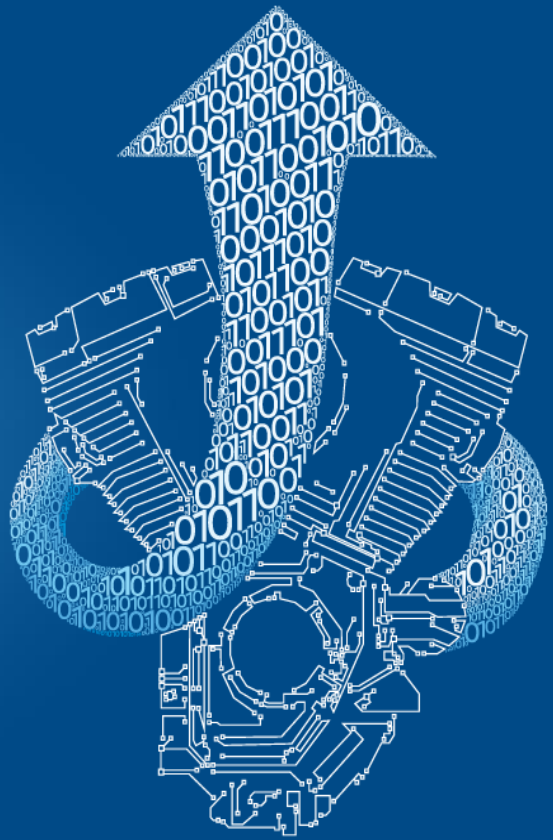
Software

跨越从嵌入式到云端的全新容器

WebAssembly Micro Runtime



Aug 2020
xin.wang@intel.com

目录

- WebAssembly Micro Runtime (WAMR)项目技术特点
- 嵌入式环境下的WAMR容器应用
- 可信计算环境的支持与应用
- WASM的云端Serverless应用机会

WebAssembly Micro Runtime (WAMR)项目技术特点

技术特点
性能与内存消耗数据
路线图规划

https://github.com/bytecodealliance/wasm-micro-runtime

WAMR使命与设计目标

独立于浏览器的WASM运行引擎，着重实用性

高度模块化与可定制化 – 满足从嵌入式到云端的容器引擎需求的碎片化

目标定义为最好的性能、最小的内存消耗

支持解释器、Ahead-of-Time(AoT)编译、Just-in-Time(JIT)编译多种模式

多CPU架构、多平台支持

自主实现AoT loader，任何平台上都能支持AoT模式

多实例支持与管理

提供可扩展的WASM微服务应用编程框架（异步模式）

远程WASM应用管理框架

WAMR项目技术特点

- Full WASM MVP spec support and aggressive support on post MVP features
- Support Interpreter, Ahead of Time (AoT) and Just-in-Time (JIT) Compilation
- C based implementation, LLVM based compilation
- Small footprint and memory consumption
  - VMCore – 50K for AoT, 80K for interpreter
  - Down to 16K DRAM for running hello-world
- Blazing fast interpreter and compilation
- AoT module loader on Linux, Zephyr, SGX(Linux)
- Native SGX support for interpreter and AoT, Native WASI for SGX (No need libOS)
- Support pthread, multiple modules inter-calling

https://github.com/bytecodealliance/wasm-micro-runtime

- CPU Arch support:
  - X86-64, X86-32
  - ARM, THUMB
  - MIPS
  - XTENSA
- Platform support:
  - Linux, SGX (Linux)
  - Windows
  - MacOS
  - Android
  - Zephyr
  - AliOS Things
  - Vxworks

WAMR性能数据(X86)

The AoT performance is close to the GCC native compilation among the workloads

benchmark	Gcc (O3)	iwasm-jit	iwasm-aot	AoT/GCC
base64	0.20	0.26	0.20	1
fib2	0.53	0.83	0.87	0.609195
gimli	0.16	0.17	0.17	0.941176
heapsort	0.72	0.81	0.66	1.090909
memmove	2.76	9.32	9.24	0.298954
nestedloop	3.09	2.93	2.89	1.069204
nestedloop2	3.56	2.80	2.77	1.282452
nestedloop3	16.52	0.93	0.95	17.33217
random	0.43	0.31	0.30	1.422222
seqhash	2.35	4.12	3.18	0.73822
sieve	0.26	0.36	0.33	0.787879
strchr	0.37	0.17	0.25	1.48
switch2	0.23	0.34	0.17	1.352941
fib	0.01	0.03	0.02	0.5
pollard	0.01	0.14	0.01	1

GCC may generates SSE code

Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
Ubuntu 18.04 64-bit
gcc-7.5.0

