

нейронную сеть – значит, сообщить ей, чего мы от нее добиваемся: для этого у нас должна быть некоторая база входных данных и соответствующих им выходных значений. Алгоритм обратного распространения ошибки – это набор формул, который позволяет по вектору ошибки вычислить требуемые поправки для весов нейронной сети [2,3,4].

После многократного предъявления примеров веса нейронной сети стабилизируются, причем нейронная сеть дает правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что "нейронная сеть выучила все примеры", "нейронная сеть обучена", или "нейронная сеть натренирована". В программных реализациях можно видеть, что в процессе обучения величина ошибки (сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда величина ошибки достигает нуля или приемлемого малого уровня, тренировку останавливают, а полученную нейронную сеть считают натренированной и готовой к применению на новых данных.

Во многих случаях применение генетического алгоритма может помочь уменьшить число циклов прогонки для обучения нейронной сети путём использования более значимых данных для подстройки весов.

Генетические алгоритмы – адаптивные методы поиска, которые в последнее время часто используются для решения задач функциональной оптимизации. Вначале генерируется определенное количество возможных решений, а затем вычисляется для каждого "уровень выживаемости" – близость к истине. Эти решения дают потомство. Те, что "сильнее", т. е. больше подходят, имеют больший шанс к воспроизводству, а "слабые" постепенно отмирают. Процесс повторяется до тех пор, пока не будет получено оптимальное решение или не получено достаточное к нему приближение.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Остроух Е.Н., Чегодарь М.Ю.* Применение алгоритма генетического поиска при тестировании программного обеспечения/ Труды Южного (Ростовского) отделения Академии информатизации образования. – Ростов н/Д: РГПУ, 2006. – 176 с.
2. *Короткий С.* Нейронные сети: основные положения. <http://lii.newmail.ru>
3. Генетические алгоритмы <http://www.neuroproject.ru>
4. *Курейчик В.М.* Генетические алгоритмы и их применение. – Таганрог: Изд-во ТРТУ, изд. 2-е, доп, 2002, – 242с.

УДК 681.35

Б.А. Державец

УСТАНОВКА UBUNTU INTREPID SERVER PV DomU В СРЕДЕ Xen 3.3 Ubuntu Hardy Dom0

Известно, что в среде естественного Xen Hypervisor Ubuntu Hardy Dom0 эта попытка терпит крах (Xen 3.2). Выполним Back port Intrepid Xen 3.3 Hypervisor на Ubuntu Hardy Dom0.

В среде Synaptic Manager пометим для установки следующие пакеты : lib-xen3-dev xen-doc-3.3 xen-hypervisor-3.3 xen-utils-3.3 и подтвердим разрешение конфликтов перед установкой (рис. 1).

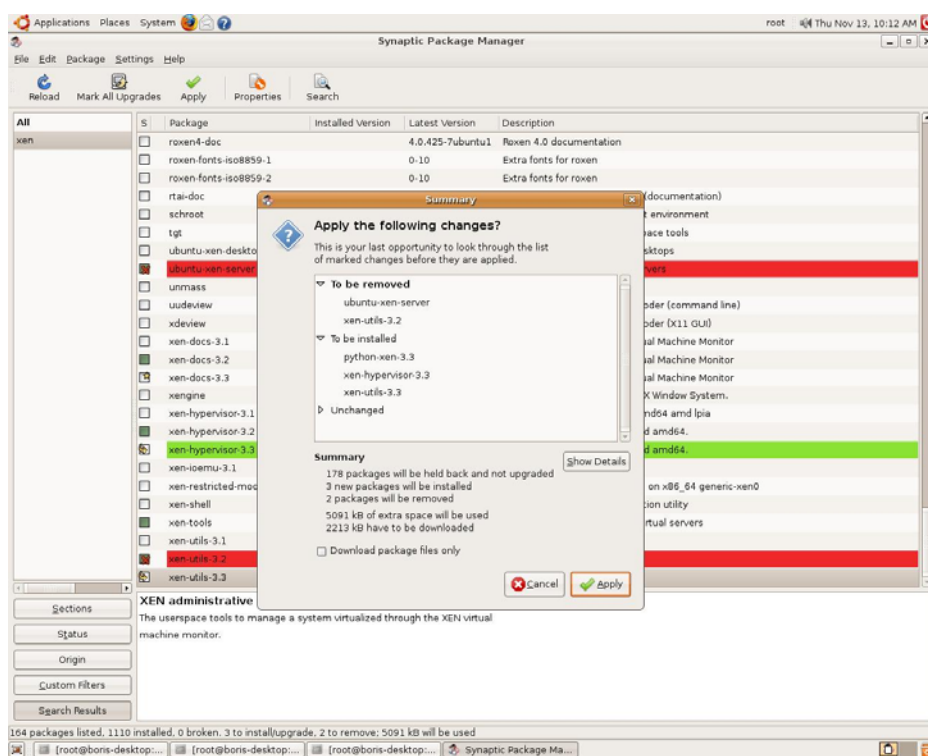


Рис. 1 Окно, показывающее установку пакетов `libxen3-dev xen-doc-3.3 xen-hypervisor-3.3 xen-utils-3.3`

```

Перезагрузим систему с вновь установленным Xen 3.3
root@boris-desktop:~# xm info
host          : boris-desktop
release      : 2.6.24-21-xen
version      : #1 SMP Wed Oct 22 01:07:57 UTC 2008
machine      : x86_64
nr_cpus      : 2
nr_nodes     : 1
cores_per_socket : 2
threads_per_core : 1
cpu_mhz      : 2400
hw_caps      :
bfefbfff:20100800:00000000:00000140:0000e3bd:00000000:00000001:00000000
virt_caps    : hvm
total_memory : 4095
free_memory  : 36
node_to_cpu  : node0:0-1
node_to_memory : node0:36
xen_major    : 3
xen_minor    : 3
xen_extra    : .0
xen_caps     : xen-3.0-x86_64 xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-
x86_32p hvm-3.0-x86_64
xen_scheduler : credit

