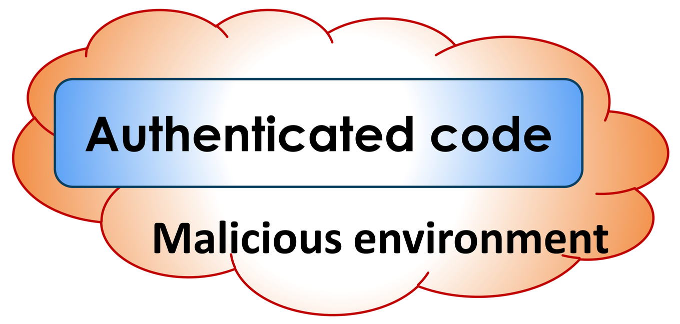
UNIVERSITY OF MICHIGAN
STARnet

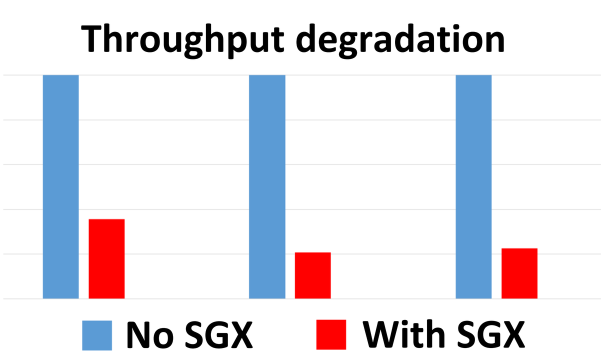
Regaining Lost Cycles with **HotCalls**: A Fast Interface for **SGX** Secure Enclaves

Ofir Weisse
Valeria Bertacco
Todd Austin

Center for Future Architectures Research

SGX Secure Execution





Is it **practical**?

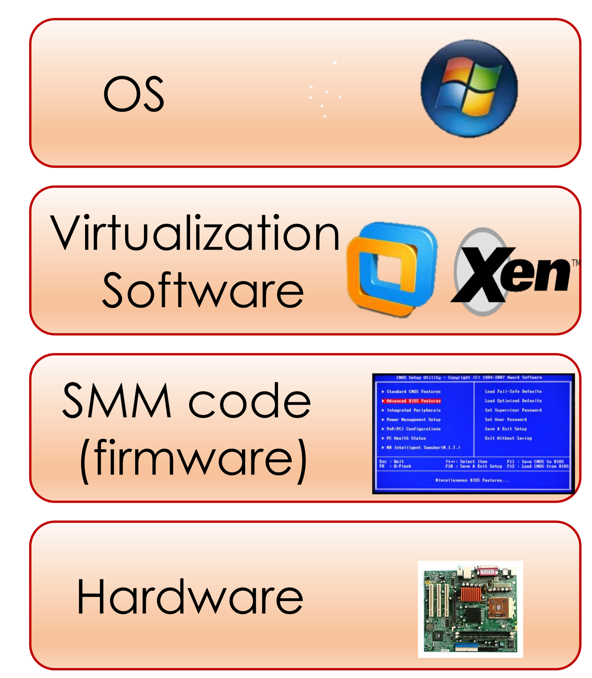
- What is the impact on overall application's performance?
- What creates the bottlenecks?

