Performance Evaluation of OSv for Server Applications

Ian Briggs, Matt Day, Eric Eide, Yuankai Guo, Peter Marheine

ianbriggs@comcast.net, matt.day@utah.edu, eeide@cs.utah.edu, quick803@gmail.com, peterm@cs.utah.edu

School of Computing, University of Utah

Abstract

This report presents preliminary results from ongoing research into the potential performance improvements of the unikernel programming model over traditional operating systems in virtualized environments. We compare OSv to a typical Linux server in DNS and HTTP server benchmarks and find that OSv seems to provide an appreciable performance improvement even despite its relative immaturity.

1 Background

The recent rise of commodity cloud computing has motivated work in optimization of hypervisors to provide secure isolation of applications on shared infrastructure.

Unikernels such as OSv [3] are designed to further reduce the overhead of running an application in a virtualized environment by building a single-purpose virtual appliance. These systems sacrifice some of the functionality and guarantees provided by traditional operating systems in favor of reduced resource footprints and greater efficiency.

Our ongoing work involves testing the performance characteristics of several unikernels in realistic situations to evaluate the potential performance benefits of the unikernel approach. This report discusses our early results with OSv and compares to a comparable Linux server.

2 Experiment configuration

All platforms were evaluated running in a Xen [1] 4.4 guest domain on Dell r710 servers with Intel Xeon E5530 processors and 12 gigabytes of RAM in the Emulab [5] network testbed. Benchmark client programs were run on identical r710 servers connected via a dedicated gigabit Ethernet link. Both client and Dom0 operating systems were Ubuntu 14.04. This configuration is shown in Figure 1.

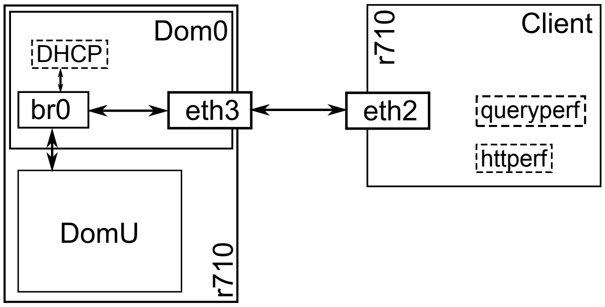

Figure 1: Experiment configuration in testbed

Guest domains were allocated one gigabyte of memory and one virtual CPU, with a virtual network interface attached to a Linux network bridge device on the host, which was in turn connected to the dedicated client-server link. The Xen Dom0 and client machine were assigned static IP addresses on this private network, while Xen DomU guests were assigned IP addresses by a DHCP server running in the Xen Dom0.

The two applications we have tested are a static HTTP server and authoritative DNS server. We benchmarked HTTP performance with httperf [2], and DNS with queryperf [4], part of the ISC BIND source distribution.

2.1 OSv

Our OSv applications were based on git HEAD as of early December 2014. The system images we compiled contained none of the OSv management tools (CLI or web interface)- only the application which was being benchmarked. OSv was run in a Xen HVM domain due to difficulties we encountered with our Xen server running a paravirtualized guest. We do not believe this to be an OSv bug.

Our DNS server is ISC BIND 9.10.1, ported to OSv.

The application itself was not modified beyond the build system changes necessary to compile a shared object executable by OSv, but we made several minor changes to OSv to compile and run the application successfully. Several C library functions were stubbed out or implemented as necessary (syslog, __strsep_g and sigsuspend). BIND provides ancillary data to the system when performing certain operations such as to set the ToS bits on outgoing UDP packets, which we found triggered assertions in the OSv implementation of the sendmsg syscall because translation of this ancillary data is not currently implemented. We modified the OSv implementation to explicitly strip this ancillary data from all packets with no ill effect on the tested server functionality.

