning techniques to automate the process extremely powerful. But for algorithms on the scale of Paxos, automating its formal verification was deemed simply too large a job to ever finish successfully.

"There have been many attempts in the past to verify Paxos, including many manual attempts," Goel said. "Everyone points to a prior theoretical result that says automating it is impossible—it's beyond the tools of automation to be able to prove it."

The team's solution makes use of a feature common to all distributed protocols: Regularity. In the systems under consideration, all servers working on a particular function will be handling large batches of requests that look fundamentally the same, and the nature of their tasks will change very little over time.

This regularity enabled Goel and Sakallah to transform what started as an impossibly large task into one that looks small and manageable. They did so quite literally—by verifying the protocol under the assumption that it had a fixed, small number of nodes, and then generalizing the solution to a "theoretically unbounded number" of nodes.

The tool the researchers designed for this proof is called IC3PO, a model checking system that looks through every state a program can enter and determines whether it matches a description of safe behavior. If the protocol is correct, IC3PO produces what's termed an inductive invariant—a proof by induction that the property holds in all cases. If instead a bug is found in the protocol, it will produce a counter-example and execution trace, showing step by step how the bug manifests.

The inductive invariant IC3PO produced for Paxos in under an hour identically matches the human-written one previously derived with significant manual effort using a technique called interactive theorem proving. On top of speeding the process up, it also produces a proof with very succinct and digestible documentation.

Verifying the correctness of Paxos automatically has major ramifications for the future. As new consensus protocols are built atop its principles for ever-changing applications, they'll need to be proven safe and secure. Using a model checker like this can enable humans to work with complex software that's proven safe without having to understand every minor detail of how it works.

More information: Towards an Automatic Proof of Lamport's Paxos, arXiv:2108.08796 [cs.LO], arxiv.org/abs/2108.08796

Provided by University of Michigan