Contribution: overcoming the bottlenecks with **HotCalls**

Cloud Attack Surface

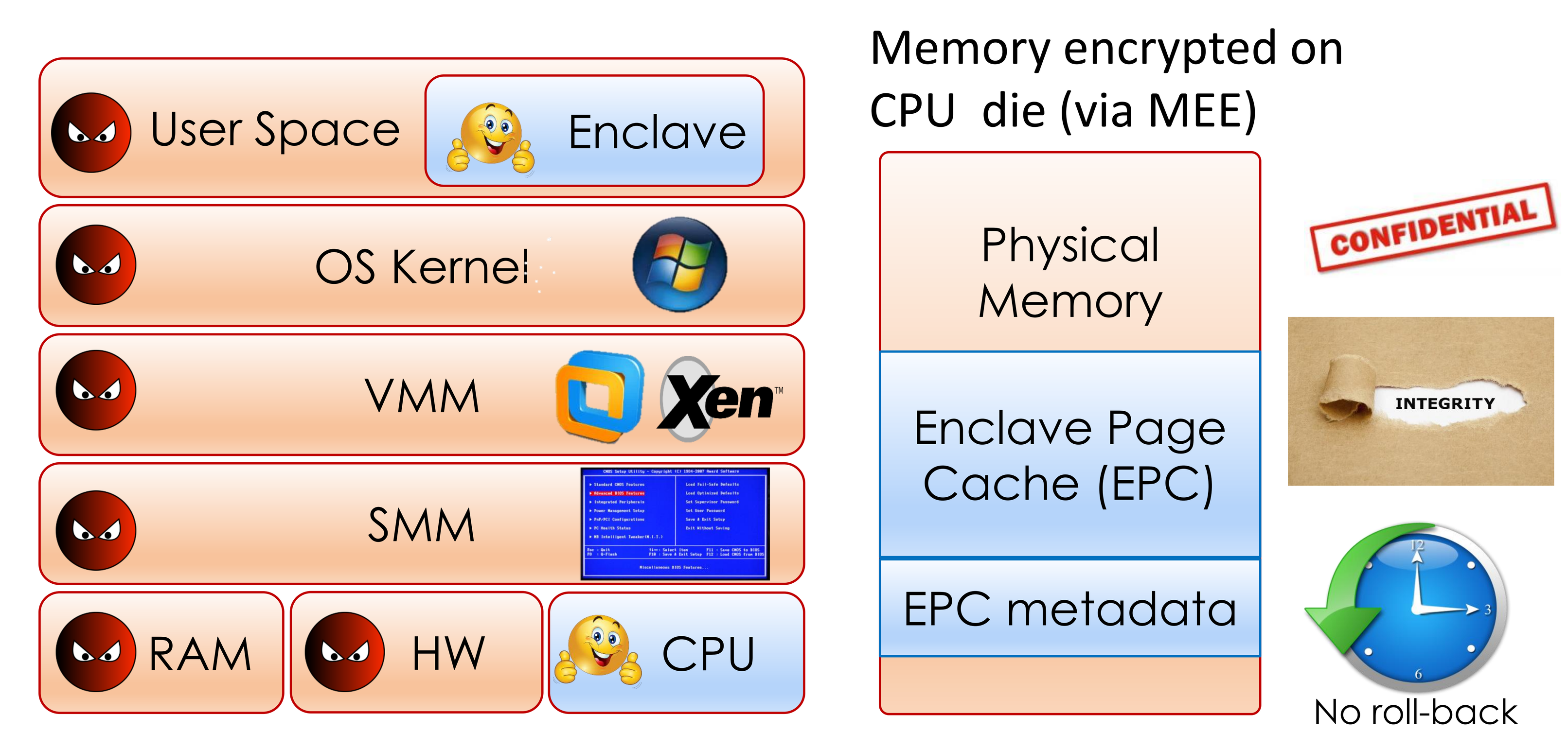
- To lower costs - computation and storage are moved to third party machines
- This implies trust across the entire software stack

