Technische
Universität
Braunschweig

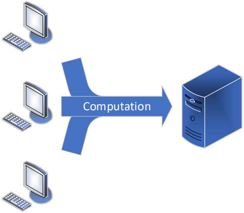
Institute of Operating Systems
and Computer Networks

A Trusted Reimbursed Computing System based on WebAssembly and Intel SGX

Manuel Nieke, Rüdiger Kapitza,

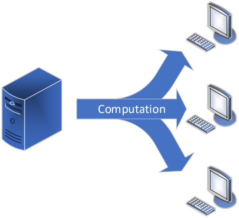
funded by intel

Computation Offloading



- Idea: Move computations to remote party
  - Gain additional computation power
  - More flexible resource usage

- Some use cases:
  - Cloud applications
  - Volunteer computing



Drawbacks

Problem: Loss of control

- Remote party can
  - access (sensitive) data
  - interfere with execution

Drawbacks

Problem: Loss of control

- Remote party can
  - access (sensitive) data
  - interfere with execution

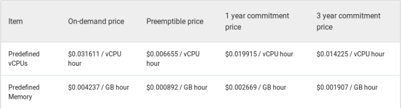
⇒ Suboptimal solutions
  - Sensitive data not moved to cloud
  - Volunteer computing workloads computed multiple times

- Better: Trusted execution
  - Relies on hardware support

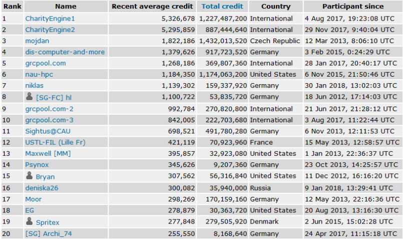
Drawbacks

Problem: No control over resource accounting

- Billing in the cloud
  - Cloud provider can “overbill”
- Leaderboards in volunteer computing
  - Volunteers cheat to get better ranking



Item	On-demand price	Preemptible price	1 year commitment price	3 year commitment price
Predefined vCPUs	$0.031611 / vCPU hour	$0.006655 / vCPU hour	$0.019915 / vCPU hour	$0.014225 / vCPU hour
Predefined Memory	$0.004237 / GB hour	$0.000892 / GB hour	$0.002669 / GB hour	$0.001907 / GB hour



Rank	Name	Recent average credit	Total credit	Country	Participant since
1	CharityEngine1	5,326,678	1,227,487,200	International	4 Aug 2017, 19:23:08 UTC
2	CharityEngine2	5,295,859	887,444,640	International	29 Nov 2017, 9:40:04 UTC
3	mojdan	1,822,186	1,432,013,520	Czech Republic	12 Mar 2013, 8:06:10 UTC
4	dis-computer-and-more	1,379,626	917,723,520	Germany	3 Feb 2015, 0:24:29 UTC
5	grcpool.com	1,268,186	369,807,360	International	28 Jan 2017, 20:40:17 UTC
6	nau-hpc	1,184,350	1,174,063,200	United States	6 Nov 2015, 21:50:46 UTC
7	niklas	1,139,302	159,337,920	Germany	30 Jan 2018, 13:02:03 UTC
8	[SG-FC] hl	1,100,722	53,835,720	Germany	18 Jun 2012, 17:14:03 UTC
9	grcpool.com-2	992,784	270,820,800	International	21 Jun 2017, 21:28:12 UTC
10	grcpool.com-3	842,005	222,703,680	International	3 Aug 2017, 11:22:44 UTC
11	Sightus@CAU	698,521	491,780,280	Germany	6 Nov 2013, 12:11:53 UTC
12	USTL-FIL (Lille Fr)	421,119	70,923,960	France	15 May 2013, 12:58:57 UTC
13	Maxwell [MM]	395,857	32,923,080	United States	1 Jan 2013, 22:36:37 UTC
14	Psynox	345,626	9,207,360	Germany	23 Oct 2013, 14:25:57 UTC
15	Bryan	307,562	56,316,840	United States	11 Dec 2012, 16:16:20 UTC
16	deniska26	300,082	35,940,000	Russia	9 Jan 2018, 13:29:41 UTC
17	Moor	298,269	170,159,160	Germany	12 May 2013, 22:16:36 UTC
18	EG	278,879	30,363,720	United States	20 Aug 2013, 13:16:30 UTC
19	Spritex	277,848	279,505,920	Denmark	2 Jun 2015, 15:02:28 UTC
20	[SG] Archi_74	255,550	8,168,640	Germany	24 Apr 2017, 11:15:18 UTC

