Open Source on the Modern Mainframe

Elizabeth K. Joseph
@pleia2

Hello! My name is Lyz.

My background is as a Linux Systems Administrator, a role I held for almost 15 years.

I've been involved with open source projects for nearly 20 years!

Open source is the common thread in my systems and operations career.

I'm speaking as a community member interested in open source on the mainframe :)

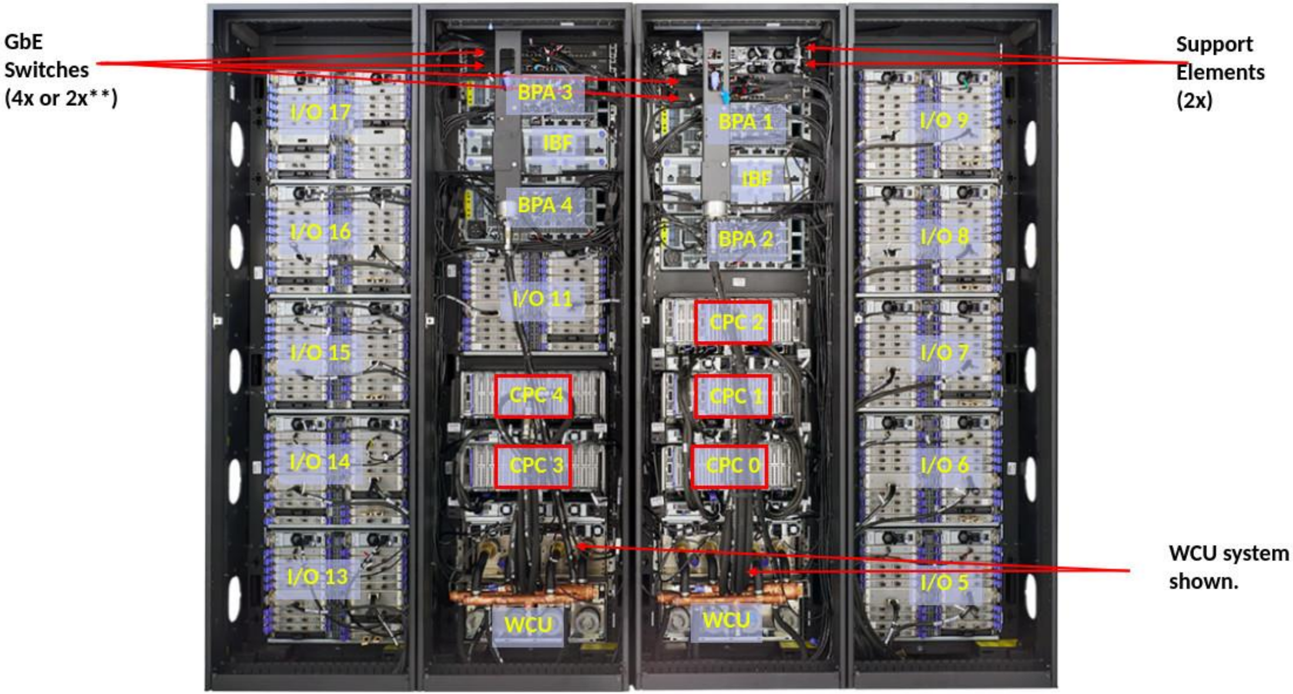
This is a lightning talk, we'll go fast!

Slides will be available on https://princessleia.com (maybe they're up already!)

Questions? Comments? Please reach out! lyz@princessleia.com

Tweet at me! @pleia2

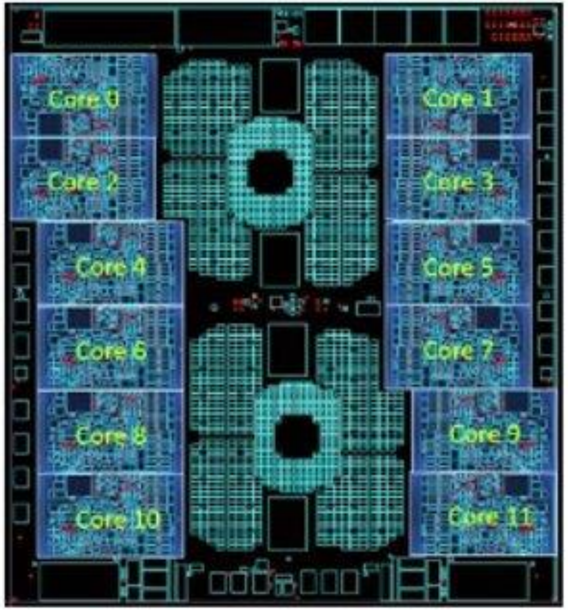
What is a mainframe?



What is a mainframe?

Not x86.

(it's s390x!)



