

# Porting TRU64 UNIX™ on Avanti™ Emulator

Mohamed Saied Essa Othman Hussien  
Level2 Automation Flat Steel Department  
EZDK for iron and steel Co.  
Alexandria, Egypt  
Email: Alexengineer2009@gmail.com

**Abstract**—Real time processing is becoming an increasingly important discipline. It is one of the mandatory requirements in industrial applications in order to achieve the five general areas: determinism, responsiveness, user control, reliability and fail soft operation; but what is the solution if you have aging hardware like my company? You will meet big problems and huge workload to maintain and develop your system and its applications. How can you keep your legacies alive? You need an alternative approach. Emulation will allow you to use new hardware without having to change the application, improve up-time and reliability, save floor space, power and maintenance cost, make the application faster, and preserve the past investment in the application. In this paper I perform a case in my company which was TRU64 UNIX™ O/S installed on DEC Alpha servers™. This paper will explain the way of porting it into two virtual emulators Avanti™ and FreeAXP™ and the problems that faced us during porting.

**Keywords**—Emulator, Virtual machine, RTOS, TRU64 UNIX, Avanti, Charon

## I. INTRODUCTION

Real-time operating systems (RTOS) have been a target of virtualization in recent years. There were many trials to implement virtual machines that support UNIX™ and Open VMS™ O/Ss, but no one could; till suddenly the dream became a real fact. It is now possible to make Intel™ architecture appear identical to legacy systems.

In the Iron and Steel industry, Level 2 of Automation is responsible for Treatment practices, Process models and Process tracking, so it requires RTOS to handle rapid communication with other systems and provide high response and stability, In our flat steel plant (FSP) we use Open-VMS™ O/S and in SMP (steel making plant) we use TRU64 UNIX™; running on obsolete DEC Alpha™ servers, Since we could no longer source spare parts for the Alpha systems, turned to virtualization as the best replacement solution. We will explain porting TRU64 UNIX™ on Avanti™ emulator.

TRU64 UNIX™ is a 64-bit UNIX operating system for the Alpha instruction set architecture (ISA), currently owned by Hewlett-Packard (HP™). Previously, Tru64 UNIX™ was a product of Compaq, and before that, Digital Equipment Corporation (DEC™), where it was known as Digital UNIX™. After Compaq's purchase of DEC in early 1998, with the release of version 4.0F, Digital UNIX was renamed to Tru64 UNIX to emphasizing its 64-bit-clean nature and de-emphasizing the Digital brand.

Avanti is a commercial virtual Alpha solution designed to support demanding Alpha replacements. Avanti is suitable to replace low end Alpha hardware. It is a virtual machine that can emulate the legacy Digital (DEC), Compaq and HP Alpha hardware. It is binary compatible with OpenVMS™ and TRU64 UNIX™ operating systems.

## II. EMULATOR CHOICES

With emulation you can get many advantages like that the original licenses can be transferred to an emulated system for a small fee, binary compatibility (source code not needed), and no changes required to system.

If you decide to emulate your server, there are many alternatives which you can use as a virtual emulator:

### 1) ES40

It is an open source developed by Camiel Vanderhoeven and Written entirely in C++; but it is slow, buggy, terribly complex, dependent on hobbyist efforts and no model for generating revenue, its development has virtually been halted since 2009.

### 2) SIMH

It is an open source VAX and PDP 11 emulator. It is free of charge. You can get some support, but there is no direct access to hardware and it not certified by HP as being a true emulation of a VAX.

### 3) CHARON VAX™

It is a commercial-grade VAX emulator, Certified by HP as being a true emulation of a VAX. Its Support is available through VARs and Stomasys. Products range from 128 MB single-processor systems to 3GB, 6-processor SMP systems. You can access physical disk and tape drives, including SAN storage. It can be hosted on Windows or Integrity OpenVMS.

### 4) NUVAX™

It is a commercial grade VAX emulator, certified by HP as being a true emulation of a VAX. Its Support is available through VARs and the logical company. It Comes with dedicated server and optional Qbus card cage, Hosted on real time OS (QNX).

### 5) FREE ALPHA EMULATORS

These emulators are a way to give back to the VMS community. They also provide a pool of volunteer beta testers.

They offer limited memory, disks, and performance. For example:

- Personal Alpha (Stromasys): It is a limited version of CHARON-AXP and not for commercial use.
- FreeAXP (Migration Specialties): It can be used for low-performance commercial applications. It is free but not an open-source.

Both run on 32-bit Windows

### 6) CHARON AXP™

It is a commercial grade Alpha emulator, it passes all Alpha architecture validation tests and its support is available through VARs and Stromasys. The Products range from Alpha Station 200 to 32-processor Alpha Server GS320. You can access physical disk and tape drives. It requires 64 bit Windows host.