- And cloud provider employees



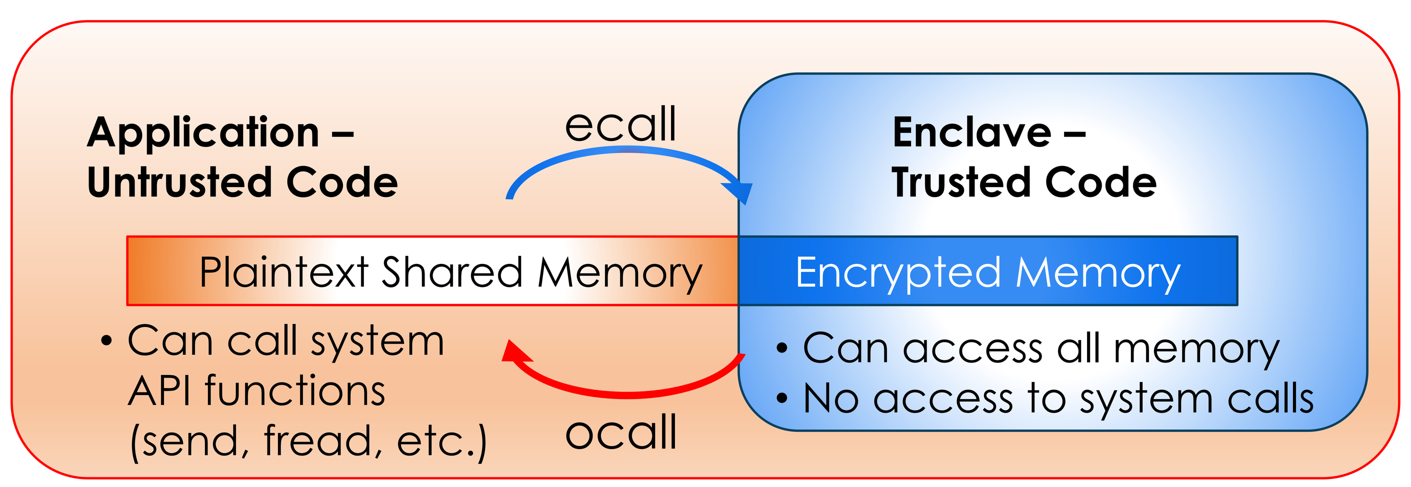
The attack surface is large

SGX in a Nutshell



Secure enclave is isolated from the system

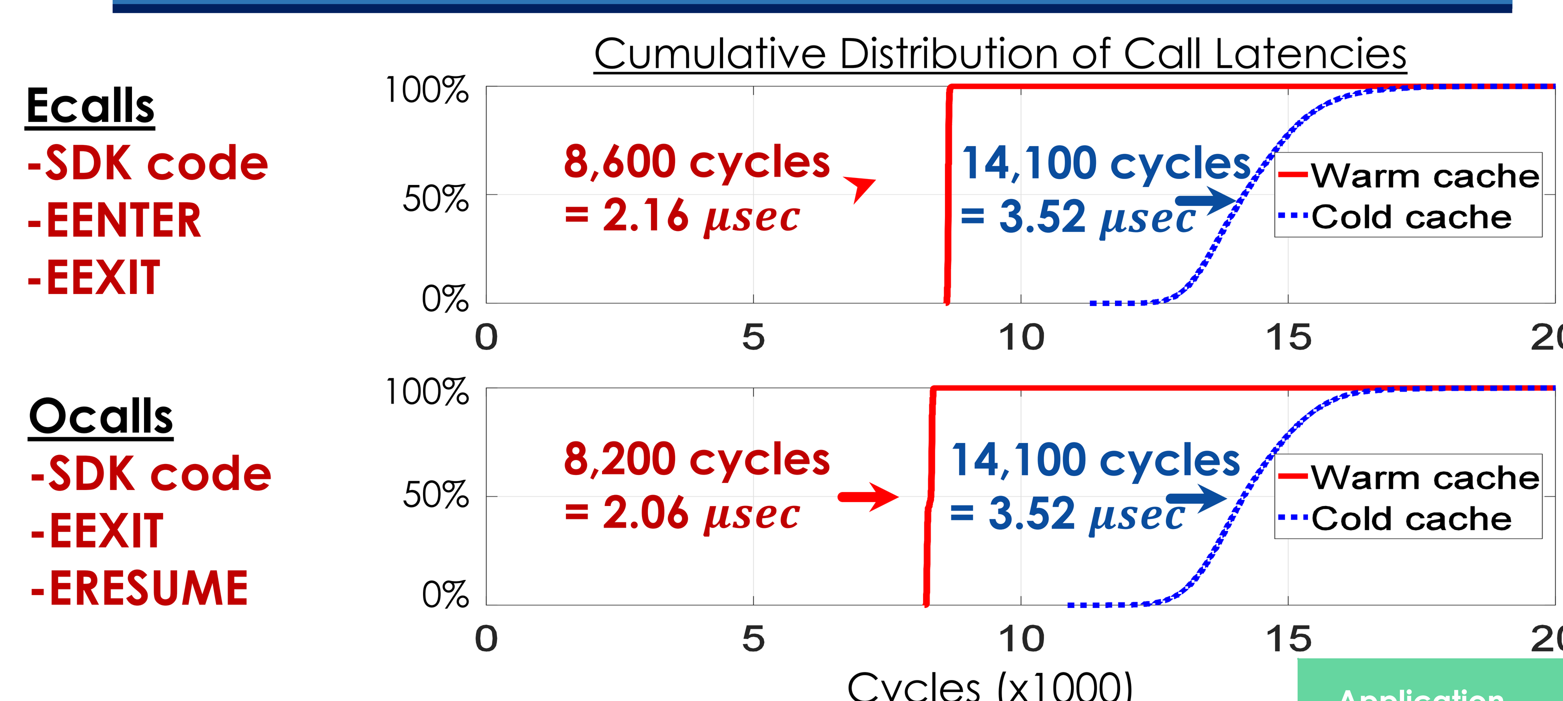
SGX Life-Cycle



- Ecall: a secure context switch to enclave code
- Ocall: reverse context switch to request OS services

