

OpenSolaris for System z

David Boyes
Neale Ferguson

Agenda

- **Why do this?**
- **Design Decisions**
- **Porting Process**
- **Progress Made**
- **Remaining Work (lots!)**
- **Where to now?**

Why?

- **Apart from the “cool hack” factor...**
 - **What's in it for IBM?**
 - **What's in it for Sun?**
 - **What's in it for users?**

What's in it for IBM

- **New System z workload**
 - **With the success of the Linux initiative, “mainframe” is less of a dirty word**
 - **System z capacity increasing to level some previous argument about CPU-intensive workloads**
 - **Opens up new avenues for “Solaris shops” to push effective virtualization**

What's in it for IBM?

- **Demonstration of z/VM being “best of breed”**
 - Sun domains para-virtualization strategy not working out to be particularly scalable
 - Recent cost structure changes to z/VM pricing leverage better per-virtual machine ROI
 - “Just one more” comment

“Why not? It’s just another virtual machine. We welcome any workload into the System Z family, we’re not picky.”

What's in it for IBM?

- **Continue the Server Consolidation push**
 - Makes Solaris workloads accessible for consolidation
 - Targets human workload as well as computational workload for better ROI

What's in it for Sun

- **Re-enfranchise StorageTek customers**
 - Counteract mixed message to STK customers wrt continued zSeries I/O and device development
- **“Stop loss” for existing enterprise customers in non-NIC fields**
 - Sun has steadily lost ground in large enterprise deployments over the last 4 years
- **Foot in the door to new customers**
 - Allows choice by superior manageability and operations concerns, not hardware platform

What's in it for Users

- **Competition**
 - Forces both IBM and Sun to concentrate on technical merit, not just price

What's in it for Users?

- **Integrated consolidation strategy**
 - **Permits concentration to fewer platforms and management tooling**
 - **Simplified D/R**
 - **Reuse of:**
 - **Skill set**
 - **Procedures**

What's in it for Users?

- **Elimination of “religious” arguments:**
 - **Anti-Linux**
 - **Anti-Sun**
 - **Anti-Open Source**

What's in it for Users?

- **New tools for improved productivity**
 - Availability of new application suites
 - Availability of desirable technology advances
 - Dtrace
 - System management enhancements
 - Printing system enhancements

Code Base

- **Current drop based on build 49**
- **Using the “mercurial” tool to keep current**

Design Decisions...

- **Codename “Sirius”**
- **_LP64 datamodel**
 - No 32-bit compatibility layer
- **Architecture Level Set**
 - Fullword immediate instructions
 - Compare-swap-and-purge (CSP/CSPG) instruction
 - Long displacement (RY) instructions
 - Long relative displacement instructions
 - Load Page Table Entry instruction (LPTE)
 - Purge DAT instruction
 - Cryptographic instructions

...Design Decisions...

- **ABI is identical to Linux for System z**
- **Assumes presence of z/VM**
 - 5.2 base
 - **DIAG interfaces:**
 - Block I/O
 - PFAULT
 - I/O discovery
 - Memory discovery
 - **Co-operative Memory Management**

...Design Decisions

- **I/O Layer similar to Linux CCW layer**
- **Separate address spaces for kernel and user processes**
 - Split code and data in separate address spaces to prevent buffer overwrite attacks
- **Full 64-bit (16EB) address space**
 - 3 levels of region table
 - Linux is 42 bit

Overall Porting Process

- **Build cross-compilation environment**
- **Write some glue code to allow proper module construction.**
- **Build library support and process entry/exit code**
- **Build I/O model and device drivers**
- **Make;make install...**

Current Build Environment

- **Cross-build environment on Sparc64**
 - Sparc is “big endian”
 - “ON Build” tools: part of OpenSolaris
 - Ported a couple of tools for s390x support
 - Normal Solaris tools: make
 - GNU tools with new target of “ibm-s390x-solaris2”
 - GCC 4.1.1
 - Binutils 2.17.50

Progress Made So Far

- **Configured V490 server provided by Sun infrastructure services**
- **Using mercurial to get updates from repository head**
- **Built cross-development tool chain on V490**
- **Updated makefiles for s390x architecture id and use of GCC**
- **Built onbid cross-build tools**

Progress Made So Far

- **Hardware initialization**
 - Control register loading
 - Memory detection (including non-contiguous)
 - I/O device detection
 - Interrupt handling (FLIH)
 - External interrupt processing (SLIH)
 - HMC driver
 - DAT-enabled kernel
- **Kernel-based Run-time Loader**
 - Used to load kernel and device drivers

