Semantics of Mergeable Data-structures

Subject

We are trying to make the Cloud a safer and saner place to live. To do so, we have writing a full operating system in OCaml, dubbed "MirageOS" [4] which compiles complex applications to single, sealed, hyper-specialized and self-contained virtual machines (or "unikernels") that can be easily deployed on any Public Cloud. These unikernels are safer than usual Virtual Machines based on standard application stacks (such as LAMP: Linux+Apache+Mysql+PHP), as they are written in a high-level language with strong static typing guarantees, with lots of whole-program analysis and optimizations done at compile-time – and they are as efficient as their counterparts as the small penalty of using a high-level language is balanced by the removal of all the legacy layers embedded in more standard operating systems.

We are now focusing on new programming techniques and language constructs to be used to coordinate hundreds thousands of such unikernels. To do so, we have designed "Irmin", a weakly consistent store of persistent data-structures and we have started extending existing persistent data-structures [5] such as queues and ropes with an efficient merge operation [3].

The goal of this internship is to formalize the merge operations on persistent data-structures by extending the semantics of concurrent revisions [1] and investigate more systematic ways of turning persistent data-structures into mergeable ones, as it has already been done for persistent data-structure themselves [2].

References

[1] S. Burckhardt and D. Leijen. Semantics of concurrent revisions. In *ESOP*, 2011.

[2] James R. Driscoll, Neil Sarnak, Daniel D. Sleator, and Robert E. Tarjan. Making data structures persistent. *J. Comput. Syst. Sci.*, 38(1):86–124, February 1989.

[3] B. Farinier, T. Gazagnaire, and A. Madhavapeddy. Mergeable data-structures. In *JFLA*, 2015.

[4] A. Madhavapeddy, M. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft. Unikernels: library operating systems for the cloud. In *ASPLOS*, 2013.

[5] C. Okasaki. *Purely Functional Data Structures*. PhD thesis, Pittsburgh, PA, USA, 1996. AAI9813847.

Profil

The successful candidate will have a background in Computer Science, with a strong interest in Functional Programming and/or Operating Systems. A working knowledge of OCaml and a taste for writing software would certainly be a plus.

Contact

For any question, please contact `thomas.gazagnaire@cl.cam.ca.uk`.