190 5.2ghz processor units (PUs), with 12 cores per chip

But also...

- 40TB of RAM
- 60 PCIe control units across 12 PCIe I/O drawers
- 22 dedicated I/O offload processors (SAPs) pre-allocated per system

What do mainframes run?

- Linux! (SLES, RHEL, Ubuntu...)
- A ton of open source software!
- Several proprietary mainframe operating systems (but we're here to talk about open source!)

Open Source Legacy

In 1955, the volunteer-run SHARE Inc was founded.

A key resource for this organization was the SHARE library of software that systems programmers would share among their peers, freely.

In 1959, SHARE released the SHARE Operating System (SOS), one of the first true "operating systems"[1] and Wikipedia says of SOS:

"SOS was one of the first instances of "commons-based peer production" now widely used in the development of free and open-source software such as Linux and the GNU project."

SHARE

EDUCATE • NETWORK • INFLUENCE

[1] https://en.wikipedia.org/wiki/SHARE_(computing)

History of Linux on the mainframe

Started out as the "Bigfoot" (i370) port by several community members in 1998-99.

IBM released the first Linux kernel patches to support s390x in December 1999.

In October 2000, SUSE Linux Enterprise Server became the first, still in production, enterprise Linux to support s390x.

Red Hat quickly followed as the second, still in production, enterprise Linux for the mainframe.

Ubuntu support was announced in 2016 and began with Ubuntu 16.04.

Open Source for Linux

Linux Distributions & Virtualization

- Red Hat
- ubuntu
- SUSE
- KVM

Community Versions

- debian
- openSUSE
- fedora
- ClefOS
- alpine linux

Networking & Monitoring

- NGINX
- HAPROXY
- Apache MESOS
- Prometheus
- ZABBIX
- APACHE ZooKeeper
- etcd

Cloud & Container Services

- docker
- LXD
- openstack.
- kubernetes
- minikube
- HELM
- Istio
- okd
- OPENSHIFT
- APACHE HTTP SERVER PROJECT
- MARATHON
- docker Compose
- Sysdig
- Terraform
- kata containers
- podman

Languages & Runtimes

- Java
- JS
- R
- python
- Open Liberty
- Ruby
- Scala
- GO
- node JS
- pypy
- ERLANG
- php
- RAILS
- OCaml
- OpenJ9
- OpenJDK
- Apache Tomcat
- Groovy
- Clojure
- HYPERLEDGER FABRIC
- WildFly
- TensorFlow

DevOps/Automation

- CHEF
- ANSIBLE
- puppet
- Jenkins
- APACHE JMeter™
- Travis CI
- ANTLR
- Maven
- sonarqube
- Gradle
- SALTSTACK

Middleware

- Apache ACTIVEMQ
- APACHE Camel
- muleESB
- RabbitMQ
- APACHE GEODE
- mosquitto
- HIBERNATE

Big Data, Observability, Analytics

- Flink
- splunk>
- APACHE Spark
- Apache Solr
- apache Ignite
- APACHE kafka
- fluentd
- Grafana
- logstash
- elasticsearch
- kibana

Databases

- CouchDB relax
- MariaDB
- mongoDB.
- RethinkDB
- redis
- PostgreSQL
- MySQL.
- cassandra
- SCYLLA
- Couchbase

Open Source Hardware support on Linux

You may be tempted to believe there's proprietary "sauce" for the hardware encryption support on Linux on the mainframe, but it's just the standard open source libraries and tools we know and love!

- dm-crypt
- OpenSSL and libcrypto (including for ssh, scp, sftp, Apache mod_ssl...)
- IPSec
- libica crypto library for s390x (https://github.com/opencryptoki/libica)
- aes for Go (https://golang.org/pkg/crypto/aes/)

Open Source Docker Images

Linux Distributions

- alpine Linux
- ubuntu
- ClefOS
- debian
- fedora

Networking & Monitoring

- Nextcloud
- REDMINE
- HAPROXY
- APACHE ZooKeeper
- jetty://
- BASH

Cloud , Web, Languages & Runtimes

- docker
- pypy
- python
- GCC
- Ruby
- JRuby
- Open Liberty
- GO
- node JS
- Perl
- php
- OpenJDK
- APACHE HTTP SERVER PROJECT
- BusyBox
- Groovy
- ownCloud
- elixir
- Apache Tomcat
- WordPress
- ghost
- Joomla!
- Drupal
- ERLANG

DevOps/Automation

- Maven
- Gradle
- BUILDPACK
- COMPOSER

Middleware

- RabbitMQ
- mosquitto
- express gateway
- Flink

Big Data, Observability, Analytics

- Matomo
- Apache Solr
- fluentd
- PIWIK
- phpMyAdmin

Databases

- mongoDB
- redis
- PostgreSQL
- Adminer
- Memcached
- STORM

Open Mainframe Project

- Full project hosting, including code, and mailing lists
- Blogs and podcasts of general interest to the open source mainframe community
- Slack and forums for communication among participants
- Project support for 3rd party open source projects seeking infrastructure (VMs, CI/CD services)
- New in 2020: Annual conference!

