The State of the Barrelfish Project

David Cock

September 8, 2015
Contents

1. Highlights
   Publications
   Changes in Barrelfish
   Joint Workshop

2. The Project
   Members & Contributors
   How Do We Organise?
   How Do We Write Code?
   Projects

3. Research Direction
   Rack-Scale Computing
   Memory
   Consensus
   Communication
   Formal Methods
Papers

- Gerber et. al., “Not your parents physical address space”, HOTOS’15
- Kästle et. al., “Shoal: Smart Allocation and Replication of Memory For Parallel Programs”, USENIX ATC’15
- Peter et. al., “Arrakis: The Operating System is the Control Plane”, OSDI’14
- Zellweger et. al., “Decoupling Cores, Kernels, and Operating Systems”, OSDI’14
- Baumann et. al., “Cosh: Clear OS Data Sharing In An Incoherent World”, TRIOS’14
Posters

• Kästle et. al., “Shoal: smart allocation and replication of memory for parallel programs”, EUROSYS’15
• Shinde at. al., “Intelligent NIC Queue Management in the Dragonet Network Stack”, EUROSYS’15
• Hoffman, “Rack - aware operating systems”, EUROSYS’15
Barrelfish Releases

10 releases so far this year!

Highlights

- Contributor sign-off process.
- Large page support (thanks HP).
- Many Xeon Phi improvements.
- Initial ARMv8 code.
- Arrakis.
- Overhauled build system.
Arrakis Is Mainline

Peter et. al., OSDI’14

- Barrelish repository is now the canonical source.
- Mostly thanks to Simon Gerber.
- Includes e10k support.
FM/Security Workshop

We had a day of talks with Prof. Basin’s group.

- Several areas of common interest: Authority (caps) & consensus.
- New projects starting up.
Contents

1 Highlights
   Publications
   Changes in Barrelfish
   Joint Workshop

2 The Project
   Members & Contributors
   How Do We Organise?
   How Do We Write Code?
   Projects

3 Research Direction
   Rack-Scale Computing
   Memory
   Consensus
   Communication
   Formal Methods
Barrelfish Facts

- 9 architectures.
- 42 applications (+51 test apps).
- 63 libraries.
- 9 languages.
- 32 committers.
- 8 years old.
- > 1.1M lines of code.
Reto Achermann, David Cock, Simon Gerber, Moritz Hoffman, Stephan Kästle, Timothy Roscoe, Pravin Shinde, Gerd Zellweger.

Kornilios Kourtis now at IBM Zürich.
External Members and Contributors

University of Washington
Simon Peter

Microsoft Research, Redmond
Andrew Baumann

Microsoft Research, Silicon Valley
Paul Barham, Rebecca Isaacs, Vijayan Prabhakaran

Microsoft Research, Cambridge
Richard Black, Tim Harris, Orion Hodson, Ross McIlroy

Hewlett Packard
Various
Barrelfish is now a big project:

- Lots of contributors.
- *Lots* of code.
- More and more users.
Barrelfish is now a big project:

• Lots of contributors.
• Lots of code.
• More and more users.
• Lots of engineering and support work now.
Barrelfish is now a big project:

- Lots of contributors.
- *Lots* of code.
- More and more users.
- Lots of engineering and support work now.
- But it’s still a *research* project!
Heavily driven by individual projects

- **Arrakis** was driven by Simon Peter at UW.
- **SKB** — Adrian Schüpbach (PhD).
- **Distributed cap system** — Mark Neville & Simon Gerber (Master).
Setting the Direction

Heavily driven by individual projects

- Arrakis was driven by Simon Peter at UW.
- SKB — Adrian Schüpbach (PhD).
- Distributed cap system — Mark Neville & Simon Gerber (Master).

But there is an overall direction

- Heterogeneous hardware.
- Highly distributed machines.
Day-to-day Management

- Mothy is the Professor.
- Day-to-day organisation (meetings, releases, …) mostly the ranking Post-Doc (currently me).
- Students work mostly independently, in a branch.
- Regular communication:
  - Compulsory weekly group meeting.
  - 15M stand-up meeting 3/week.
  - Everyone idles in IRC.
External Contributions

A lot of work is done outside ETH:
- Former SG members e.g. Andrew Baumann (MSR), Simon Peter (UW).
- External contributors e.g. HP Bangalore — ARMv8.