SGX operations may become a bottleneck

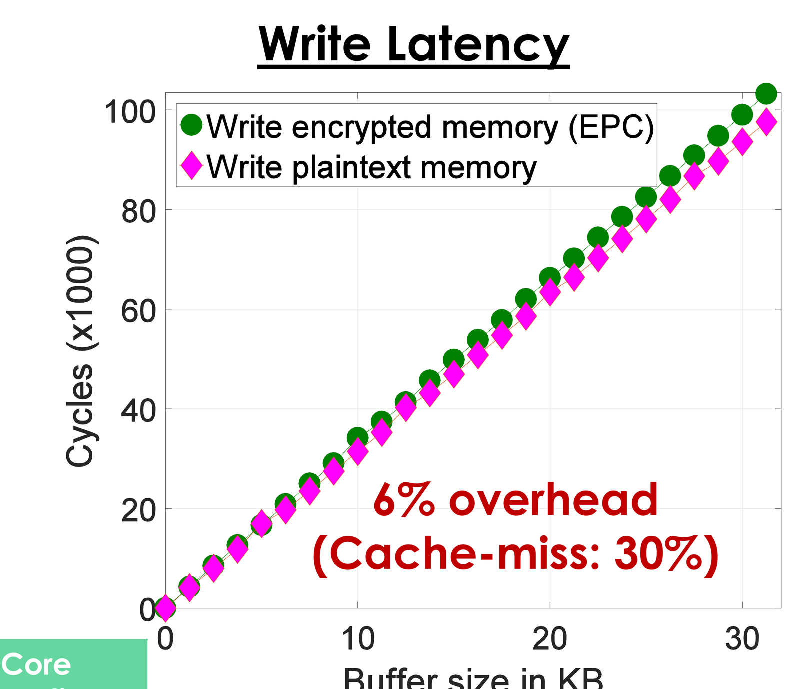
Cost of Ecalls/Ocalls

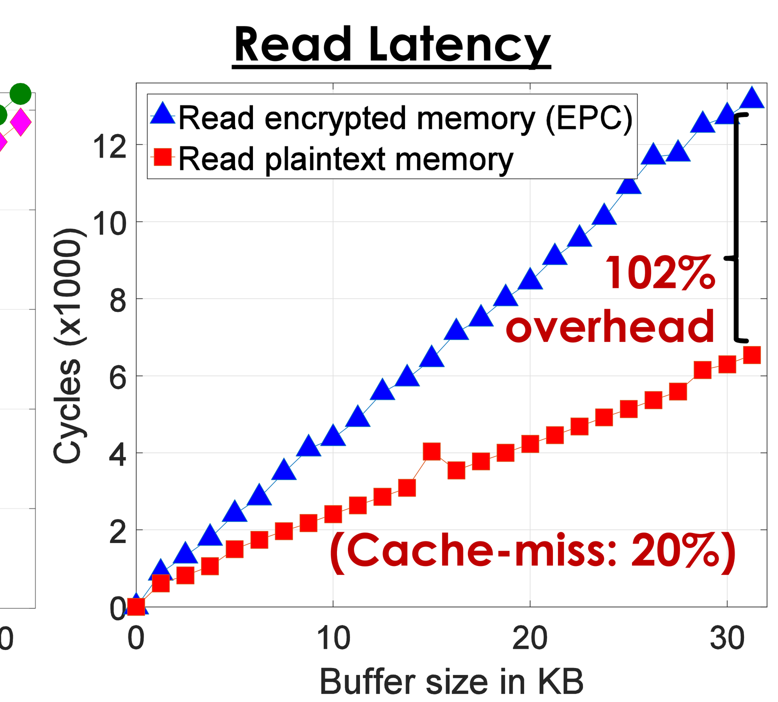


Application	# Calls /second	Core spending
Memcached	200,000	43%
OpenVPN	275,000	57%
Lighttpd	270,000	56%

