Arrakis: The Operating System is the Control Plane

Simon Peter, Jialin Li, Irene Zhang, Dan R. K. Ports, Doug Woos, Arvind Krishnamurthy, Thomas Anderson, Timothy Roscoe*

*University of Washington, *ETH Zurich*

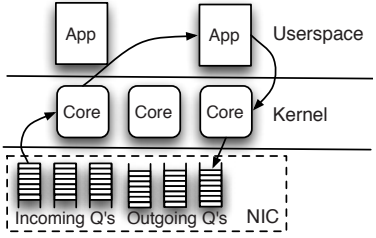
Symposium on Operating Systems Design and Implementation (OSDI), October 2014

Motivation

- Fast, low-latency hardware: Ethernet, SSD
- I/O-intensive applications
- OS on critical (data) path

Motivation

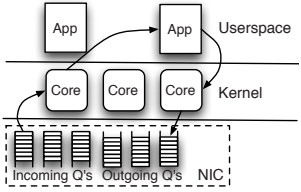
- Fast, low-latency hardware: Ethernet, SSD
- I/O-intensive applications
- OS on critical (data) path



Motivation

		Receiver Running		CPU Idle	
Network Stack	in	1.26	(37.6 %)	1.24	(20.0 %)
	out	1.05	(31.3 %)	1.42	(22.9 %)
Scheduler		0.17	(5.0 %)	2.40	(38.8 %)
Copy	in	0.24	(7.1 %)	0.25	(4.0 %)
	out	0.44	(13.2 %)	0.55	(8.9 %)
Kernel Crossing	return	0.10	(2.9 %)	0.20	(3.3 %)
	syscall	0.10	(2.9 %)	0.13	(2.1 %)
Total		3.36	($\sigma = 0.66$)	6.19	($\sigma = 0.82$)

UDP echo: 1,024 Byte payload, 1,000 samples, times in μs



Goal

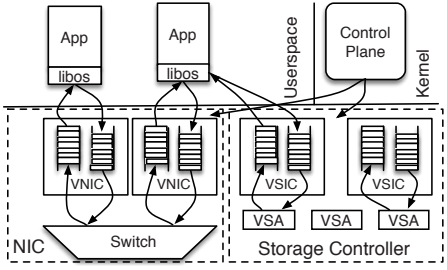
- Direct application-level access to I/O devices
- Maintain properties of classical server OS: isolation, rate limiting, global naming

Goal

- Direct application-level access to I/O devices
- Maintain properties of classical server OS: isolation, rate limiting, global naming
- Use hardware virtualisation mechanisms
  - SR-IOV: virtual PCI devices with dedicated resources
  - Standard for NIC, expected for RAID
  - Memory protection via IOMMU
- Support only policies that hardware can implement efficiently

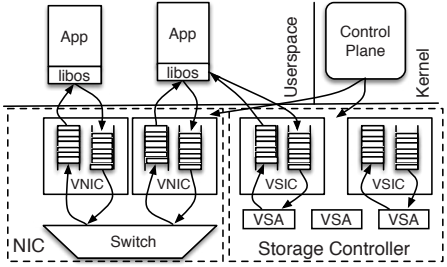
Virtual interface cards:

- Managed by control plane
- Dedicated queues and rate limiters
- Virtual network interface card (VNIC)
  - Transmit and receive filters
  - Optional support for offloading
- Virtual storage interface controller (VSIC)
  - Virtual storage areas (VSA)
  - Emulated by dedicated CPU core



Arrakis — Control Plane

- Provides VICs, filters, VSAs, rate specifiers
- IPC endpoints ("doorbells") associated to events on VIC
- Virtual file system interfaces with directories exported by applications



Arrakis — Data Plane

- Applications use (virtual) hardware directly
- Asynchronous operations, doorbells signal completion
- Optimised native interfaces with POSIX-compatibility layer on top
- Network
  - Send/Receive packet to/from queue
  - *Extaris*: user-level network stack partly based on *lwIP*
- Storage
  - Read/Write/Flush in a VSA; any offset, arbitrary size
  - *Caladan*: library of persistent data structures (log, queue)

Evaluation

- Implemented by extending Barrelfish
- 6 machine cluster for evaluation
  - 6-core Intel Xeon E5-2430 (Sandy Bridge) @ 2.2 GHz
  - Intel X520 (82599-based) 10Gb Ethernet adapter
  - Intel MegaRAID RS3DC040 RAID controller
  - 100GB Intel DC S3700 SSD
- One machine executes application, others act as clients
- "Tuned" Ubuntu Linux 13.04 (Linux 3.8)

UDP Echo

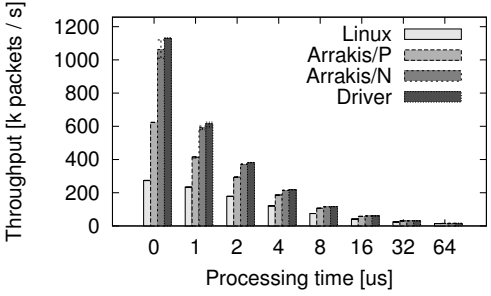
(1,024 Byte payload, 1,000 samples, times in μs)

