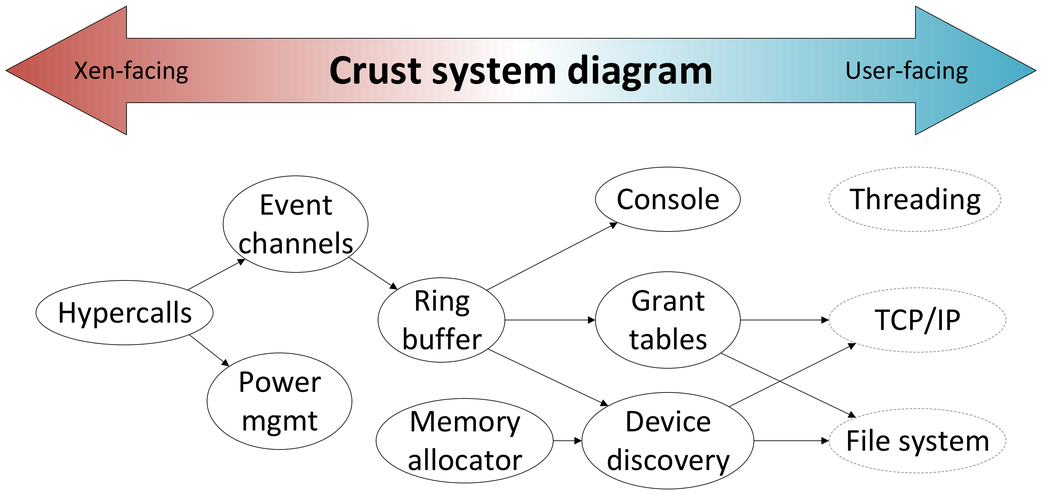
Crust

A unikernel for fast, secure cloud computing

Scott Cooper, Satyajit Kanetkar, and Carl Patenaude-Poulin
Supervised by Muthucumaru Maheswaran



Project Scope

- Initially: build on top of existing solutions (MiniOS, Newlib)
- Later: build from scratch
- **Iterative refinement of scope**

Design Strategy

- Modular design
- Type-driven design and development
- Abstract away Xen for portability

Challenges

- Cross-compiling
- Remote debugging
- Rust-specific constraints

Testing

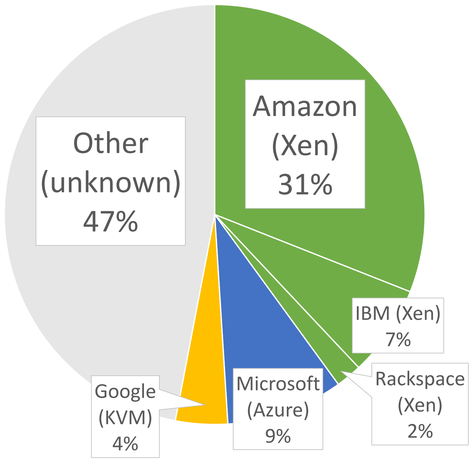
- Observable effects
- Remote GDB debugging
- “Printf debugging”
- Memory stress-testing

Target: Xen

Unikernels run on special software systems called **hypervisors.**

One hypervisor is **Xen**, used by Amazon to run their massive cloud infrastructure (>1M servers worldwide).

Cloud provider market share



Source: *AWS Remains Dominant Despite Microsoft and Google Growth Surges* (Synergy Research Group)

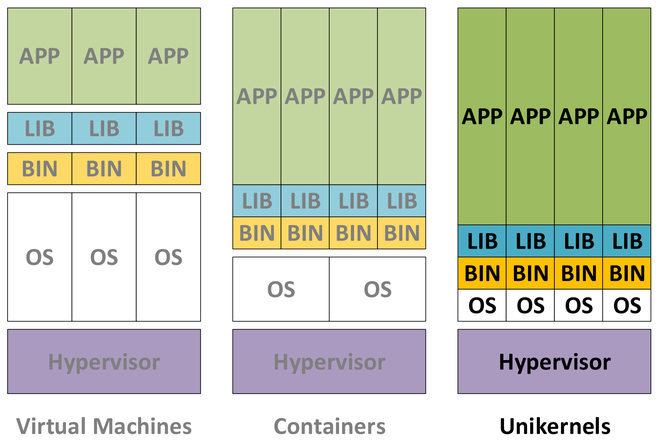
Prior Work

- **MiniOS:** Reference unikernel implentation for Xen
- **Redox**: Unix-like operating system written in Rust

Unikernels

Scalable, secure cloud computing

The application, runtime, and operating system are statically compiled into a single binary.



Compiled binaries are then deployed on rented cloud infrastructure.

Why Unikernels?

Improved security

- Hardware process isolation
- Reduced attack surface (compared to VMs and containers)

Improved scalability

- Instant boot and shutdown – pay only for the processor time you actually need
- Fewer layers at runtime = reduced overhead of context-switching

Rust

A systems language for safety, concurrency, and speed

Why Rust?

Full control on memory

- No garbage collector
- Lay out data structures in memory
- Custom memory allocators

Affine types

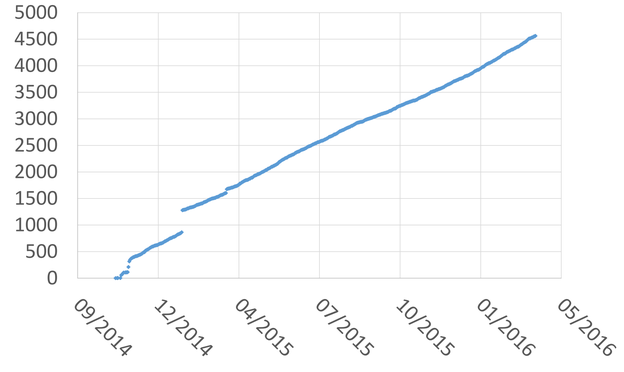
Building on developments in non-classical logic, affine types provide strong safety guarantees at **zero runtime cost**:

- Memory safety
- Data race-freedom

A vibrant ecosystem

Build and publish your code with *Cargo*, Rust’s build system and package manager. Leverage the thousands of Rust libraries available for free on *crates.io*.

Rust libraries published on crates.io



Modular standard library

Operating systems developers can get started with *libcore*, Rust’s library for bare-metal development.

Crust-OS/Crust-OS