Context switches may dominate performance

Cost of Encrypted Memory

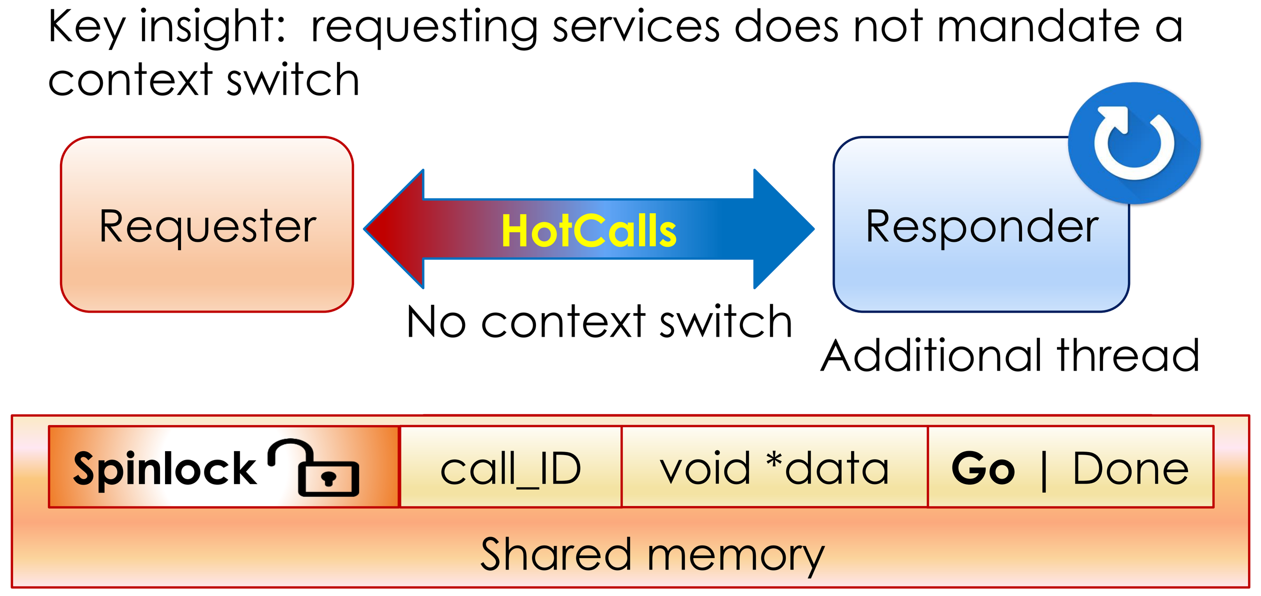




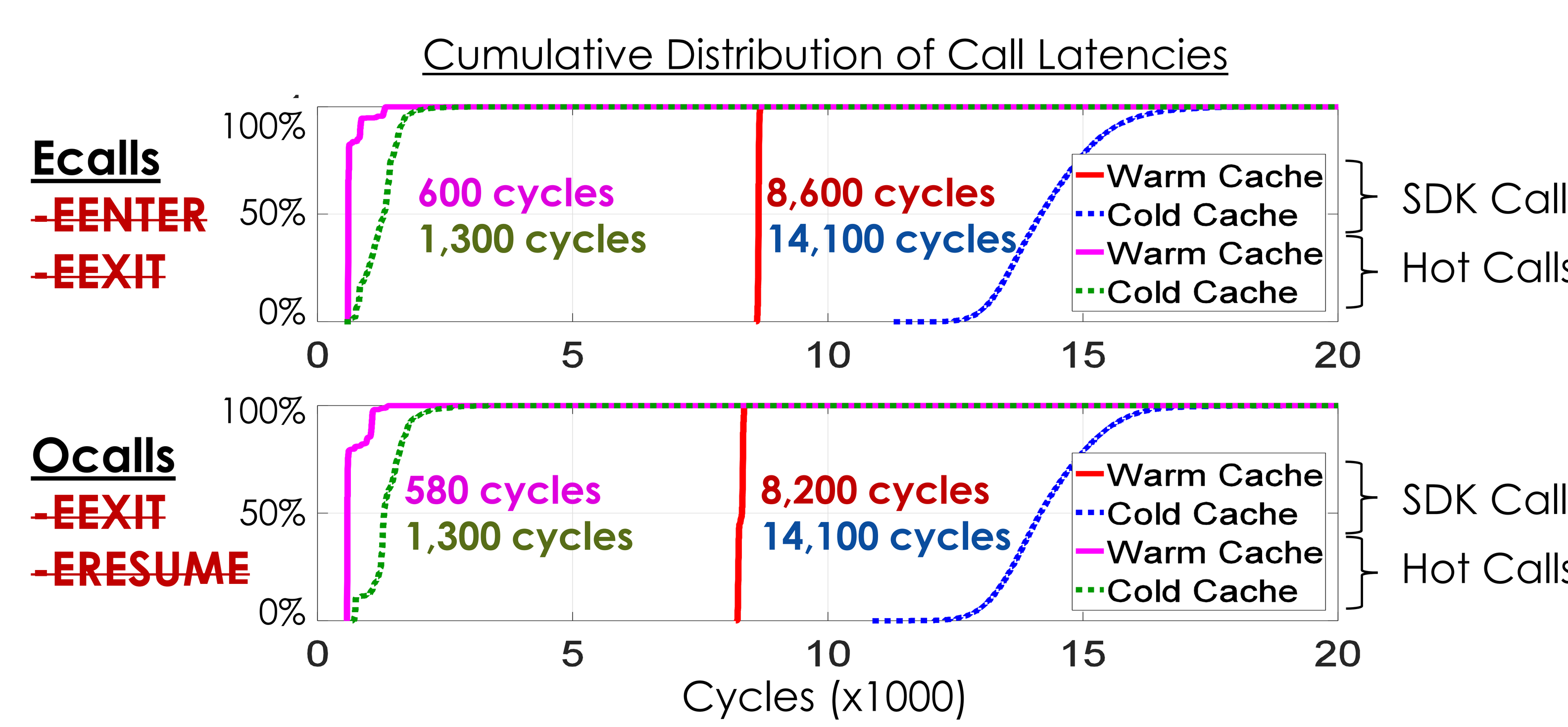
Encrypted memory is a potential bottleneck

