A Trusted Reimbursed Computing System based on WebAssembly and Intel SGX

Manuel Nieke
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Rüdiger Kapitza
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Volunteer computing provides scientists with computational resources required for complex computation tasks of their research. For this purpose, volunteers worldwide donate unused processing time of their machines and are often motivated by altruistic or competitive reasons as volunteer computing systems often keep leaderboards of their top donators. However, these incentives often fail to keep volunteers motivated for long and even popular volunteer computing frameworks such as BOINC only demonstrate an active user base of 3.4% of all users. Furthermore, much of the resources gained are wasted on multiple executions of the same workload in order to compare their results and ensure correctness. This is necessary as there are no restrictions on who can volunteer and participants may be malicious or attempt to cheat for better leaderboard positions.

To provide better incentives for participants we, therefore, propose the concept of *reimbursed computing* where participants are rewarded with small amounts of money or online goods in return for their computational resources. To ensure the confidentiality and integrity of the computations, we also employ the wide-spread, novel hardware technology *Intel SGX* which provides verifiable trusted execution. As a result, more scientists will become interested in using reimbursed computing for their research projects as especially companies favour to keep their research secret. This also allows us to improve the efficiency of volunteer computing by several times as computations no longer have to be performed multiple times to ensure the correctness of the untrusted participants' results.

Furthermore, we implemented a prototypical framework for reimbursed computing based on the novel technology *WebAssembly*, which allows the development of portable high performance applications and provides sandboxing mechanisms for protecting users from malicious applications. Using this technology additionally enables us to directly integrate our framework into commonly used browsers such as *Google Chrome* for an improved user experience and seamless integration into the web for the discovery of projects to support as well as payment for online services.

In order to reimburse participants according to their supplied computational resources we implemented a secure, precise, and fair accounting system which counts executed WebAssembly instructions. As these instructions are platform-independent, the accounted resources stay the same regardless of the platform used, while still maintaining a high precision regarding the complexity of the workload.

Our evaluation of adapted volunteer computing projects shows that our framework introduces only a 15% performance overhead compared to non-secure, non-accounted applications which is highly preferred over the wasted resources of redundant execution and the lacking secrecy of normal volunteer computing frameworks.