OPEN MAINFRAME PROJECT

Software Discovery Tool

- Just launched in 2020!
- First release will search SLES, RHEL, and Ubuntu for open source packages built for the mainframe
- Learn more and contribute at https://github.com/openmainframeproject/software-discovery-tool

OPEN **MAINFRAME** PROJECT
Software Discovery Tool

Application Modernization with Zowe

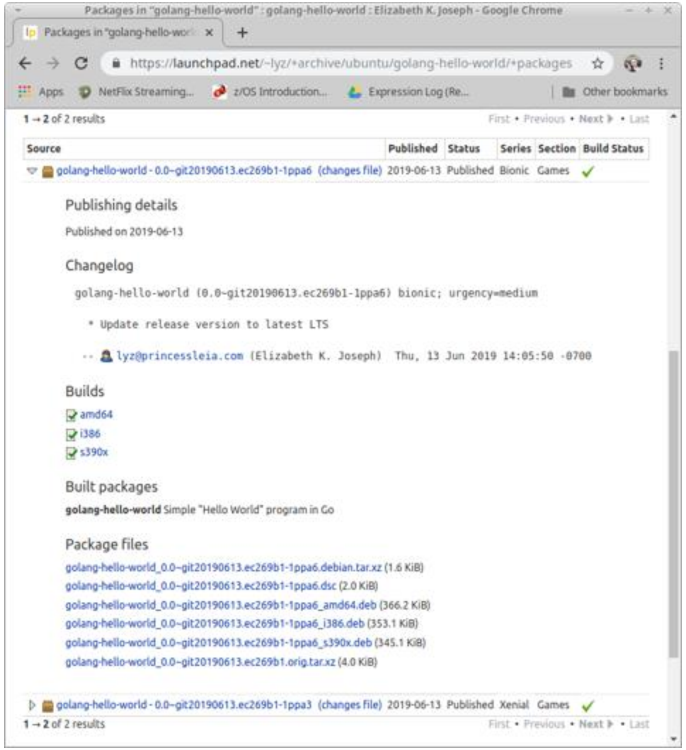
At the core, Zowe provides an API that allows easier, unified access to the mainframe.

This has allowed an ecosystem of tools to flourish.

- A CLI that can be installed on any machine for interacting with the mainframe
- "Desktop" style client that allows graphical access to mainframe data
- DevOps and related CI/CD tooling
- VS Code integration for developers

OPEN MAINFRAME PROJECT Zowe

Your open source software on the mainframe



Packages in "golang-hello-world" : golang-hello-world : Elizabeth K. Joseph - Google Chrome

https://launchpad.net/~lyz/+archive/ubuntu/golang-hello-world/+packages

1 → 2 of 2 results   First • Previous • Next ▸ • Last

Source	Published	Status	Series	Section	Build Status
golang-hello-world - 0.0~git20190613.ec269b1-1ppa6 (changes file)	2019-06-13	Published	Bionic	Games	✓

Publishing details

Published on 2019-06-13

Changelog

```
golang-hello-world (0.0~git20190613.ec269b1-1ppa6) bionic; urgency=medium

  * Update release version to latest LTS

 -- lyz@princessleia.com (Elizabeth K. Joseph)  Thu, 13 Jun 2019 14:05:50 -0700
```

Builds

- amd64
- i386
- s390x

Built packages

golang-hello-world Simple "Hello World" program in Go

Package files

- golang-hello-world_0.0~git20190613.ec269b1-1ppa6.debian.tar.xz (1.6 KiB)
- golang-hello-world_0.0~git20190613.ec269b1-1ppa6.dsc (2.0 KiB)
- golang-hello-world_0.0~git20190613.ec269b1-1ppa6_amd64.deb (366.2 KiB)
- golang-hello-world_0.0~git20190613.ec269b1-1ppa6_i386.deb (353.1 KiB)
- golang-hello-world_0.0~git20190613.ec269b1-1ppa6_s390x.deb (345.1 KiB)
- golang-hello-world_0.0~git20190613.ec269b1.orig.tar.xz (4.0 KiB)

Source	Published	Status	Series	Section	Build Status
golang-hello-world - 0.0~git20190613.ec269b1-1ppa3 (changes file)	2019-06-13	Published	Xenial	Games	✓