HotCalls Mechanism

Key insight: requesting services does not mandate a context switch

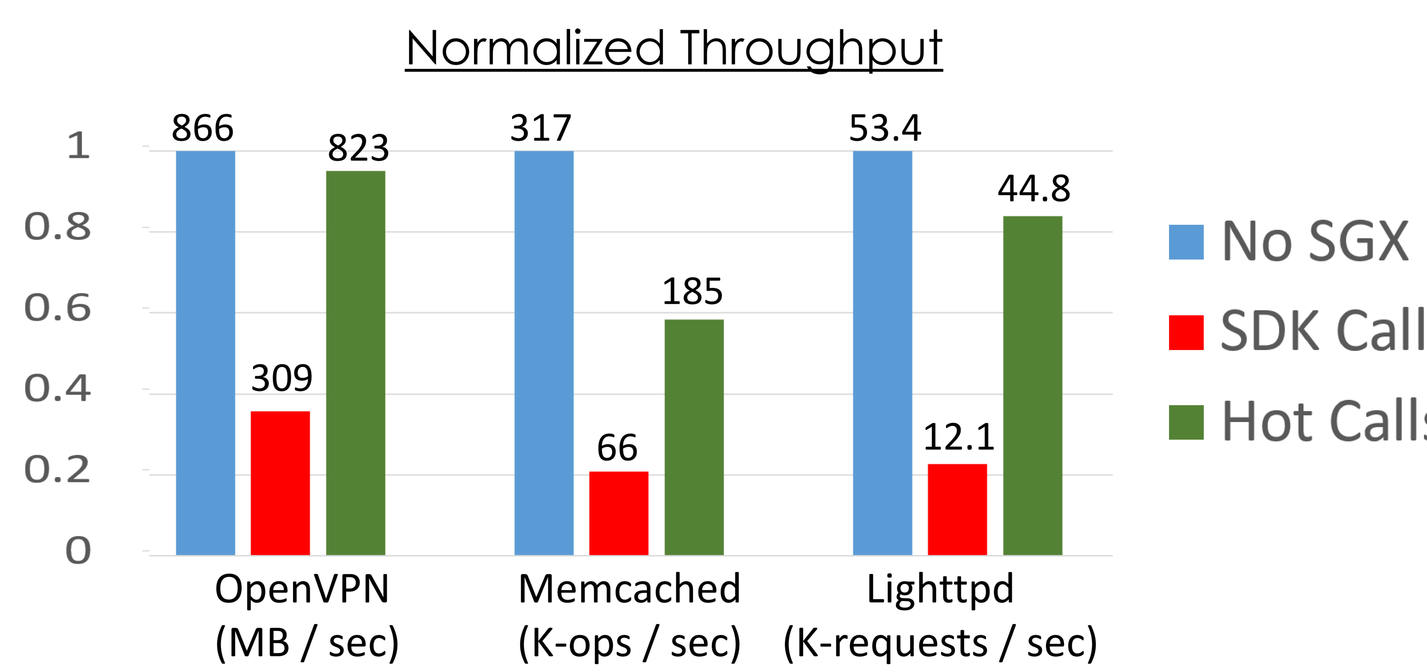


HotCalls vs. SDK Calls



HotCalls are order of magnitude faster

HotCalls in Action



Published at ISCA 2017

Making SGX great again