\Orchestrating a brighter world NEC

Towards Dom0 Disaggregation

Simon Kuenzer & Felipe Huici [simon.kuenzer|felipe.huici]@neclab.eu

Systems and Machine Learning Group

NEC Laboratories Europe (Heidelberg)

Xen Summit 2019, Chicago

This work has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreements no. 675806 (“5G CITY”) and no. 825377 (“UNICORE”). This work reflects only the author’s views and the European Commission is not responsible for any use that may be made of the information it contains.

Overview

Goal: use Unikraft as a basis for disaggregating Dom0

- Xenstore stub domain
- Toolstack/ssh stub domain
- QEMU stub domain
- Netback stub domain?
- Blkback stub domain?
- etc…

First Step – Xenstore Stub Domain

- Background: Minios-based stub domain

- `stubdom/Makefile : xenstore`

  →`tools/xenstore/Makefile : xenstored.a`

  → builds `XENSTORED_OBJS`

```
XENSTORED_OBJS = xenstored_core.o xenstored_watch.o xenstored_domain.o
XENSTORED_OBJS += xenstored_transaction.o xenstored_control.o
XENSTORED_OBJS += xs_lib.o talloc.o utils.o tdb.o hashtable.o
XENSTORED_OBJS_$(CONFIG_MiniOS) = xenstored_minios.o
```

- Points to multiple Minios-specific files:

```
libs/foreignmemory/minios.c
libs/call/minios.c
libs/evtchn/minios.c
libs/gnttab/minios.c
libxc/xc_minios.c
```

Building a Unikraft + Xenstore Unikernel

Approach: drive the build with Unikraft (**Makefile.uk**)

```
################################################################
# App registration
################################################################
$(eval $(call addlib,appcxenstored))


################################################################
# Fetch and unzip sources
################################################################
APPCXENSTORED_VERSION=xen-365aabb
APPCXENSTORED_URL='https://xenbits.xen.org/gitweb/?p=xen.git;a=snapshot;h=365aabb6e5023cee4\
76adf81106729efd49c644f;sf=tgz'
APPCXENSTORED_PATCHDIR=$(APPCXENSTORED_BASE)/patches

$(eval $(call fetchas,appcxenstored,$(APPCXENSTORED_URL),$(APPCXENSTORED_VERSION).tgz))
$(eval $(call patch,appcxenstored,$(APPCXENSTORED_PATCHDIR),$(APPCXENSTORED_VERSION)))


################################################################
# Create auto-generated header files
################################################################
$(APPCXENSTORED_BUILD)/.prepared: $(APPCXENSTORED_BUILD)/.origin
        $(call verbose_cmd,CONFIGURE,appcxenstored: $@,\
        cp $(APPCXENSTORED_BASE)/support/tools/Makefile $(APPCXENSTORED_TOOLS)/ && \
        cd $(APPCXENSTORED_ROOT) && ./configure && \
        $(MAKE) -C $(APPCXENSTORED_ROOT) tools  && \
        $(TOUCH) $@)

UK_PREPARE += $(APPCXENSTORED_BUILD)/.prepared
```

Building a Unikraft + Xenstore Unikernel

Approach: drive the build with Unikraft (**Makefile.uk**)

```
################################################################
# Build from source
################################################################
# ./stubdom/xenstore/xenstored.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_XENSTORE)/xenstored_core.c
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_XENSTORE)/xenstored_watch.c
…
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_XENSTORE)/hashtable.c

# ./tools/libs/toolcore/libxentoolcore.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/toolcore/handlereg.c

# ./tools/libs/toollog/libxentoollog.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/toollog/xtl_core.c
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/toollog/xtl_logger_stdio.c

# ./tools/libs/evtchn/libxenevtchn.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/evtchn/core.c|evtchn

# ./tools/libs/gnttab/libxengnttab.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/gnttab/gnttab_core.c
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/gnttab/gntshr_core.c

# ./tools/libs/call/libxencall.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/call/buffer.c
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/call/core.c|call
```

Building a Unikraft + Xenstore Unikernel

- Approach: drive the build with Unikraft (**`Makefile.uk`**)

```
################################################################
# Build from source
################################################################
# ./tools/libs/foreignmemory/libxenforeignmemory.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/foreignmemory/core.c|foreignmemory

# ./tools/libs/devicemodel/libxendevicemodel.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libs/devicemodel/core.c|devicemodel

# ./tools/libxc/libxenctrl.a
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libxc/xc_altp2m.c
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libxc/xc_core.c
...
APPCXENSTORED_SRCS-y += $(APPCXENSTORED_TOOLS)/libxc/xc_devicemodel_compat.c
```

Status – Compiles but Undefined References

Need to provide equivalents for `minios.c` files

```
OS functions
=====================================
mlock - (stub, return 0 in minios)
munlock - (stub, return 0 in minios)

tools/xenstore/xenstored_minios.c
=====================================
init_pipe
deamonize (stub)
finish_daemonize (stub)
unmap_xenbus
write_pidfile
xenbus_evtchn
xenbus_map
xenbus_notify_running

tools/libs/call/minios.c
=====================================
osdep_alloc_pages
osdep_hypercall
osdep_xencall_close
osdep_xencall_open
xencall_buffers_never_fault
osdep_free_pages
```

```
tools/libs/evtchn/minios.c
=====================================
osdep_evtchn_close
osdep_evtchn_open
osdep_evtchn_restrict
xenevtchn_bind_interdomain
xenevtchn_bind_unbound_port
xenevtchn_bind_virq
xenevtchn_fd
xenevtchn_notify
xenevtchn_pending
xenevtchn_unbind
xenevtchn_unmask

tools/libs/gnttab/linux.c
(no minios-specific file...)
=====================================
osdep_gntshr_close
osdep_gntshr_open
osdep_gntshr_share_pages
osdep_gntshr_unshare
```

Status – Compiles but Undefined References

```
tools/libs/gnttab/minios.c
=====================================
osdep_gnttab_close
osdep_gnttab_dmabuf_exp_from_refs
osdep_gnttab_dmabuf_exp_wait_released
osdep_gnttab_dmabuf_imp_release
osdep_gnttab_dmabuf_imp_to_refs
osdep_gnttab_grant_copy
osdep_gnttab_grant_map
osdep_gnttab_open
osdep_gnttab_set_max_grants
osdep_gnttab_unmap
```

```
tools/libs/devicemodel/linux.c
=====================================
osdep_xendevicemodel_close
osdep_xendevicemodel_op
osdep_xendevicemodel_open
osdep_xendevicemodel_restrict
```

```
tools/libs/foreignmemory/minios.c
=====================================
osdep_xenforeignmemory_close
osdep_xenforeignmemory_map
osdep_xenforeignmemory_open
osdep_xenforeignmemory_unmap
```

What‵s Missing

- Need to map undefined references to Unikraft functionality

- Almost all there, except:
  - `mmap`
  - gnttab support: add a new abstraction, a file descriptor type for dealing with grant maps

Is this the right approach?
Is this as minimal as we can build it?

\Orchestrating a brighter world

NEC