Our Goals

Execution platform for computation offloading

- Execution and data protected from host system
- Host system isolated from malicious programs

Resource accounting

- Not forgeable
- Independent of platform

- **WebAssembly and SGX**
- **Resource Accounting**
- **Trusted Execution Platform**
- **Evaluation**

WebAssembly

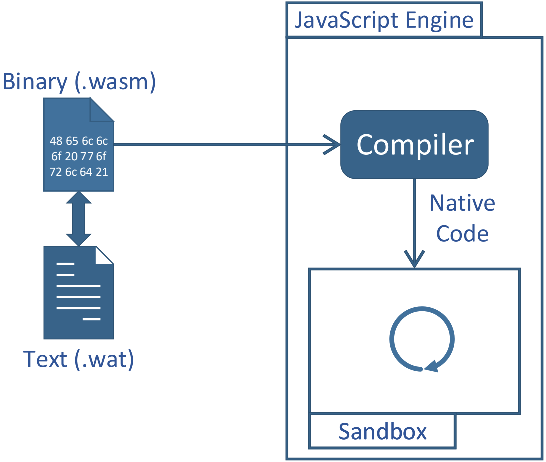
- Goal: Fast, isolated code execution in browser

- Mozilla: asm.js
  - JavaScript subset with better performance
  - Transcompile regular programs

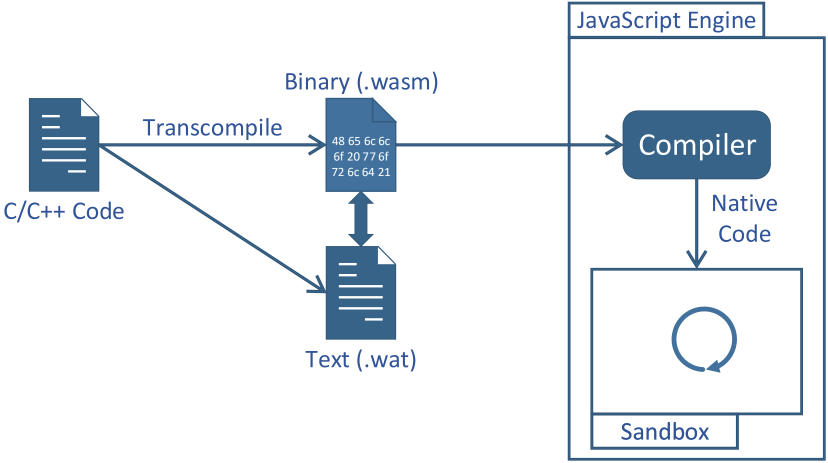
WebAssembly

- Goal: Fast, isolated code execution in browser
- Mozilla: asm.js
  - JavaScript subset with better performance
  - Transcompile regular programs
- Google: PNaCl
  - Native code in sandbox
- Combine both ⇒ WebAssembly (WASM)

WebAssembly



WebAssembly



WebAssembly

- Polyglot
  - C/C++, Rust, Go, …
- Platform independent
- Sandboxed execution
- “Near native” speed

WebAssembly

- Polyglot
  - C/C++, Rust, Go, ...
- Platform independent
- Sandboxed execution
- “Near native” speed

Trusted Execution Platform

✓ Host isolation

Intel SGX

- Instruction set extension for Intel CPUs
- Introduced in Skylake (2015)

- Parts of applications run inside *enclaves*
  - Encrypted
  - Integrity protected
  - Remote attestation
    - Verify that applications runs correctly in enclave

Intel SGX

- Instruction set extension for Intel CPUs
- Introduced in Skylake (2015)
- Parts of applications run inside *enclaves*
  - Encrypted
  - Integrity protected
  - Remote attestation
    - Verify that applications runs correctly in enclave