WAMR在社区的应用案例与展望

IoT与穿戴式小程序

区块链智能合约：Hyperledger Private Data Objects

可信运行环境(TEE)应用运行引擎

联邦共享计算

动态固件引擎

工业自动化控制引擎

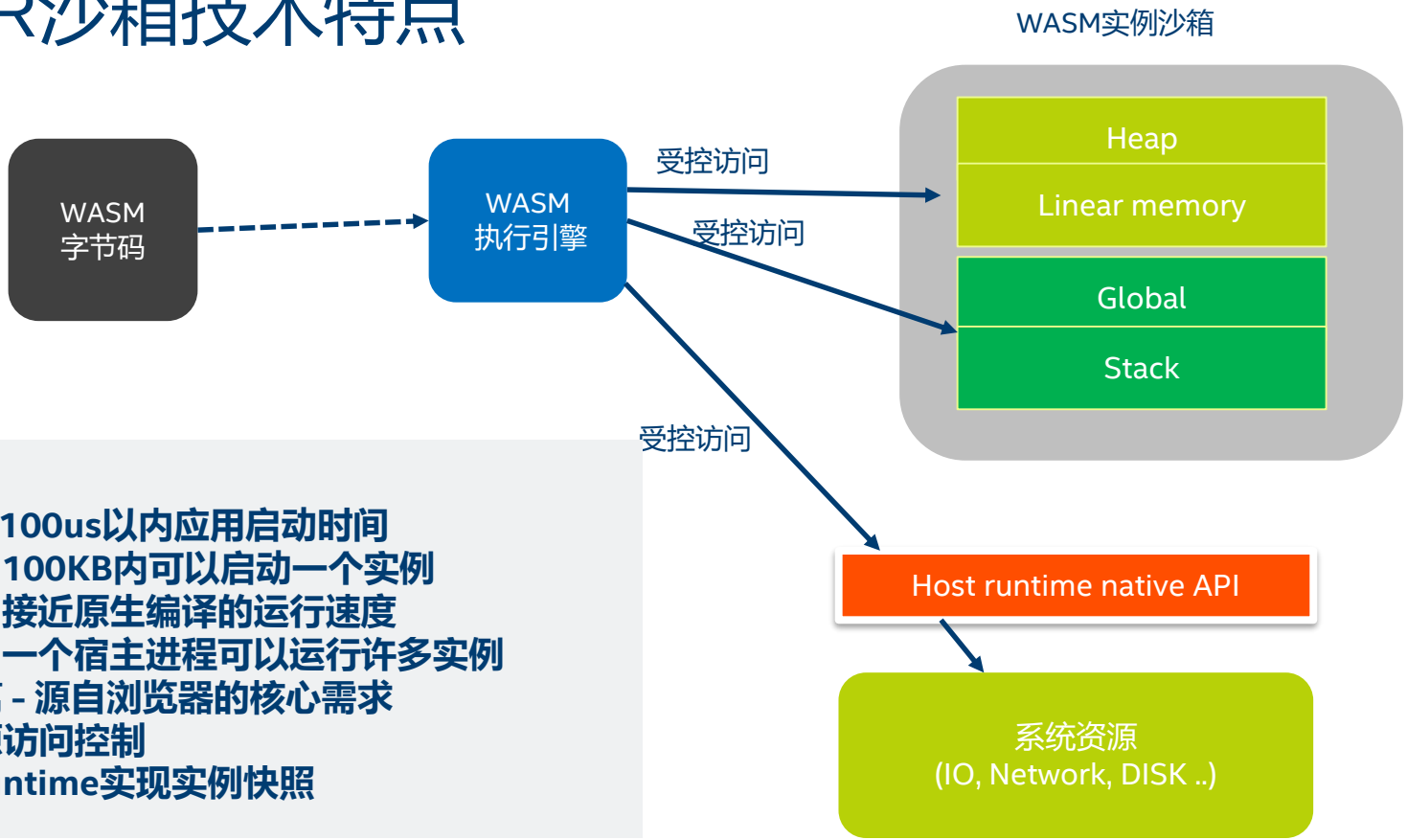
游戏引擎

存储加速

手机小程序引擎

超轻量级 serverless 容器

WAMR沙箱技术特点



特点：

- **启动快 - 100us以内应用启动时间**
- **消耗小 – 100KB内可以启动一个实例**
- **速度快 – 接近原生编译的运行速度**
- **高并发 – 一个宿主进程可以运行许多实例**
- **可靠隔离 - 源自浏览器的核心需求**
- **系统资源访问控制**
- **可通过runtime实现实例快照**