### 7) AVANTI™

It is a commercial grade Alpha emulator. Its support is available through VARs and Migration Specialties. It requires 64 bit Windows host.

You can choose any emulator from the above alternatives; but we will use Avanti™ emulator in this paper.

## III. AVANTI™ EMULATOR ARCHITECTURE

A legacy hardware emulator like Avanti is a self-contained program that provides all of the interfaces a legacy O/S and its applications expect. As such, an emulator can fully replace physical hardware. Avanti™ runs as a program on a Windows system. It presents a hardware interface to OpenVMS™ and Tru64 UNIX. The emulated hardware interface provides access to both virtual and real hardware such as disk drives, CD drives, NICs, and serial ports. The OpenVMS™ and Tru64 UNIX™ systems running on Avanti™ are completely unaware they are being hosted on a Windows system. From the OpenVMS™ or Tru64 UNIX™ perspective, they are running on real Alpha hardware, see fig1.

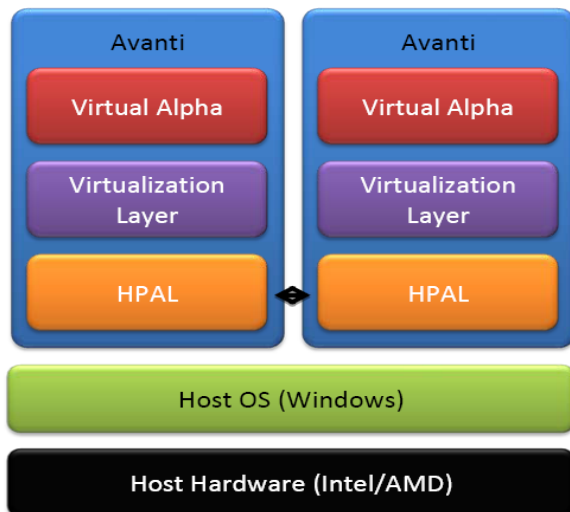


Figure1 Overall Avanti™ emulator architecture

## IV. PORTING PROCEDURES

Porting is not a hard operation; it does not require an expert in UNIX O/S. As you will see, it consists of certain procedures. By following those procedures you will port your machine successfully. Ready? Let's start:

1. First of all, you must make sure that the hardware of the host server must meet with the minimum requirements or higher which was determined by Migration Specialties Company as mentioned in table 1. But you should take into your consideration that the performance of Virtual Alpha is highly dependent upon how the host server is configured. A slow host will adversely impact emulator performance. Network and disk I/O present similar issues. Both are typically much faster than legacy Alpha hardware by virtue of the enhanced speed of modern hardware. However, final performance values are going to be dictated by configuration, network load, I/O load, and CPU speed. When I asked Mr. Bruce Claremont (Current. CEO at Migration Specialties) about that he said "I can say that host CPU speed has a direct correlation to virtual Alpha CPU speed. Avanti running on a 3GHz CPU will be 100% faster than Avanti running on a 2GHz CPU" which means that higher host specifications, higher performance in emulator.

Component	Minimum Requirement
O/S	Windows 64bit (XP, Vista, 7, Server 2003, Server 2008)
CPU	Dual core
RAM	2 GB for Windows XP and Windows Server 2003 4 GB for Windows 7, Vista, Windows Server 2008
NIC	2 NICs

Table1 Minimum requirements of Host O/S

2. Install the Avanti™ or FreeAXP™ on your system, and Configure Avanti to match the legacy Alpha hardware such as (Hard disks, CD ROM, RAM, SCSI, Port number, JIT...etc.), you can use the following command to display all the devices on your server:

```
# sys_check -all > sys.html
```

Avanti™ emulator provides a Variety of virtual hardware as mentioned in table 2. In case of Tru64 UNIX™ installations, they are currently limited to 1GB of memory.

Component	Specification
CPU	1 EV4
Memory	32MB - 1.5GB
Disk Controllers	1 - 3
Disks per Controller	7 (KZPAA) or 15 (KZPCA)
NICs	1 - 3
Maximum Disk Size	1 TB (limit determined by operating system)
Serial Ports	2 (virtual console + one virtual/physical)
CD-ROM	45
PCI slots	4

Table2 Avanti™ virtual hardware summary

You must choose the network interface device according to the version of your operating system as mentioned in table 3. Then, save this configuration into a .CFG file. Then run the machine.