Trusted Execution Platform

✓ Data not visible

✓ Protection against interference with execution

CPU Accounting

- Accounting usually reliant on time
  - E.g. vCPU/h
- Problem: No accurate trusted time in enclave

CPU Accounting

- Accounting usually reliant on time
  - E.g. vCPU/h
- Problem: No accurate trusted time in enclave

- Solution: Accounting based on executed instructions
  - Instrument application code to count instructions
  - Based on (platform independent) WASM instructions
  - Utilize text representation

- Original approach
  - Based on basic blocks
    - – No if, loop, return, ...
  - Counter incremented at end of block

```
get_global 12
set_local 3
i32.lt_s
<Increment counter by 3>
if (result i32)
    get_local 0
    i32.load offset=4
    <Increment counter by 2>
else
    get_local 4
    i32.const 255
    i32.and
    <Increment counter by 3>
end
tee_local 4
get_local 1
<Increment counter by 2>
```

- Optimised approach
  - Consider only possible counter values
  - Based on possible control flows

```
get_global 12
set_local 3
i32.lt_s

if (result i32)
   get_local 0
   i32.load offset=4

else
   get_local 4
   i32.const 255
   i32.and
   <Increment counter by 1>
end
tee_local 4
get_local 1
<Increment counter by 7>
```

- Optimised approach
  - Consider only possible counter values
  - Based on possible control flows

- Further optimisation
  - Identify loop iterators with constant increment
  - Compare iterator before and after loop to calculate iterarions
  - Increment counter <u>once</u>

```
get_global 12
set_local 3
i32.lt_s

if (result i32)
    get_local 0
    i32.load offset=4

else
    get_local 4
    i32.const 255
    i32.and
    <Increment counter by 1>
end
tee_local 4
get_local 1
<Increment counter by 7>
```

Other Resources

- Memory
  - WASM memory as contiguous blocks
  - Easy to determine size
- File and network I/O
  - Platform provides functions to WASM
  - Modify functions to measure I/O volume

Other Resources

- Memory
  - WASM memory as contiguous blocks
  - Easy to determine size

- File and network I/O
  - Platform provides functions to WASM
  - Modify functions to measure I/O volume

Instrumentation

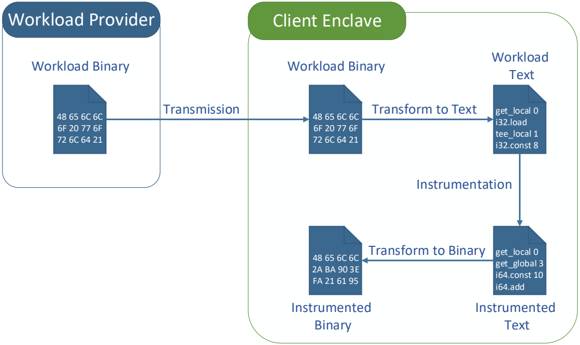
✓ Platform independent

? Non-forgeable

Instrumentation

- Accounting needs to be trusted by all parties

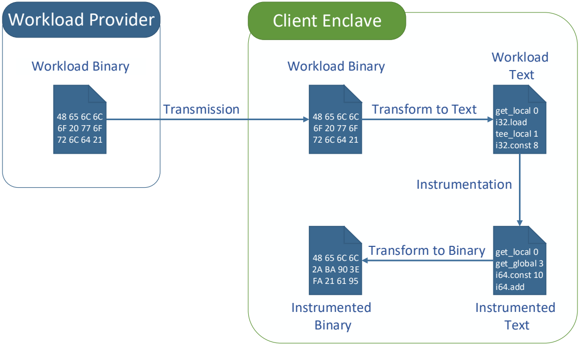
⇒ Instrumentation inside enclave!



Instrumentation

- Accounting needs to be trusted by all parties

⇒ Instrumentation inside enclave!