2020开发规划

Post MVP features - WIP

Threading and pthread – 完成

Multi modules – 完成

Source code debugging – Q4

SIMD support – Q4

Performance and footprint optimizations - WIP

Software

嵌入式环境中的WAMR容器应用

WAMR应用框架

一个应用框架(App Framework)基于两个基本操作

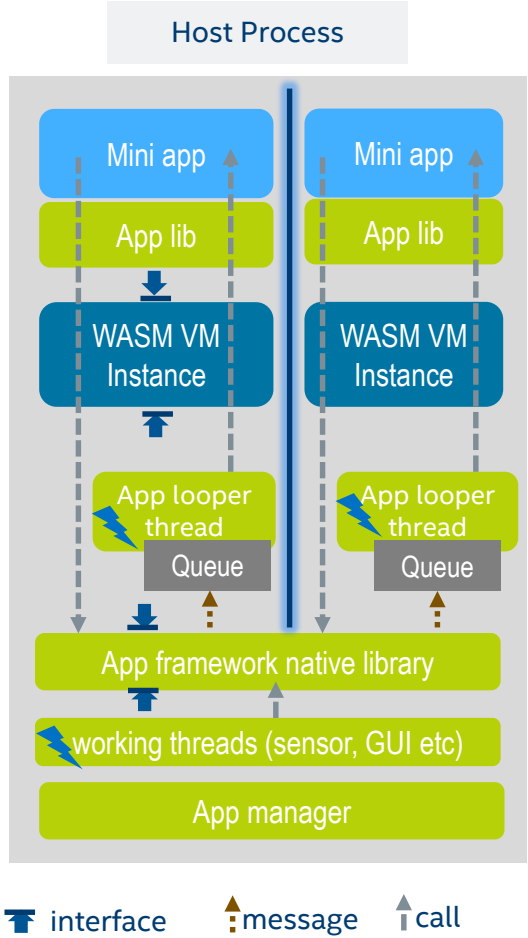
- 从用户WASM APP调入原生(native)的函数
- 从原生代码调用WASM函数