1 → 2 of 2 results   First • Previous • Next ▸ • Last

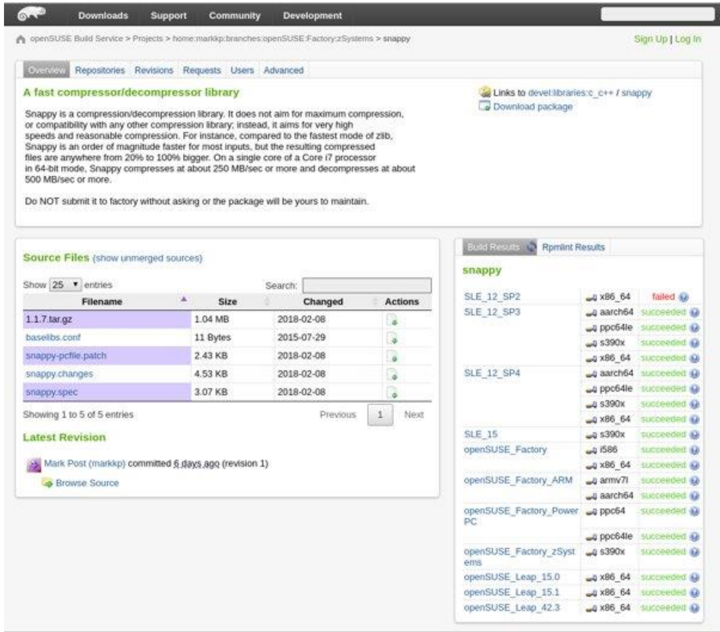
Free Tooling from the distributions

ubuntu

Ubuntu Personal Package archives on Launchpad.net https://help.launchpad.net/Packaging/PPA

SUSE

OpenSUSE build service https://build.opensuse.org/



openSUSE Build Service > Projects > home:markkp:branches:openSUSE:Factory:zSystems > snappy

A fast compressor/decompressor library

Snappy is a compression/decompression library. It does not aim for maximum compression, or compatibility with any other compression library; instead, it aims for very high speeds and reasonable compression. For instance, compared to the fastest mode of zlib, Snappy is an order of magnitude faster for most inputs, but the resulting compressed files are anywhere from 20% to 100% bigger. On a single core of a Core i7 processor in 64-bit mode, Snappy compresses at about 250 MB/sec or more and decompresses at about 500 MB/sec or more.

Do NOT submit it to factory without asking or the package will be yours to maintain.

Links to devel:libraries:c_c++ / snappy

Download package

Source Files (show unmerged sources)

Show 25 entries   Search:

Filename	Size	Changed	Actions
1.1.7.tar.gz	1.04 MB	2018-02-08	
baselibs.conf	11 Bytes	2015-07-29	
snappy-pcfile.patch	2.43 KB	2018-02-08	
snappy.changes	4.53 KB	2018-02-08	
snappy.spec	3.07 KB	2018-02-08	

Showing 1 to 5 of 5 entries   Previous 1 Next

Latest Revision

Mark Post (markkp) committed 6 days ago (revision 1)

Browse Source

snappy

SLE_12_SP2	x86_64	failed
SLE_12_SP3	aarch64	succeeded
	ppc64le	succeeded
	s390x	succeeded
	x86_64	succeeded
SLE_12_SP4	aarch64	succeeded
	ppc64le	succeeded
	s390x	succeeded
	x86_64	succeeded
SLE_15	s390x	succeeded
openSUSE_Factory	i586	succeeded
	x86_64	succeeded
openSUSE_Factory_ARM	armv7l	succeeded
	aarch64	succeeded
openSUSE_Factory_PowerPC	ppc64	succeeded
	ppc64le	succeeded
openSUSE_Factory_zSystems	s390x	succeeded
openSUSE_Leap_15.0	x86_64	succeeded
openSUSE_Leap_15.1	x86_64	succeeded
openSUSE_Leap_42.3	x86_64	succeeded

Your open source software on the mainframe

Open Mainframe Project Supported Projects Virtual Machines, hosted by Marist College

https://www.openmainframeproject.org/projects/supported-projects

MARIST

SUSE

Red Hat Enterprise Linux

Your open source software on the mainframe

Free CI/CD Tooling

Jenkins instance for s390x maintained by the Oregon State University Open Source Lab (OSU OSL): s390x Development Hosting

TravisCI build service for s390x (Beta trial for open source projects): Building on Multiple CPU Architectures

But what language?

Source code across architectures will generally be identical, but it needs to be compiled (C, C++), or otherwise interpreted (Python, Node.js) for this architecture.

That means you need a compiler or an interpreter built for the mainframe.

Language tips

The mainframe architecture is big-endian, but most of the supported architectures today are little-endian.

Your code will probably build and run, give it a try!

If not, the higher level a language is, the more luck you'll have. Lower level languages like C do more hardware-specific operations, higher level languages like Node.js have much of that abstracted away.

Thank you!