Workload Provider

Workload Binary

48 65 6C 6C
6F 20 77 6F
72 6C 64 21

Transmission

Client Enclave

Workload Binary

48 65 6C 6C
6F 20 77 6F
72 6C 64 21

Transform to Text

Workload Text

get_local 0
i32.load
tee_local 1
i32.const 8

Instrumentation

Instrumented Text

get_local 0
get_global 3
i64.const 10
i64.add

Transform to Binary

Instrumented Binary

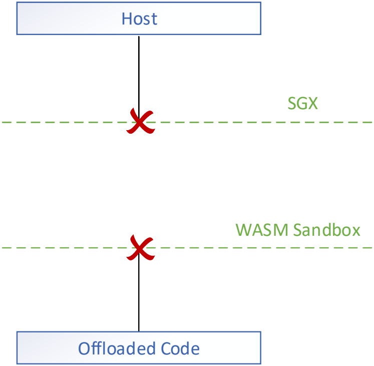
48 65 6C 6C
2A BA 90 3E
FA 21 61 95

Instrumentation

✓ Platform independent
✓ Non-forgeable

Two-sided Sandbox

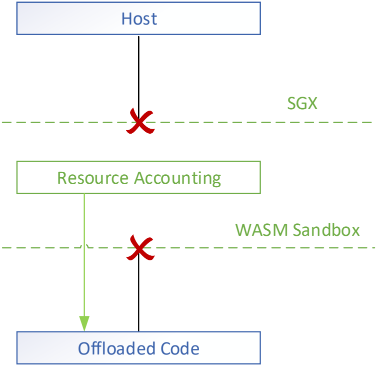
- Protection of both host and offloaded code
  - Host by sandbox
  - Code by SGX



Two-sided Sandbox

- Protection of both host and offloaded code
  - Host by sandbox
  - Code by SGX

- “Intermediate layer” protected by both
  - Code management
  - Resource accounting



Evaluation Goal & Setup

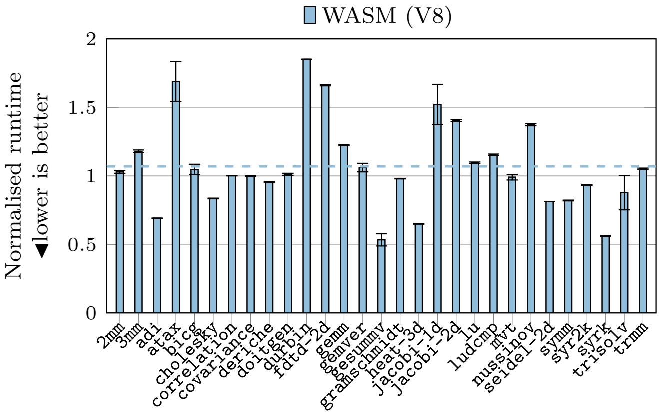
- How performant is WebAssembly?
- What overhead is introduced by
  - trusted execution (SGX)?
  - resource accounting?