应用异步编程模式的设计方式

- 每个app有一个独立线程和沙箱
- 消息驱动

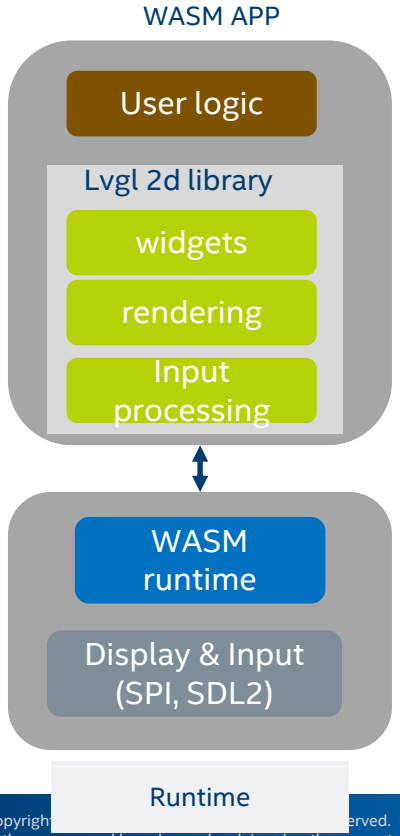
应用管理

- 安装、启动、卸载小应用

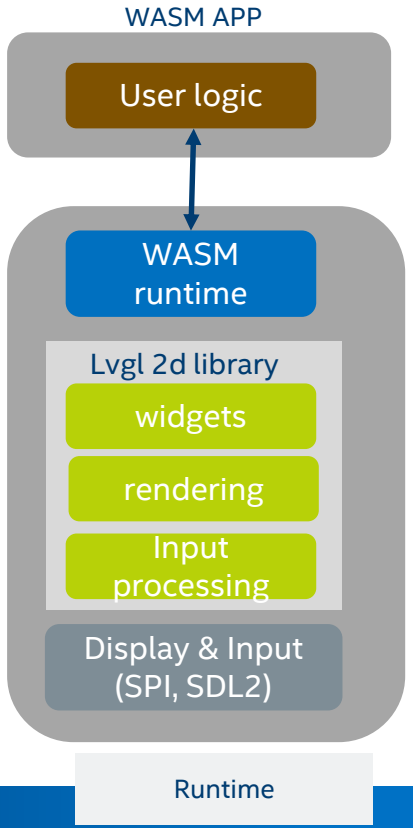


WASM人机图形界面示例

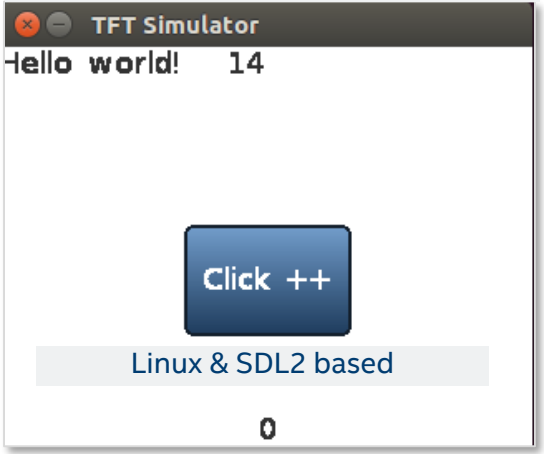
Sample littlevgl: whole lvgl 2d library is compiled to WASM



Sample gui: whole lvgl 2d library is compiled into runtime



The same WASM binary cross different hardware and systems



Software

可信计算环境中的WAMR容器应用

WAMR对SGX的全面支持与应用

WASM成为可信计算界的“网红”

- Language independent virtual machine allows application diversity
- Hardware independent virtual machine allows HW-specific capabilities and optimizations

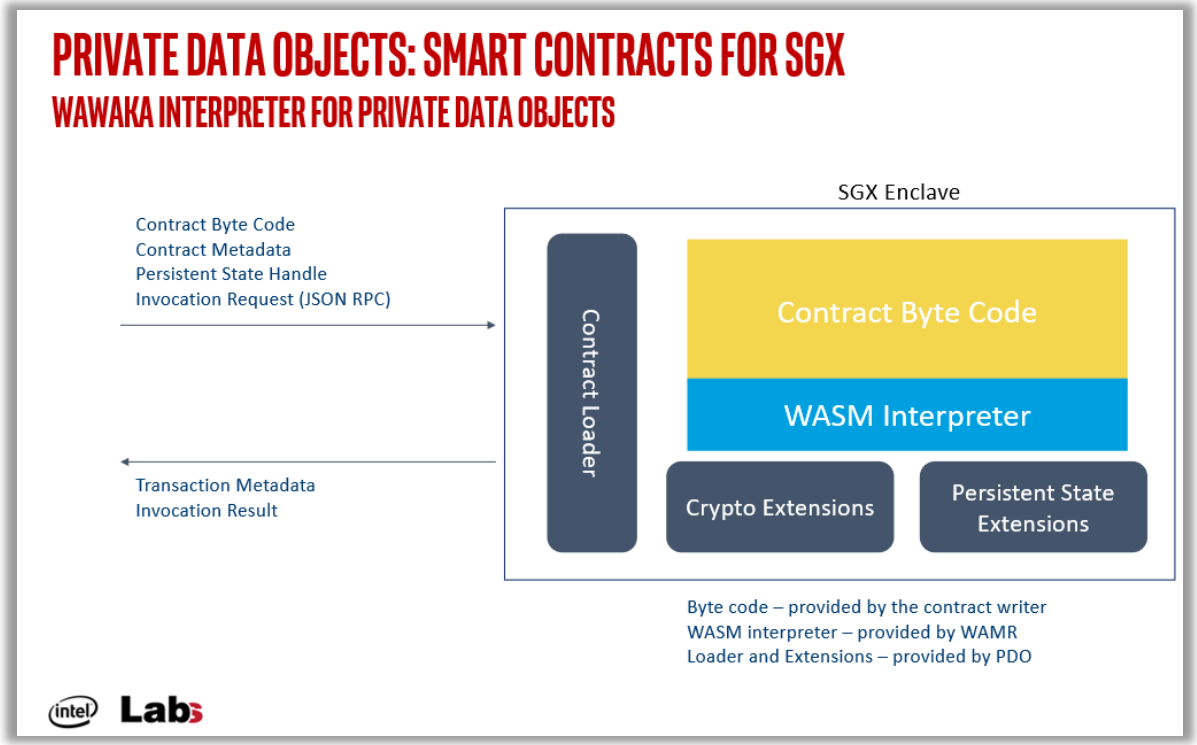
WAMR提供了对AoT和解释器在SGX enclave的原生支持，JIT仍然需要libOS