Progress Made So Far

- **Completed clean build of kernel, usr/lib, and most commands**

Gratuitous Screen Shots...

```
initialize scratch memory
Installed physical memory:
(0x00, 0x08000000)
Booter occupied memory (including modules):
(0x0c00000, 0x0380000)
Ramdisk memory:
(0x0400000, 0x0800000)
Available physical memory:
(0x00, 0x0400000)(0xf80000, 0x07080000)
Available virtual memory:
(0x00, 0x0c00000)(0xf80000, 0xff080000)
DAT Enabled using RTO f82000
Relocating the UNIX executable
Loaded section at 0x50000 for 0xc3418 bytes
Loaded section at 0x114000 for 0x4a1a0 bytes
Section is located at 0xd00000
Section .interp relocated from 0xd00190 to 0x50190 (11 bytes)
:
Relocating the KRTLD executable
_edata set to 015e1a8 at 015e1a0
Section is located at 0xf00000
Section .text relocated from 0xf00190 to 0x3400 (49096 bytes)
:
krtld: file=/platform/s390x/unix
text: 0x50000 size: 0x0
data: 0x114000 dsize: 0x0
krtld: file=misc/krtld
text:0x3400 size: 0x110018
data:0x12000 dsize: 0x14fd9a
loading module genunix (genunix)
```

...Gratuitous Screen Shots...

```
OpenSolaris on System z - Startup commenced
Memory size: 128MB Chunks: 1
0. 0000000000000000 08000000 0
Discovering CPUs
Boot CPU logical address: 0 hardware address: 0
1 CPUs detected
Initializing timers
Initializing memory
.../s390x/os/startup.c:710: 'sysSize' is 0x8000000
.../s390x/os/startup.c:711: 'physmax' is 0x7fff
.../s390x/os/startup.c:720: 'availmem' is 0x7fffffff
.../s390x/os/startup.c:722: 'nalloc_sz' is 0x7fbfaf5
.../s390x/os/startup.c:732: 'moddata' is 0x50a
:
Initializing I/O structures
Console address = 0009
Highest subchannel address encountered: 001f
I/O Device List starts at 106b000 for 5536 bytes
:
```

...Gratuitous Screen Shots...

CP Q V STOR
STORAGE = 128M

PSW - DAT On, Home space mode

PSW = 0400E001 80000000 00000000 000415DE

Virtual Address Space

```

CP D 8000000.
V0000000008000000 to 000000007FFFFFFF non-addressable storage - segment
translation exception
V0000000008000000 to 000002FFFFFF non-addressable storage -
Region-Third-Translation exception
V0000030000000000 to 00000300000000FF non-addressable storage - page
translation exception
V0000030000000100 00000000 00000000 00000000 04 R00790000
V00000300000001010 to 00000300000001FFF suppressed line(s) same as
above ....
V00000300000002000 to 0000030000000FFFF non-addressable storage - page
translation exception
V00000300000100000 to 000003007FFFFFFF non-addressable storage - segment
translation exception
V00000300800000000 to 000003FFFFFF non-addressable storage -
Region-Third-Translation exception
V00000400000000000 to 001FFFFFFF non-addressable storage -
Region-Second-Translation exception
V00200000000000000 to FFFFFFFF non-addressable storage -
Region-First-Translation exception

```

...Gratuitous Screen Shots...

Home Space Control Register

CRG 13 = 000000000064000F

Region First Level Table

```

V00640000 00000000 0064400F 00000000 00000020 06 R00640000
V00640010 00000000 00000020 00000000 00000020
V00640020 to 00643FFF suppressed line(s) same as above ....

```

Region Second Level Table

```

V00644000 00000000 0064800B 00000000 00000020 06 R00644000
V00644010 00000000 00000020 00000000 00000020
V00644020 to 00647FFF suppressed line(s) same as above ....

```

Region Third Level Table

```

V00648000 00000000 0064C007 00000000 00000020 06 R00648000
V00648010 00000000 00000020 00000000 00000020
V00648020 to 0064AFFF suppressed line(s) same as above ....
V0064B000 00000000 00791007 00000000 00000020 06 R0064B000
V0064B010 00000000 00000020 00000000 00000020
V0064B020 to 0064BFFF suppressed line(s) same as above ....

```

...Gratuitous Screen Shots

Segment Table for Region 0

```
V0064C000 00000000 00650000 00000000 00650800 06 R0064C000
V0064C010 00000000 00651000 00000000 00651800
```

Segment Table for Region 0x600

```
V00791000 00000000 00795000 00000000 00000020 06 R00791000
V00791010 00000000 00000020 00000000 00000020
```

Page Table for Region 0x0 Segment 0

```
V00650000 00000000 00000000 00000000 00001000 06 R00650000
V00650010 00000000 00002000 00000000 00003000
```

Page Table for Region 0x0600 Segment 0

```
V00795000 00000000 00000400 00000000 00790000 06 R00795000
V00795010 00000000 00000400 00000000 00000400
```

Mapping of virtual address 0x3000001000

```
V0000030000001000 00000000 04 R00790000
```

Remaining Development Areas

- **Non-kernel virtual memory support**
- **I/O support (DDI)**
 - Interface similar to Linux CCW device layer
 - Device drivers
- **Machine check handling/error management**
- **External interrupt handling**
- **dtrace and mdb**
- **Port of Solaris linker to s390x**
 - Link process uses several options not supported with GCC

Where to now?

- **Support from Sun**
 - Meeting with developers in Menlo Park
 - Conference call with CTO
 - Using the OpenSolaris.org community website
- **Get infrastructure in place**
- **Create “Sirius” community on OpenSolaris.org**
- **Open for public participation in port**