Our HTTP server is lighttpd 1.4.35, ported to OSv. OSv already provides an HTTP server for its management interface as well as application servers in several common programming languages (Java, Ruby), but we believe none of the existing ones are appropriate for our use case of a user-configured static web server. lighttpd by comparison is a full-featured web server suitable for production use in a wide variety of applications, with the additional advantage of relative simplicity compared to other common web servers.

lighttpd requires modification only to build a OSv shared object, but we found what appears to be a threading-related bug in OSv which is exposed by lighttpd, which only seems to manifest when run under Xen. lighttpd would correctly serve a single connection, following which OSv would become completely unresponsive to any network activity (including ICMP ping) or console input. We expect to work with the OSv developers to get this bug fixed, but until that time we have tested our OSv lighttpd image under the Linux KVM hypervisor rather than Xen.

2.2 Linux

Our Linux server is based on Ubuntu Server 14.04 for x86_64 in a paravirtualized Xen domain. The DNS server is BIND 9.8.4 as packaged by the distribution, reconfigured with our domain file. The HTTP server is Apache 2.4.7, again packaged by the distribution and run in the default configuration.

3 Benchmark results

3.1 DNS

Our DNS benchmark made queries to the server for a single name for which the server was configured as the authoritative name server. We increased the request rate from a base rate of 1000 requests per second until the server's response rate no longer scaled with the request rate.

The results of these tests are shown in Figure 2 and Figure 3, with captured response rate and response latency statistics.

We see that even running the same server application (BIND in both cases), OSv outperforms Linux by a significant margin. While Linux can sustain a response rate of about 19000 per second, OSv can handle approximately 28000 requests per second. OSv's typical latency is also slightly lower.

3.2 HTTP

The results of our HTTP benchmarks are shown in Figure 4. Shaded regions on the graph represent an increase of one standard deviation from the average latency for each given request rate.

The Apache+Linux server performs well until about 4,000 requests/second, at which point it becomes unable to keep up with the requests- the response rate drops dramatically and response latency increases by several orders of magnitude

lighttpd on OSv consistently well all the way through 5,000 requests/second. We speculate that the increase in response latency near 5000 requests per second is due to client resource exhaustion, described below, rather than representing the limits of the server's capacity.

We were unable to test higher than about 5,000 requests/second using a single test client because httperf creates a new TCP connection for each HTTP request, in order to simulate real-world requests coming from different browsers. This exhausts the client's pool of available TCP connections when running at high connection rates because closed sockets must linger in the TIME_WAIT state. Our tests were run for ten seconds, so at 5000 requests per second there will be 50000 previously-open sockets which cannot be reused for nearly a minute. httperf's performance under these conditions becomes exceedingly poor, because it must wait a long time to get a new socket from the kernel.

3.3 Memory pressure

In addition to simple load testing of the HTTP servers, we also tested them under increased memory pressure. We reduced the memory allocated to each machine by a factor of two from the base of one gigabyte until the system failed to run correctly or performance was reduced significantly.

The Linux server performed normally down to 256 MB of memory, the minimum amount specified for Ubuntu Server. OSv performed normally to that point,