WAMR原生支持WASI for SGX 并且在enclave中运行TensorFlow的WASM版本

和学术界的WASM/SGX合作项目：Smart contract, Trusted FaaS and WASM remote attestation

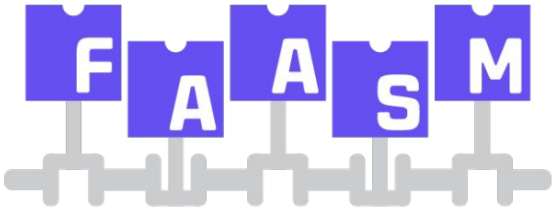
区块链大型商业系统中基于SGX的智能合约应用

应用案例：Private Data objects项目智能合约



应用案例：Faasm and Trusted FaaS

Faasm: Lightweight Isolation for Efficient Stateful Serverless Computing



Serverless with WebAssembly:

- **Faaslets** - lightweight serverless isolation
- **Local and global state** - locally shared, globally synchronised
- **Fast initialisation** - snapshot and restore, with cross-host migration
- **Flexible (POSIX-like) host interface** - serverless-specific, WASI and more

Paper preprint https://arxiv.org/pdf/2002.09344.pdf

Github: https://github.com/lsds/Faasm

WASM的云端Serverless应用

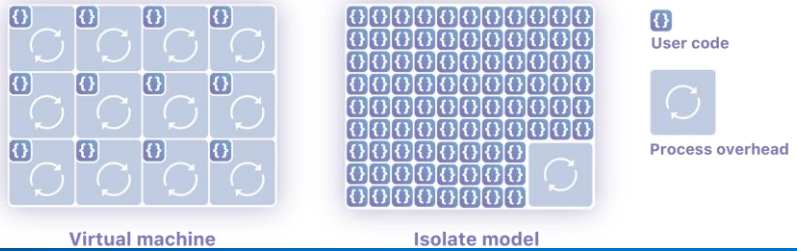
WASM超轻量级Serverless容器

Serverless与FaaS给云端应用开发者带来极大便利与迅捷

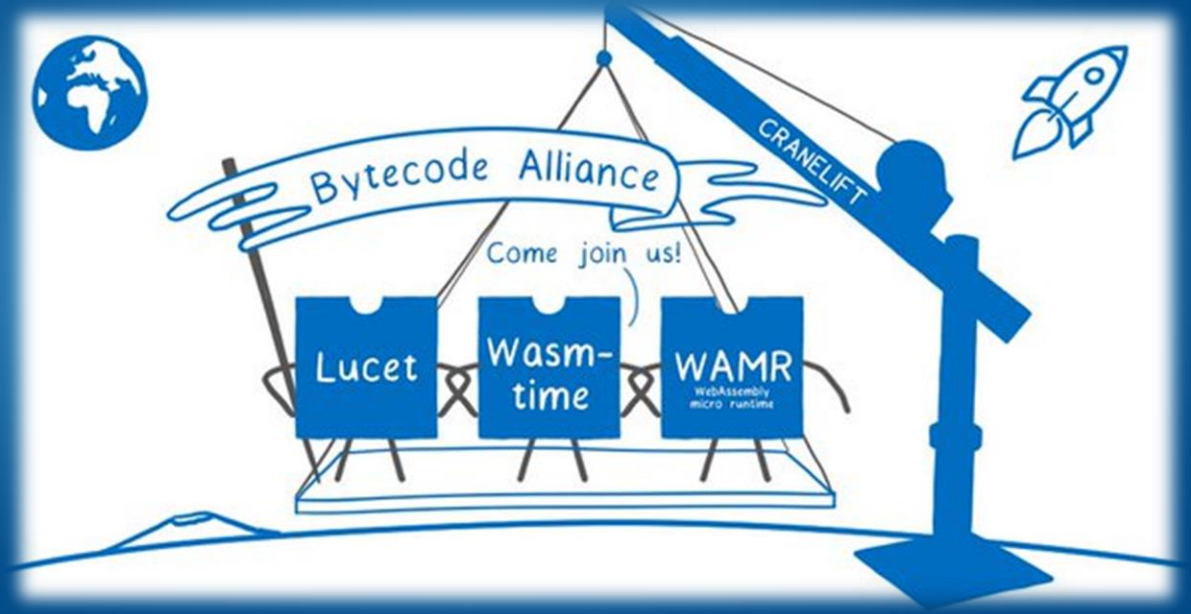
传统的Docker容器之上的serverless方案性价比对于云厂商仍有较大提升空间 (也许可以说是痛点)

使用WASM作为一种新型超轻量级容器正在兴起

- 解决Serverless性价比问题
- Fastly, Cloudflare已经使用WASM技术指数级提升边缘服务器的用户程序吞吐量
- 从公开信息中各大云商的积极尝试 - 微软Krustlet, 阿里巴巴/蚂蚁，腾讯...



CALL FOR COLLABORATION



微信加联系入群

钉钉扫码入群