		Linux				Arrakis			
		Receiver Running		CPU Idle		POSIX		Native	
Network stack	in	1.26	(37.6 %)	1.24	(20.0 %)	0.32	(22.3 %)	0.21	(55.3 %)
	out	1.05	(31.3 %)	1.42	(22.9 %)	0.27	(18.7 %)	0.17	(44.7 %)
Scheduler		0.17	(5.0 %)	2.40	(38.8 %)	—		—	
Copy	in	0.24	(7.1 %)	0.25	(4.0 %)	0.27	(18.7 %)	—	
	out	0.44	(13.2 %)	0.55	(8.9 %)	0.58	(40.3 %)	—	
Kernel crossing	return	0.10	(2.9 %)	0.20	(3.3 %)	—		—	
	syscall	0.10	(2.9 %)	0.13	(2.1 %)	—		—	
Total		3.36	($\sigma = 0.66$)	6.19	($\sigma = 0.82$)	1.44	($\sigma < 0.01$)	0.38	($\sigma < 0.01$)

UDP Echo

(1,024 Byte payload, 1,000 samples, times in μs)

		Linux				Arrakis			
		Receiver Running		CPU Idle		POSIX		Native	
Network stack	in	1.26	(37.6 %)	1.24	(20.0 %)	0.32	(22.3 %)	0.21	(55.3 %)
	out	1.05	(31.3 %)	1.42	(22.9 %)	0.27	(18.7 %)	0.17	(44.7 %)
Scheduler		0.17	(5.0 %)	2.40	(38.8 %)	—		—	
Copy	in	0.24	(7.1 %)	0.25	(4.0 %)	0.27	(18.7 %)	—	
	out	0.44	(13.2 %)	0.55	(8.9 %)	0.58	(40.3 %)	—	
Kernel crossing	return	0.10	(2.9 %)	0.20	(3.3 %)	—		—	
	syscall	0.10	(2.9 %)	0.13	(2.1 %)	—		—	
Total		3.36	($\sigma = 0.66$)	6.19	($\sigma = 0.82$)	1.44	($\sigma < 0.01$)	0.38	($\sigma < 0.01$)



Redis

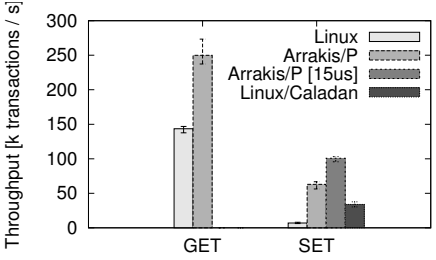
(64k random keys, 1,024 Byte value size, 1,000 samples, times in μs)

	Read Hit				Durable Write			
	Linux		Arrakis/P		Linux		Arrakis/P	
`epoll`	2.42	(27.91 %)	1.12	(27.52 %)	2.64	(1.62 %)	1.49	(4.73 %)
`recv`	0.98	(11.30 %)	0.29	(7.13 %)	1.55	(0.95 %)	0.66	(2.09 %)
Parse Input	0.85	(9.80 %)	0.66	(16.22 %)	2.34	(1.43 %)	1.19	(3.78 %)
Lookup/Set Key	0.10	(1.15 %)	0.10	(2.46 %)	1.03	(0.63 %)	0.43	(1.36 %)
Log Marshaling	—		—		3.64	(2.23 %)	2.43	(7.71 %)
`write`	—		—		6.33	(3.88 %)	0.10	(0.32 %)
`fsync`	—		—		137.84	(84.49 %)	24.26	(76.99 %)
Prepare Response	0.60	(6.92 %)	0.64	(15.72 %)	0.59	(0.36 %)	0.10	(0.32 %)
`send`	3.17	(36.56 %)	0.71	(17.44 %)	5.06	(3.10 %)	0.33	(1.05 %)
Other	0.55	(6.34 %)	0.46	(11.30 %)	2.12	(1.30 %)	0.52	(1.65 %)
Total	8.67	($\sigma = 2.55$)	4.07	($\sigma = 0.44$)	163.14	($\sigma = 13.68$)	31.51	($\sigma = 1.91$)

Redis

(64k random keys, 1,024 Byte value size, 1,000 samples, times in μs)

	Read Hit				Durable Write			
	Linux		Arrakis/P		Linux		Arrakis/P	
`epoll`	2.42	(27.91 %)	1.12	(27.52 %)	2.64	(1.62 %)	1.49	(4.73 %)
`recv`	0.98	(11.30 %)	0.29	(7.13 %)	1.55	(0.95 %)	0.66	(2.09 %)
Parse Input	0.85	(9.80 %)	0.66	(16.22 %)	2.34	(1.43 %)	1.19	(3.78 %)
Lookup/Set Key	0.10	(1.15 %)	0.10	(2.46 %)	1.03	(0.63 %)	0.43	(1.36 %)
Log Marshaling	—		—		3.64	(2.23 %)	2.43	(7.71 %)
`write`	—		—		6.33	(3.88 %)	0.10	(0.32 %)
`fsync`	—		—		137.84	(84.49 %)	24.26	(76.99 %)
Prepare Response	0.60	(6.92 %)	0.64	(15.72 %)	0.59	(0.36 %)	0.10	(0.32 %)
`send`	3.17	(36.56 %)	0.71	(17.44 %)	5.06	(3.10 %)	0.33	(1.05 %)
Other	0.55	(6.34 %)	0.46	(11.30 %)	2.12	(1.30 %)	0.52	(1.65 %)
Total	8.67	($\sigma = 2.55$)	4.07	($\sigma = 0.44$)	163.14	($\sigma = 13.68$)	31.51	($\sigma = 1.91$)



Conclusion

- Control plane configures hardware to enforce policies
- Direct application access to (virtualised) hardware
- Specialised, application-level I/O stacks
- Significant performance improvements
- Merged back into Barrelfish mainline