- SGX protected JS-engine
  - Google's V8

- Machine:
  - Intel(R) Xeon(R) CPU E3-1230 v5 @ 3.40GHz
  - 32GB memory

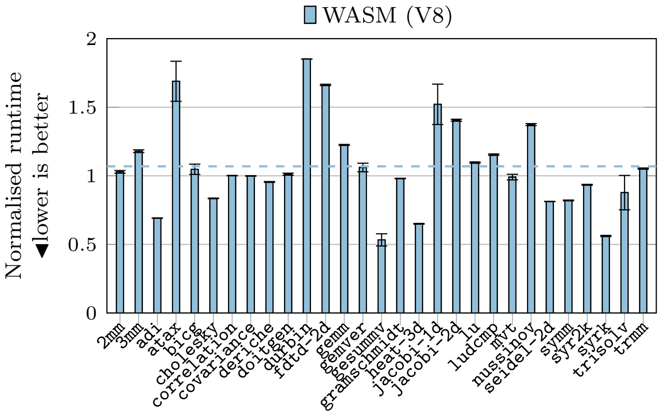
WASM Performance

- PolyBench
- Baseline:
  Native execution



WASM Performance

- PolyBench
- Baseline:
  Native execution

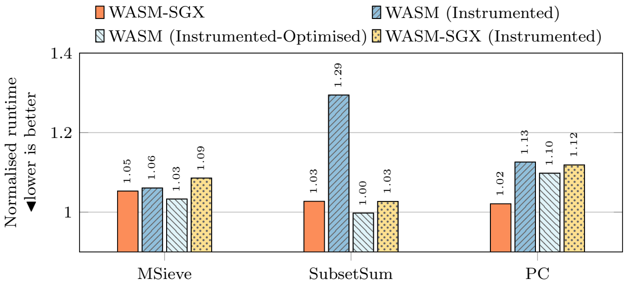


WASM Performance

Depends on application but overall comparable to native speed

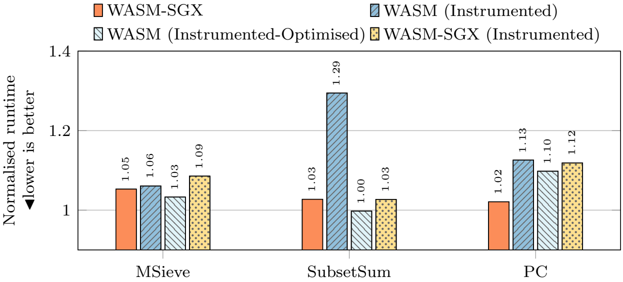
SGX and Instrumentation

- 3 volunteer computing projects
- Baseline: WASM



SGX and Instrumentation

- 3 volunteer computing projects
- Baseline: WASM



Performance Impact

- SGX: $\leq$ 5% overhead
- Instrumentation: $\leq$ 10% overhead
- Overall: $\leq$ 12% overhead

Conclusion

Trusted Execution Platform

- Application and host protected with SGX and WebAssembly
- Good performance with WASM and SGX

Resource accounting

- Platform independent with byte-code instrumentation
- Trusted by host and application provider
- Low performance impact

Conclusion

Trusted Execution Platform

- Application and host protected with SGX and WebAssembly
- Good performance with WASM and SGX

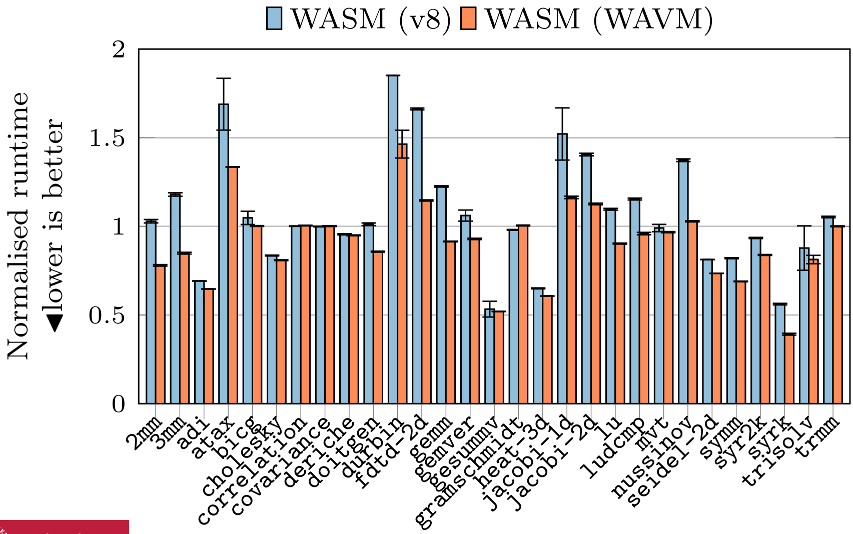
Resource accounting

- Platform independent with byte-code instrumentation
- Trusted by host and application provider
- Low performance impact

In progress

- Use cases: Serverless cloud, execution-as-payment
- Standalone WASM execution environments

Polybench



WASM (v8)
WASM (WAVM)
Normalised runtime
▼lower is better
0
0.5
1
1.5
2
2mm
3mm
adi
atax
bicg
cholesky
correlation
covariance
deriche
doitgen
durbin
fdtd-2d
gemm
gemver
gesummv
gramschmidt
heat-3d
jacobi-1d
jacobi-2d
lu
ludcmp
mvt
nussinov
seidel-2d
symm
syr2k
syrk
trisolv
trmm