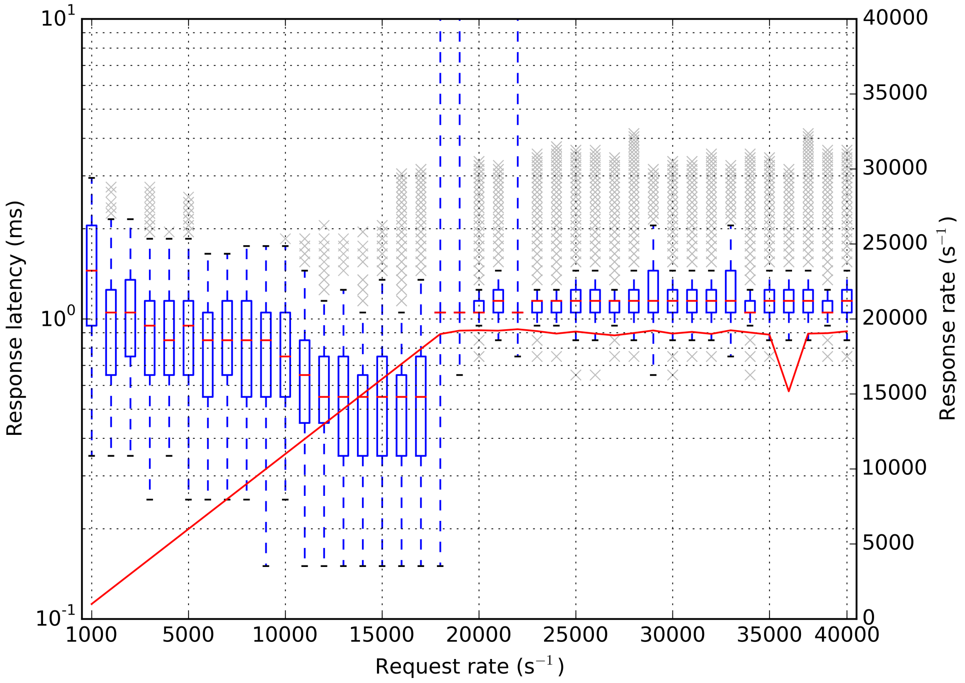

Figure 2: Linux DNS server results.

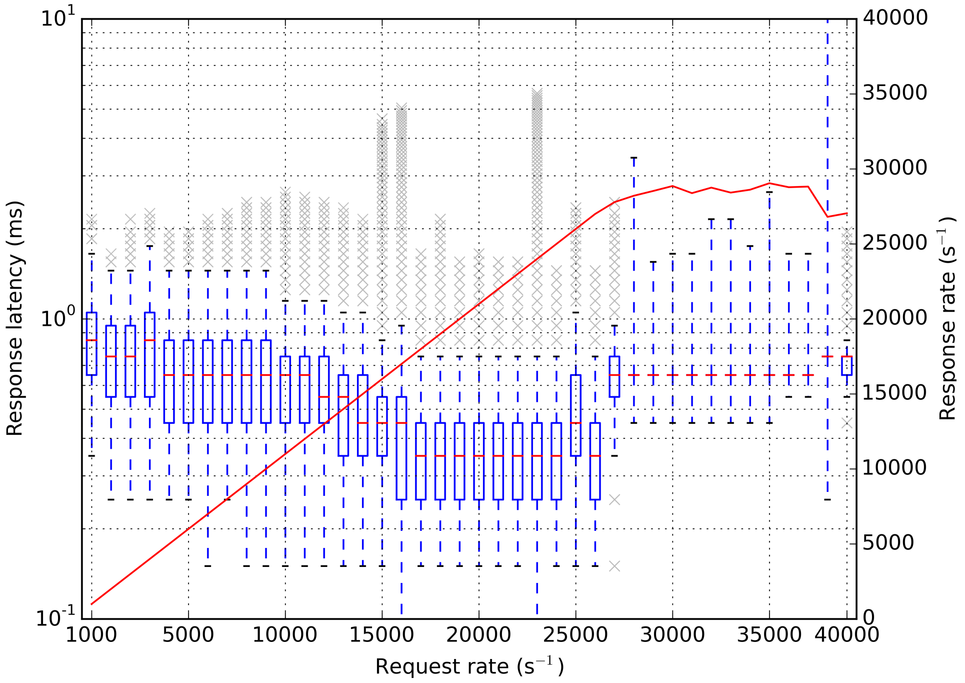

Figure 3: OSv DNS server results.