```

```
xen_pagesize      : 4096
platform_params   : virt_start=0xffff800000000000
xen_changeset     : unavailable
cc_compiler       : gcc version 4.2.4 (Ubuntu 4.2.4-1ubuntu3)
cc_compile_by     : builddd
cc_compile_domain : builddd
cc_compile_date   : Wed Oct 29 23:39:07 UTC 2008
xend_config_format : 4
```

```
root@boris-desktop:~# brctl show
bridge name      bridge id          STP enabled  interfaces
eth1              8000.000c76e01ec5  no          peth1
vif5.0.
```

Теперь создадим HVM DomU, используя профайл

```
kernel = "/usr/lib/xen/boot/hvmloader"
builder = 'hvm'
memory = 2048
name = "IntrepidHVM"
vcpus = 1
vif = [ 'type=ioemu,bridge=eth1' ]
disk = [ 'phy:/dev/sdb5,hda,w!', 'phy:/dev/loop0,hdc:cdrom,r' ]
device_model = '/usr/lib/xen/bin/qemu-dm'
vnc=1
boot='d'
```

В силу того, что ядро Intrepid Server поддерживает паравиртуализацию, профайл, приведенный ниже, загрузит PV DomU, используя уже созданный образ.

```
name = 'IntrepidPV'
bootloader="/usr/bin/pygrub"
memory = 2048
disk = ['phy:/dev/sdb5,hda,w']
vif = [ 'bridge=eth1' ]
vfb = ['type=vnc,vncunused=1']
on_reboot = 'restart'
on_crash = 'restart'
```

```
# xm create intrepid.py
# vncviewer localhost:0
```

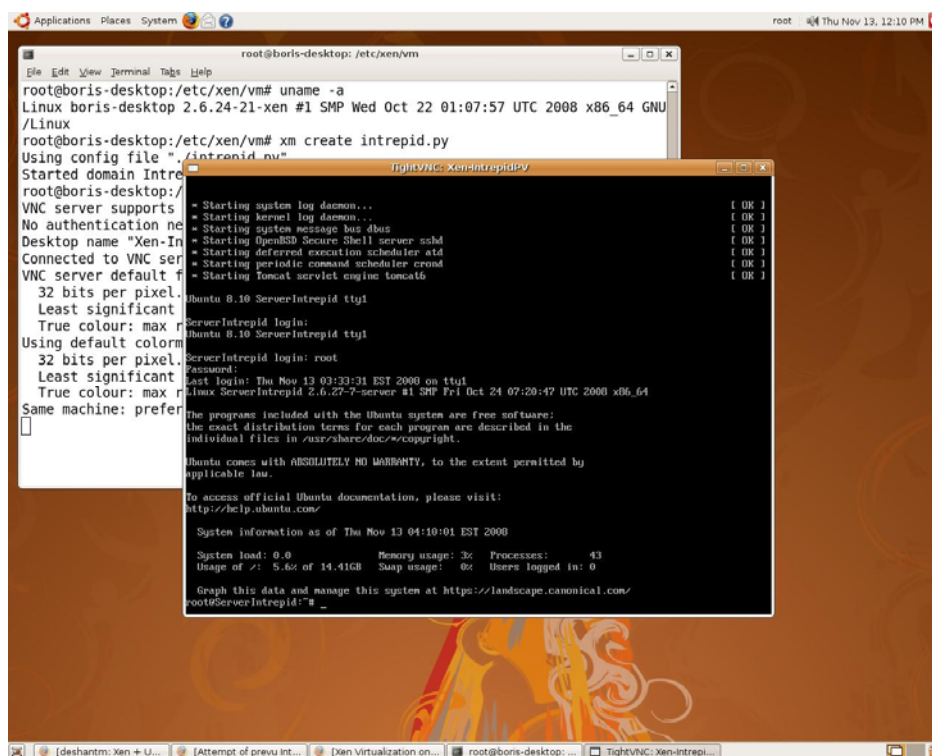


Рис. 2. Загрузка PV DomU, с использованием созданного образа

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <http://xer.com/module/newswire/view/111403/index.html>.

УДК 621.396.965.621.391.26

Е.А.Самойличенко

АЛГОРИТМ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ ПО ЭХОСИГНАЛУ

В настоящее время стали особенно актуальны задачи охраны объектов, находящихся либо целиком в водной среде, либо в непосредственной близости от водоёма, таких как: плавучие нефтедобывающие вышки и гостиничные комплексы, атомные и гидроэлектростанции и т.п. Для этих задач используются охранные системы, построенные на базе гидролокатора.

Гидролокатор – аппаратный комплекс для определения с помощью акустических сигналов положения подводных и плавучих объектов. Главными элементами гидролокатора являются подводный излучатель мощного акустического сигнала и чувствительный приемник, реагирующий даже на слабые отражения этого сигнала от погруженных в воду объектов.

Последнее десятилетие ознаменовалось дальнейшим развитием гидролокационных систем (ГЛС), которое опиралось на успехи, достигнутые в ряде областей науки, в частности, в области цифровых методов формирования и обработки сигналов. Существенное влияние на развитие ГЛС оказало и развитие элементной