NIC	Recommended Usage
DE435	OpenVMS™ version 6.1-1H1 and higher. Tru64 UNIX™ version 3.2C and higher.
DE500 (21140)	Tru64 UNIX™ version 4.0 up to 4.0E.
DE500 (21143)	OpenVMS™ version 7.2 and higher. Tru64 UNIX™ version 4.0F and higher.

Table3 Avanti™ virtual NIC

Avanti™ emulator emulates up to 3 network interface cards (NICs) by making a virtualization layer for the network, you can see fig2 which shows that the host system is unaware that it is supporting virtual legacy hardware. Windows just sees a program that uses files and physical devices much like any other program. The consequence of this is that for the host system to communicate with the legacy Alpha, it must use standard network protocols such as TCP/IP. Hence the requirement is that the host system provides two NICs, one for itself and one for host virtual Alpha communications. In order to avoid interference between host network traffic and the guest network traffic, it is recommended to use a dedicated NIC for the guest rather than sharing the host NIC and uninstall all network protocols on guest NIC from the host operating system.

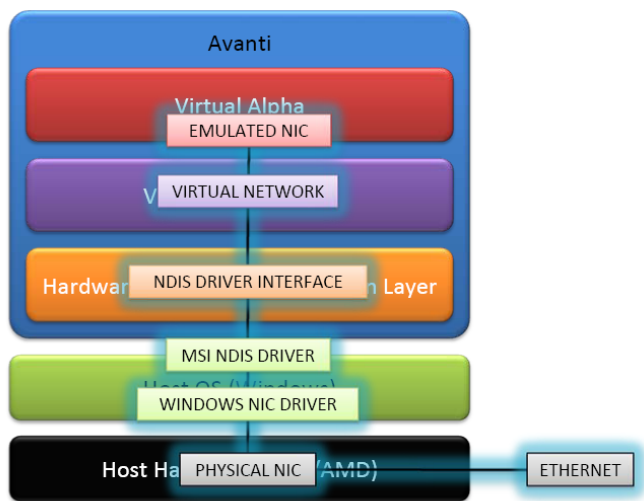


Figure2 Avanti™ emulator network architecture

Avanti™ supports up to three KZPAA (narrow) and KZPCA (wide) SCSI controllers. Which controller is available for use is dependent upon the O/S version installed.

3. Install any version of UNIX™ O/S as you want (in my trials, the version was 4.0F, for supported Tru64/Digital UNIX™ releases see table 4).

Release	Supported from	Release	Supported from
V3.2C	Migration Specialties	V5.0A	Migration Specialties
V4.0B	Migration Specialties	V5.1	Migration Specialties

V3.2G	Migration Specialties	V5.1A	Migration Specialties
V4.0D	Migration Specialties	V5.1B	Migration Specialties
V4.0E	Migration Specialties	V5.1B-5	HP
V4.0F	Migration Specialties	V5.1B-6	HP
V4.0G	Migration Specialties		

Table4 Supported Tru64/Digital UNIX™ releases

4. Backup the disks of your legacy Alpha™ server to the Avanti™ system.
5. Restore the legacy system disks on the Avanti system.
6. Now, if you try to use any device you will find that all devices are not working correctly. For example, if you try to connect to a computer on the network using the PING command or try to execute any network commands, it will not run. You can check the configuration of the network interface card (NIC) using *ifconfig -a* command and the */etc/rc.config*,

So you must allow UNIX™ O/S to support the devices. To do that you must execute the following procedures:

- A) Install those drivers  
(This step In case of new installation not in case of restoring the backup of the legacy server)
- B) Edit target configuration file
- C) Build a new kernel  
(It is a must after installing the drivers to build a new kernel, because any changes happened on the hardware will affect on the kernel as shown in fig.3)
- D) Reboot the system

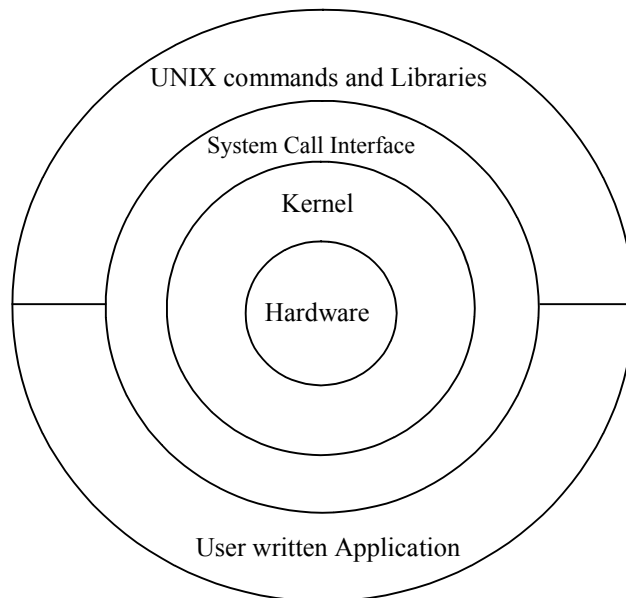


Figure3. General UNIX architecture

#### V. DETERMINATION OF DEFINITION KEYWORD

TRU64 UNIX™ knows which platform it is running on, and has a */vmunix* built specifically for each type of Alpha. It supports many devices and each device has its keyword, so we must determine which device definition keyword we should add.