Contribution Process

1. Clone the latest public release.
2. Work in a branch.
3. Send a diff, with no proprietary or GPL code.
4. Internal comitter signs off and merges.
Changes & Questions

• What is core infrastructure (libc)? Who maintains it?
• Should bug tracking be public? How?
• Who is responsible for bugs?
Repositories & Branches

- Private git repository.
- Pushed to public repo on releases.
- Shared repos for collaborators (e.g. Huawei).
- Repos hosted in Fabricator.
**Bug Tracking**

- Managed in Fabricator.
- Integrated with repository commits.
Bug Tracking

- Managed in Fabricator.
- Integrated with repository commits.
- Not public.
Continuous Integration

- Barrelfish is tested nightly.
- Release only when tests are green.
- Current system is creaky.
- Bitten seems to be unsupported.
Continuous Integration

- Barrelfish is tested nightly.
- Release only when tests are green.
- Current system is creaky.
- Bitten seems to be unsupported.

Replacement CI tool

- Supported.
- Test private branches.
- Easy to configure plenty of tests.
Technical Debt

From the Barrellfish source:
Technical Debt

From the Barrelfish source:

XXX: workaround for backwards compatibility: prepend default path
XXX: workaround to allow working around the previous workaround
Technical Debt

From the Barrelfish source:

XXX: workaround for backwards compatibility: prepend default path
XXX: workaround to allow working around the previous workaround

$ rgrep ‘TODO’ kernel — wc -l
91
Technical Debt

From the Barrelfish source:

XXX: workaround for backwards compatibility: prepend default path
XXX: workaround to allow working around the previous workaround

$ rgrep 'TODO' kernel — wc -l
91

$ rgrep 'voodoo' kernel — wc -l
1
Projects

- **ARMv8 (64-bit)**
  - Huawei grant.
  - Much more like an x86 server.
- **UEFI (Hagfish)**
  - UEFI required for ARMv8 servers.
  - x86 is going this way anyway.
- **Build System (Hake)**
  - Was really slow → faster now.
  - Still questions about platform configuration.
- **Infiniband & RDMA**
  - This is how rack-scale machines communicate.
  - Extend the multikernel idea (can’t share state).
Student Projects

- Go, Martynas. Integration with LMP/UMP channels.
- Dynamic Libraries, David.
- Graphene, Yves.
- RDMA Capabilities, Benjamin.
Contents

1 Highlights
   Publications
   Changes in Barrelfish
   Joint Workshop

2 The Project
   Members & Contributors
   How Do We Organise?
   How Do We Write Code?
   Projects

3 Research Direction
   Rack-Scale Computing
   Memory
   Consensus
   Communication
   Formal Methods
Single System Image

- The Datacenter as a secure, programmable machine.
- Run one OS ‘image’ on a whole rack.
- Memory management.
- Coordination & Consensus.
- Communication.
Memory

Cichlid
Self-paging with exposed physical addresses.
  • Submitted to SOSP’15, didn’t get up.
  • Develop and resubmit to Eurosys’16.

Shoal
  • NUMA and cache-aware allocation.
  • Even more important if we’re doing RDMA.
Consensus

Collaboration with InfSec

• Consensus protocols.
• High-level security models (using caps).

Multilevel Broadcast

• What’s the fastest way to communicate on a multicore?
Communication

Infiniband & RDMA

- How to do security? Caps?
- Sensible abstractions.

Dragonet

- The machine is a network.
- Use the hardware efficiently.
Machine Representation

Memory

Consensus

Communication
Machine Representation

What does the machine look like?

- We need a way to talk about these things.
- Preferably a single way.
- We need some sort of ontology.

Update the SKB

- This is what it was designed for.
- Extend its capabilities (NUMA placement, miniSAT).
Formal Methods

We’ve got an MLoC research project, with some code quality issues. Can FM help?
Formal Methods

We’ve got an MLoC research project, with some code quality issues. Can FM help?

Apply FM where there’s low effort, or high benefit.
- Automatic static analysis e.g. Goanna (low effort).
- Concurrency (lots of bugs).
- Hardware interface (lots of bugs, bad models).
Questions?