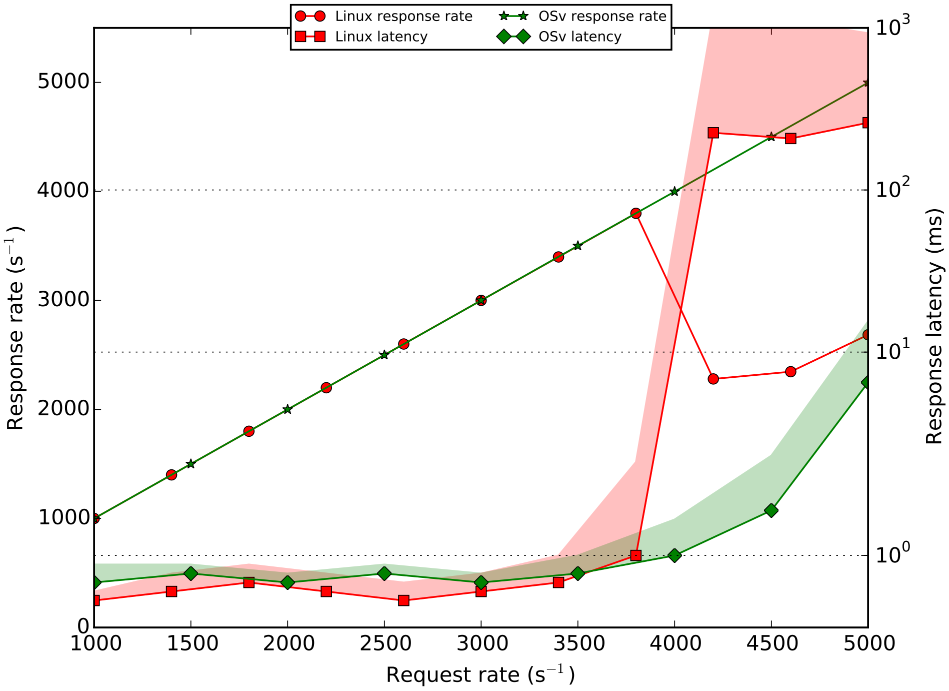

Figure 4: HTTP Server Benchmark Results

and reached an out-of-memory condition when servicing 4000 requests per second with 128 MB of memory.

4 Conclusions

We find here that OSv's performance with simple ports of common Linux server applications is superior to Linux itself in these tests, suggesting that OSv (and possibly other unikernels) could be attractive to server operators seeking to improve performance. Small modifications to these servers could improve performance further, such as allowing BIND to use the OSv APIs for zero-copy network I/O.

The relative difficulty in porting the applications (possibly beyond the skills of most server administrators) and bugs we found, however, suggest that caution is appropriate. OSv is still immature (indeed, our tests were run on an "alpha" version of OSv), so we expect these issues will be addressed in the future, making the system more appropriate to general use.

The architecture of OSv seems well-suited to these applications, because it incurs remarkably low overhead in common operations. Everything in an OSv instance runs in a shared address space so the overhead of context switches as found in full-featured operating systems does not exist, be it between different processes or between user- and kernel-space. This suggests that particular care must be taken to ensure that applications are not vulnerable to remote attack, since compromise of any application in an OSv instance can give complete control to an attacker. In the virtual appliance use case however (where any given virtual machine runs only one application), this is no more a concern than it is on other operating systems.

References

[1] BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I., AND WARFIELD, A. Xen and the art of virtualization. In *Proc. of the 19th ACM Symposium on Operating Systems Principles* (2003), pp. 164–177.

[2] HEWLETT-PACKARD RESEARCH LABORATORIES. httperf tool for measuring web server performance. http://www.hpl.hp.com/research/linux/httperf/.

[3] KIVITY, A., LAOR, D., COSTA, G., ENBERG, P., HAR'EL, N., MARTI, D., AND ZOLOTAROV, V. OSv: Optimizing the operating system for virtual machines. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference* (2014), pp. 61–72.

[4] NOMINUM, INC. queryperf dns query performance testing tool. http://ftp.isc.org/isc/bind9/9.9.0rc1/bind-9.9.0rc1/contrib/queryperf/.

[5] WHITE, B., LEPREAU, J., STOLLER, L., RICCI, R., GURUPRASAD, S., NEWBOLD, M., HIBLER, M., BARB, C., AND JOGLEKAR, A. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation* (Boston, MA, Dec. 2002), USENIX Association, pp. 255–270.