IF you have the system documentation, there will be no problem because you will find the appropriate definition keyword by searching in it; but if you are like us and you did not have the documentation what could you do?

Do not give up; there is another way to do it. If you have another source which has the same connected devices and has an existing configuration file

1. Log in as root or any super user and set your default directory to the following directory

```
/usr/sys/conf
```

Note: for paths of directories which will be needed, see fig.4 which shows the configuration files directories hierarchy in bolded text with blue lines.

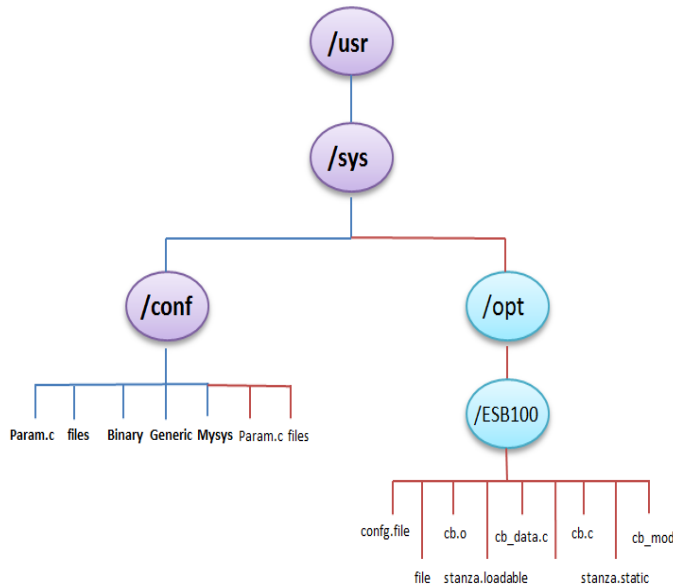


Figure4. Configuration Files Directory Hierarchy

2. Save a copy of the existing /vmunix file. If possible, save the file in the root (/) directory, as follows:

```
# cp /vmunix /vmunix.save
```

3. Shutdown and halt the system as follows:

```
# shutdown -h now
```

4. At the console prompt, boot the generic kernel, /genvmunix. The generic kernel contains support for all valid devices, so if you boot it while adding the devices to your target kernel, the devices are known to the kernel. To boot the generic kernel, issue the following command:

```
>>> boot -fi "genvmunix"
```

5. If the /genvmunix file does not exist on your system, or the generic kernel fails to recognize the device, rebuild the generic kernel.

After all kernel subsets are installed, issue the following command:

```
# doconfig -c GENERIC
```

At the single-user mode prompt, check and mount local file systems by issuing the following command, unless you are using the Logical Storage Manager software (LSM):

```
# /sbin/bcheckrc
```

If you are using the Logical Storage Manager (LSM) software, check local file systems and start LSM by issuing the following command:

```
# /sbin/lsmstartup
```

6. Run the sizer program to size your system hardware and create a new target configuration file that includes the devices:

```
# sizer -n MYSYS
```

The sizer -n command creates a new target configuration file for your system that includes the appropriate device definition keyword for the devices. The sizer program stores the new target configuration file in the /tmp directory.

7. Compare the new target configuration file created by sizer with the existing target configuration file for your system:

```
# diff /tmp/MYSYS MYSYS
```

Check the differences between these files until you find the new device definition keyword.

By searching in our file we found the definition keyword as follows:

```
"controller tu0 at pci0 slot 11"
```

So we changed the controller name and the slot number in the new configuration file.

