EbbRT: A Framework for Building Per-Application Library Operating Systems

Dan Schatzberg, James Cadden, Han Dong, Orran Krieger, Jonathan Appavoo
Boston University
Lower Effort

Performance

Existing General Purpose Systems

Higher Effort

Performance

Custom Special Purpose Systems

Ideal
Cloud Computing allows users to rent entire virtual machines to run their own OS and applications.
Cloud Computing allows users to rent entire virtual machines to run their own OS and applications.

The OS no longer needs to provide protection and isolation.
Elastic Building Block Runtime (EbbRT)

A framework for constructing per-application Library Operating Systems in order to achieve high performance with low effort.
Construct customized environments for individual applications with reusable components.
Implementation Details

- Build bootable (by grub, kexec, qemu, etc.) ELF images using a modified GCC 5.3 toolchain
- Most code written in C++14 (~15kloc)
- Library of core system components
- Includes libc, libstdc++, boost libraries, and more
Elastic Building Block Runtime (EbbRT) Architecture
Elastic Building Block Runtime (EbbRT) Architecture

1. Low-level, event-driven execution environment
   - Allow applications to be written more directly to hardware interfaces for high performance
Elastic Building Block Runtime (EbbRT) Architecture

1. Low-level, event-driven execution environment
   - Allow applications to be written more directly to hardware interfaces for high performance

2. Heterogeneous distributed system architecture
   - Offload functionality to general purpose operating systems for compatibility
Elastic Building Block Runtime (EbbRT) Architecture

1. Low-level, event-driven execution environment
   - Allow applications to be written more directly to hardware interfaces for high performance

2. Heterogeneous distributed system architecture
   - Offload functionality to general purpose operating systems for compatibility

3. Elastic Building Blocks
   - Encapsulate system and application functionality in order to promote customization and reuse
Linux Memcached

Interrupt ➔ Device Driver ➔ Network Stack ➔ Device Driver ➔ Network Stack ➔ Memcached

Schedule thread

Context Switch

Request Data ➔ sk_buff ➔ Request Data
Linux Memcached

Interrupt → Device Driver → Network Stack → Device Driver → Network Stack → Memcached

- Schedule thread
- Context Switch

- sk_buff
- Request Data

Copy packet to userspace
Linux Memcached

Interrupts Enabled

Interrupt → Device Driver → Network Stack → Device Driver → Network Stack → Memcached

Schedule thread

Context Switch

Copy packet to userspace

sk_buff

Request Data

Request Data
General purpose operating systems introduce inefficiencies for individual applications.
• **EbbRT applications execute at highest privilege level**

• **Identity mapped memory with large pages**

• **Execution as non-preemptive events generated by hardware or software**
EbbRT Memcached

- No paging
- No complex scheduling

Interrupts Disabled

Function Call

Device Driver → Network Stack → Memcached

Zero Copy

IOBuf

Request Data
Accept Up Call

Receive Up Call

Send Down Call

Network Manager

Data

VirtIO In

VirtIO Out

Receive Queue

Direct Transmit

VirtIO Net Driver

Memory Allocator

Buffers Read and Written By VirtIO Net Device
Server running on 6 threads

Mutilate request generator w/ 7 clients

Facebook ETC workload

Using TCP

Latency (us)

Throughput (RPS)
EbbRT Memcached is able to attain >2x throughput at SLA
Other Benefits

See paper for details

- Blocking semantics
- Read-Copy-Update (RCU)
- Core-local data structures
Elastic Building Block Runtime (EbbRT) Architecture

1. Low-level execution environment
2. Heterogeneous distributed system architecture
3. Distributed object model
Reuse event-driven network stack to provide networking to Node.js.
Avoid duplicating functionality for compatibility
Offload functionality for rapid development
Elastic Building Blocks
System-wide distributed objects

Filesystem Ebb
Elastic Building Blocks

System-wide distributed objects

Representative
V8 Javascript Benchmark Suite

3000 new lines of code and two weeks
Other Benefits

See paper for details

• Distributing EbbRT as a toolchain
• Providing language and library compatibility
• Language-level primitives (Lambdas, Futures, IOBufs)
Elastic Building Block Runtime (EbbRT)

- Low-level execution environment enables applications to get much closer to the hardware
- Heterogeneous distributed system allows for functionality offload for incremental development
- Distributed object model encapsulates functionality, enabling customization and reuse
Questions?

https://github.com/sesa/ebbrt

Dan Schatzberg, James Cadden, Han Dong, Orran Krieger, Jonathan Appavoo

dschatz@bu.edu  jmcadden@bu.edu  handong@bu.edu  okrieg@bu.edu  jappavoo@bu.edu