LibrettOS: A Dynamically Adaptable Multiserver.Library OS

Ruslan Nikolaev, Mincheol Sung, Binoy Ravindran

16th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2020), March 17, 2020
Motivation

- The monolithic OS design is inadequate for modern systems
  - Lack of isolation, failure recovery, large trusted computing base (TCB)
  - Kernel-bypass libraries or library OS improve performance

[Herder et al. ACSAC'06], [Nikolaev et al. SOSP'13],
[Kantee login'14], [Lankes et al. ROSS'16], [Decky 2017]
Motivation

- The monolithic OS design is inadequate for modern systems
  - Lack of isolation, failure recovery, large trusted computing base (TCB)
  - Kernel-bypass libraries or library OS improve performance
- Multiple OS paradigms *seamlessly* integrated in the same OS are desirable
  - Application-specific requirements (performance, security)
  - Shared driver code base
  - No code rewrite (POSIX compatibility)
  - Limited physical (e.g., SR-IOV) resources
  - Dynamic switch

[Herder et al. ACSAC’06], [Nikolaev et al. SOSP’13], [Kantee login’14], [Lankes et al. ROSS’16], [Decky 2017]
Example: Server Ecosystem

- The network server for most applications
Example: Server Ecosystem

- Direct access for certain applications
Rump Kernels and Rumprun

- The concept is introduced by Antti Kantee and NetBSD community
- NetBSD code consists of anykernel components with can be used in both kernel and user space
- The rumprun unikernel is effectively a library OS
Rump Kernels and Rumprun

Pros
- Very flexible
- Reuse most of NetBSD code (both drivers and the user-space environment)
- The rump kernel part is upstreamed
- A permissive license (2-Clause BSD) for the most code

Cons
- Rumprun lacks SMP and Xen HVM support
- The unikernel model is not always suitable
LibrettOS

- Based on rumprun
  - Adds SMP and Xen HVM support
- Reuses NetBSD’s device drivers and user-space environment
- Uses the Xen hypervisor
- A more advanced OS model
  - Our prototype implements the network server
  - Applications can also directly access resources (NIC, NVMe)
  - Dynamic switch
LibrettOS Architecture

- Direct mode applications
LibrettOS Architecture

- Network server
LibrettOS Architecture

- Applications that use servers
Network Server

- A low-level design (direct L2 forwarding)
  - TCP runs in the application address space
  - A full recovery is possible as long as TCP does not time out
  - Accommodates two paradigms easily
  - A dynamic switch is feasible
- Fast IPC
  - Uses Xen-specific capabilities (e.g., shared memory, VIRQ)
  - Lock-free queues
Network Server

- The IPC channel exchanges mbufs
  - Rx/Tx lock-free ring buffers (shared memory)
  - Virtual interrupts (VIRQ)
The portmap (port-to-domain map) table is kept in Xen:
- 64K entries for TCP and 64K entries for UDP.
- Can be accessed (read-only) by the network server.
- Applications issue a port-bind hypercall.
Applications that do not need a dynamic switch, use the network server and share the same IP
Applications that need a dynamic switch, reserve a dedicated IP when connecting to the network server.

Initially their VIF redirects packets to the network server.

```
Application 1
IP 1
VIF 1

Network Server
IP 1
IP 2
IF

Application 2
IP 2
VIF 2
```
When the dynamic switch is requested, the corresponding IP is deactivated on the network server side, and the corresponding physical interface is configured.
Applications that always need direct access avoid an intermediate VIF and access the physical interface directly.
Evaluation: System Configuration

<table>
<thead>
<tr>
<th>Processor</th>
<th>2 x Intel Xeon Silver 4114, 2.20GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cores</td>
<td>10 per processor, per NUMA node</td>
</tr>
<tr>
<td>HyperThreading</td>
<td>OFF (2 per core)</td>
</tr>
<tr>
<td>TurboBoost</td>
<td>OFF</td>
</tr>
<tr>
<td>L1/L2 cache</td>
<td>64 KB / 1024 KB per core</td>
</tr>
<tr>
<td>L3 cache</td>
<td>14080 KB</td>
</tr>
<tr>
<td>Main Memory</td>
<td>96 GB</td>
</tr>
<tr>
<td>Network</td>
<td>Intel x520-2 10GbE (82599ES)</td>
</tr>
<tr>
<td>Storage</td>
<td>Intel DC P3700 NVMe 400 GB</td>
</tr>
</tbody>
</table>

Xen 4.10.1  
Linux 4.13  
NetBSD 8.0 + NET_MPSAFE  
Jumbo Frames (mtu = 9000)
Evaluation

- NetPIPE: network throughput (a ping pong benchmark)
  - 64 bytes .. 512 K
  - All systems except the original Rumprun-PV have comparable performance
Evaluation

- NFS server
  - Executing Sysbench/FileIO from the client machine
  - Direct NVMe initialized with ext3, mixed I/O
Evaluation

Nginx HTTP server

- 10,000 requests from the client side
- Concurrency 1 .. 60
- Blocks 4K .. 128K
- LibrettOS has a better performance for smaller blocks
Evaluation

- Nginx: Dynamic Switch
  - Concurrency 20
  - LibrettOS-Hybrid: 50% in direct mode and 50% in server mode
Evaluation

- Memcached (a distributed memory caching system)
  - The memcache_binary protocol
  - 1:10 of SET/GET operations (read-dominated)
  - Each thread runs 10 clients, each client performs 100,000 operations
Evaluation

- Redis (in-memory key-value store)
  - 1,000,000 SET/GET operations, 128 bytes
  - Various number of concurrent connections
Evaluation

- Failure recovery
  - One application (Nginx uses the network server)
  - Two applications: Nginx and Redis
Related Work

- Multiserver OS
  - MINIX 3 [ACSAC’06], HelenOS, QNX
- Multiserver approaches for monolithic systems
  - SawMill, VirtuOS [SOSP’13], Snap [SOSP’19]
- Kernel-bypass libraries
  - DPDK, SPDK
- Library OS approaches
  - IX [OSDI’14], Arrakis [OSDI’14]
- Unikernels
  - UKL [HotOS’19]
Conclusions

- LibrettOS is an OS that unites two models: multiserver and library OS
- LibrettOS is the first to seamless integrate these two models
  - The same driver base (inherited from NetBSD)
  - Applications do not need to be modified
- A dynamic switch is possible
  - Applications can switch from the network server to direct mode with no interruption at runtime
- Our prototype solves a number of technical challenges
  - SMP support, Xen HVM support
Availability

- LibrettOS’s source code is available at
  [http://librettos.org](http://librettos.org)
Availability

- LibrettOS’s source code is available at
  
  http://librettos.org

THANK YOU!

Artwork attribution: NetBSD, Xen, nginx, memcached, redis, 10 GEA, NVM Express logos are from Wikipedia. The rump kernel logo is from rumpkernel.org. Xen logo/mascot belongs to XenProject.org. All other logos belong to their corresponding authors and/or projects.