## VI. CONFIGURATION FILE ENTRIES

The system configuration file contains the following keyword definitions:

- Global keyword definitions
- System definition keywords
- Device definition keywords
- Callout keyword definitions
- Options keyword definitions
- Make options keyword definitions
- Pseudo-device keyword definitions

We will mention only the common devices definition keywords in table 5.

keyword	Default	Configurable	Required
controller	All supported controller types	Yes	Yes
Bus	All supported bus types	Yes	Yes
device disk	All supported disk device types	Yes	Yes
device tape	All supported tape device types	Yes	Yes

Table5 Common Keyword Definitions

## VII. EDITING OF CONFIGURATION FILE

If the driver provides a subset control program (SCP), there is no need to edit the configuration file manually. But in case of no SCP, we must edit the target configuration file, By using any text editor to add your devices definition keywords to your existing Configuration file (in this case, MYSYS and we used the VI editor). Adding the new keyword allows your existing configuration file to support your devices without losing any changes you made to that file in the past

## VIII. VI EDITOR

To edit the configuration file you will need a text editor like VI editor. VI editor is a full-screen text editor. . The vi editor is a modal editor. In command mode, it accepts commands for cursor movement, text deletion, and so forth. To insert text into the file, the user gives the editor a command that places the editor in input mode, and all keystrokes thereafter are interpreted as input data until the Escape key is pressed. You will use the following commands during editing of the target configuration file as mentioned in table 6.

Command	Function
vi	To edit a certain file:
h	left one space
l	right one space
j	down one space
k	up one space
w	moves the cursor forward one word
b	moves the cursor backward one word
e	moves to the end of a word.
x	deletes the character under the cursor.
X	deletes the character to the left of your cursor.
dw	deletes from the character selected to the end of the word.
dd	deletes all the current line.
D	deletes from the current character to the end of the line.
:w	to save your file but not quit vi (this is good to do periodically in case of machine crash!).
:q	to quit if you haven't made any edits.
:wq	to quit and save edits (basically the same as ZZ).

Table 6 VI Editor impotent commands

## IX. BUILDING NEW KERNEL

Build a new kernel by issuing the following /usr/sbin/doconfig command:

```
# /usr/sbin/doconfig -c MYSYS
```

Answer the following prompt to indicate that you do not want to edit the configuraton file:

```
Do you want to edit the configuration file? (y/n) [n]: n
```

The new kernel is /usr/sys/MYSYS/vmunix

When the kernel configuration completed and the building process finished without errors, move the new vmunix file to /vmunix. On a system named MYSYS, issue the following command:

```
# mv /usr/sys/MYSYS/vmunix /vmunix
```

Then, reboot the system as follows:

```
# /usr/sbin/shutdown -r now
```

## X. FAILURE OF NEW /VMUNIX FILE

In case of the new /vmunix file fails to boot, you can boot using the kernel that you saved at the beginning of the procedures. To use the saved kernel, follow these steps:

1. Check all local file systems as follows:

```
# fsck -p
```

2. Write enable the root file system as follows:

```
# mount -u /
```

3. If necessary, mount the file system where the /vmunix.save file is stored. For example, if you copied the /vmunix file to the /usr file system, issue the following command:

```
# mount /usr
```

4. Restore the saved copy. For example, if you saved your running kernel in the /vmunix.save file, issue the following command:

```
# mv /vmunix.save /vmunix
```

5. Shutdown and reboot the system, as follows:

```
# shutdown -r now
```

After your system is running again, you can modify the target configuration file as needed and rebuild the kernel starting at step 3.

## XI. TESTING YOUR APPLICATIONS

On our legacy Alpha we were using the Oracle™ database engine for data warehousing and a PRODAT database as the run time database. We succeeded installing it and it is now working correctly with Avanti™. It does not work with FreeAXP™ because it is a limited emulator; it has only 128-MB of RAM. Also the clients of the old system have many applications which were working using the configuration X; we succeeded to install them and all applications now working correctly with the same performance of the real legacy servers.

## XII. FREEAXP-32 DISK CORRUPTION

A disk corruption may be happened under the 32-bit edition of FreeAXP 412, so keep your FreeAXP™ system backed up and if it is possible, move to FreeAXP-64 because Migration Specialties Company has ascertained that this problem does not exist in the 64-bit edition of FreeAXP.

## XIII. CHANGE FROM CHARON™ TO AVANTI™

We also tried to change the virtual machine from Charon-AXP™ to Avanti™ and we succeeded with no errors using the same procedures and using the same virtual hard disks files (.vhd).

## ACKNOWLEDGMENT

I would like to express my gratitude to my management in EZDK Co. for providing me with support.

## REFERENCES

- [1] William Stallings , Operating Systems internal and design principles ,5th ed J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [2] TRU64 UNIX – System Administration – Compaq Computer Corporation, Houston, Texas
- [3] Digital UNIX – Command and shell user's guide – Digital Equipment Corporation – Maynard, Massachusetts
- [4] TRU64 UNIX – Documentation Overview – Compaq Computer Corporation, Houston, Texas
- [5] Avanti™ SPD V1.0 - Digital Emulator Collection –Migration Specialties.