Contribution ID: **395**   Type: **not specified**

Automatically reasoning about the cache usage of network stacks

Wednesday, 18 September 2024 16:00 (30 minutes)

Efficient use of the CPU cache is critical for network stacks to demonstrate good performance. However, reasoning about cache usage is hard, as demonstrated by a recent kernel patch [1] that showed how the fast path of the Linux TCP stack had been accessing 50% more cache lines than necessary for several years.

We present CFAR, a tool that enables developers to automatically reason about the cache usage of their own and third-party code. CFAR achieves this using a two-step process. First, CFAR uses automated program analysis to transform the code into an abstract representation (a "distillate") of how the code accesses memory. Then, CFAR allows developers to write simple queries that transform the distillate into answers to specific questions about cache usage. Since the distillate is a precise abstraction of the code's memory usage (i.e., it contains all the information relevant to how the code accesses memory), developers can use projectors to answer diverse questions about the code's cache usage.

Our initial results with CFAR are promising. In addition to being able to automatically replicate the results of the patch, CFAR has identified similar inefficient cache access patterns in other kernel-bypass stacks. Finally, as a side benefit, we used CFAR to analyze more than just performance properties related to caches and show how it can be used to identify cache-related leakages in cryptographic code.

[1] - https://lore.kernel.org/netdev/20231129072756.3684495-1-lixiaoyan@google.com/

Primary author: IYER, Rishabh

Co-authors: Prof. CANDEA, George (EPFL); Prof. ARGYRAKI, Katerina (EPFL)

Presenter: IYER, Rishabh

Session Classification: Networking Track

Track Classification: Networking Track