Energy Saving with WOL
Hands On Guide
peter-thoemmes.org research

© Peter Thoemmes
Weinbergstrasse 3a
D-54441 Ockfen, Germany

June 14, 2011

Abstract

This hands on guide is showing you how you can save energy when running a Linux machine 24 hours a day, say as NAS or Web server. This paper is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Contents

1 The Basic Idea

The basic idea behind the 'energy saver' is the WOL concept, which stands for Wake On LAN. This means that by any trigger a machine is put asleep and can then be woken up only via network sending a magic packet. When the network adapter receives a magic packet it makes the computer resuming its work. Yes, it is as simple as that. And it saves a whole lot of energy. Typically a machine running on an Intel® Core™ i7 CPU with 8 cores at 3.20 GHz, having 16 GB of RAM, two network adapters, two highly available power supplies and two mirrored hard disk drives of 2 TB size each, will consume at least **100 Watts** when idle. When suspended, waiting for a WOL magic packet it will only consume about just **5 Watts**, which is **95 % less**. Also beside saving 95 % of power during the time your machine is suspended, you can save energy, which is eaten by your cooling system (HVAC system). That is because in some server rooms a lot of servers do have potential idle time and can be suspended to wait for being waked up by WOL. When running on 5 Watts only those machines produce almost no heat. So if the responsible engineers implement an intelligent system to suspend machines running in idle mode and at the same time adjust the cooling system, a lot of energy saving is possible.

2 How to make a Linux machine sleeping

In principle a Linux machine can be put asleep by sending following command:

```
echo 3 > /proc/acpi/sleep
```

So you write a 3 into /proc/acpi/sleep. Most important thing to do first is to ensure that in the **BIOS** settings (Power Management) **Resume on LAN** is enabled. Otherwise it will never work!

To make things more sensitive, you should only suspend the machine when it is idle. Whether the machine is idle or not can be tested using the tool **sar**. To do so you have to install sar first, as shown here for ubuntu:

```
# apt-get install sysstat
```

Then you can check different things, like the average CPU usage in percent:

```
# sar -u 1 3 | grep Average | awk '{ printf "%d\n",100-$8}'
```

You can check the disk usage by reading the average number of blocks written or read per second:

```
# sar -b 1 3 | grep Average | awk '{ printf "%d\n",$5 + $6}'
```

To check the network activity read the number of transferred KB/s on the adapter, here shown for the adapater eth0:

```
# sar -n DEV 1 3 | grep eth0 | grep Average | awk '{ printf "%d\n",$5+$6}'
```

You can see the number of locked SAMBA files (typically locked by Microsoft® Windows™ file sharing) by executing following command:

```
# smbstatus --locks | grep -e '^[0-9]'
```

To see all the users logged on from a remote computer:

```
# last | grep 'still logged in'
```

All these checks are automatically done by the script **try-to-suspend.sh** written by Peter Thoemmes. You can get this script for free:

```
# wget https://downloads.sourceforge.net/project/energy-saver/linux-energy-saver.tar.gz
# tar zxvf linux-energy-saver.tar.gz
```

Follow the instructions to setup everything fine in the README file. When setup well you can use this script as follows. For example if you want to make the script trying for the next 5 minutes to get the machine suspended, type this:

```
# try-to-suspend.sh 5
```

To automatically try to get a machine suspended once every 30 minutes, simply run the script **suspender.sh** in the background like this:

```
# nohup suspender.sh >/dev/null
```

As shown in the setup instructions you may add this line to /etc/rc.local to automatically run the suspender each time starting up the server.

3 How to wake up a machine

The magic packet to wake up a machine is a data link layer frame (Ethernet frame) and so uses the 48 bit MAC address for addressing the recipient. The payload of the frame must contain a UDPv4 packet, meaning an IPv4 header followed by a UDPv4 header followed by the actual payload, let's call it WOL payload. This UDPv4 packet is typically send to the IPv4 broadcast address 255.255.255.255. The WOL payload must contain a sequence of 6 bytes 0xFF (starting delimiter) followed by 16 times the destination MAC address:

```
... FF FF FF FF FF FF aa aa aa aa aa aa ... aa aa aa aa aa aa ...
    |<- Start Del.->| |<- MAC addr -->| ... |<- MAC addr -->|
                      |<------ 16 times MAC address ------->|
```

This means **102 bytes** in total. The packet can also be sent directly to the destination IPv4 address, which makes the process of waking up a machine a bit faster and more efficient. Port number 9 is a good choice for the destination port number.

A network card may support **SecureOn**, which allows to set a password, which is 6 bytes long. Those 6 bytes have to follow the normal magic packet, which means that in total the sequence is **108 bytes** long.

We can sent a normal password-free WOL frame using the **ethercat** tool developed by Peter Thoemmes. The command might look like this:

```
$ echo "FFFFFFFFFFFF8033..." | ethercat \
-i wlan0 -dst 80:33:CB:82:24:27 -udpv4 255.255.255.255:9 -hex_str_input
```

Simply sending a UDPv4 datagram to the IPv4 address (say to 192.168.2.33) will not do the job, as the recipient is currently sleeping. So following way to send the magic packet **will not work**:

```
$ dd if=./magic-pkt.bin > /dev/udp/192.168.2.33/9
0+1 records in
0+1 records out
102 bytes (102 B) copied, 4.4e-05 s, 2.3 MB/s
```

The script **wol.sh** coming with the ethercat package makes life very easy here. It makes use of **ethercat** and allows to wake up a machine by typing a command like this:

```
$ wol.sh wlan0 80:33:CB:82:24:27
```

The first argument is name of the local Ethernet adapter and the second argument is the MAC address of the server's Ethernet adapter. So simply use this command to wake up a machine.

If you run an operating system other then Linux, then use any WOL tool to send the magic packet. You may download free WOL tools from the Internet (e.g. http://www.gammadyne.com/cmdline.htm#wol) or check out if your operating system distribution supports WOL by default.