User Mode Linux (UML):

An overview and experiences.

Matthew Grove
SLUG Tech Talk
Red Hat Europe
12th January 2007

Outline

- What is UML?
- A very brief overview of x86 virtualisation,
- UML Architecture,
- Running UML,
- Gonzo – a mini case-study.

What Is UML?

- UML is a port of the Linux kernel to Linux, Jeff Dike is maintainer and work started in 1999.
- Linux has been ported to a wide variety of architectures, for example:
  - Alpha, ARM, i386, IA64, M68K, MIPS, MIPS-64, PA-RISC, PowerPC, PowerPC64, IBM S/390, IBM S/390x, SH, SH64, SPARC, SPARC64, x86-64.
  - UML lets you compile the vanilla Linux kernel sources into a standard Linux user-land executable.
- UML is an example of a virtual machine (VM). In simple terms a VM allows multiple operating systems to share a single hardware resource.

Why Use Virtual Machines?

- Support heterogeneous applications using one physical machine (I.E. Win32 and Linux apps),
- Consolidate workloads,
- Allows the execution environment to be tailored to the application,
- Run legacy applications,
- Secure sandboxes for untrusted applications,
- By managing resources you can get QoS,
- Make system mobility easier to implement.

x86 Virtualisation

- The x86 was not designed with virtualisation in mind.
  - You must either trap and modify some calls (overhead),
  - Or modify the guest OS (can perform better).
- Two techniques for accessing resources:
  - Full virtualisation (looks like a normal x86),
  - Para-virtualisation (present real and virtual resources to the OS).
- Some new x86 chips contain some features to overcome some virtualisation issues.

UML Architecture

```
+-----------+-----------+----+
| Process 1 | Process 2 | ...|
+-----------+-----------+----+
|       Linux Kernel         |
+----------------------------+
|         Hardware           |
+----------------------------+
```

Normally your user-land processes talk to the Linux kernel which in turn accesses the hardware.

```
            +----------------+
            | Process 3 | ...|
+-----------+----------------+
| Process 1 | User-Mode Linux|
+----------------------------+
|       Linux Kernel         |
+----------------------------+
|         Hardware           |
+----------------------------+
```

Under UML your process talks to the UML kernel which then communicates with the host kernel in the same way that any user-land application would.

Some UML Details

- UML will execute on Linux >=2.2. The UML kernel does not need to match the host kernel (you could run 2.4 on 2.6).
- The Separate Kernel Address Space (SKAS) patch makes the UML kernel inaccessible from the guest making UML secure and suitable for honey-pots.

Gonzo – Mini Case-Study

- Situation:
  - I have consolidated most of my computing needs into one server.
  - This server is critical to life as we know it, because among other things it provides the email and the TV backend.
  - The server runs Debian Sarge.
- Requirements:
  - I wanted to have Internet facing services and provide remote access to home without having my server directly accessible on the net.
  - If possible I also wanted the option to easily host services not available in Sarge.

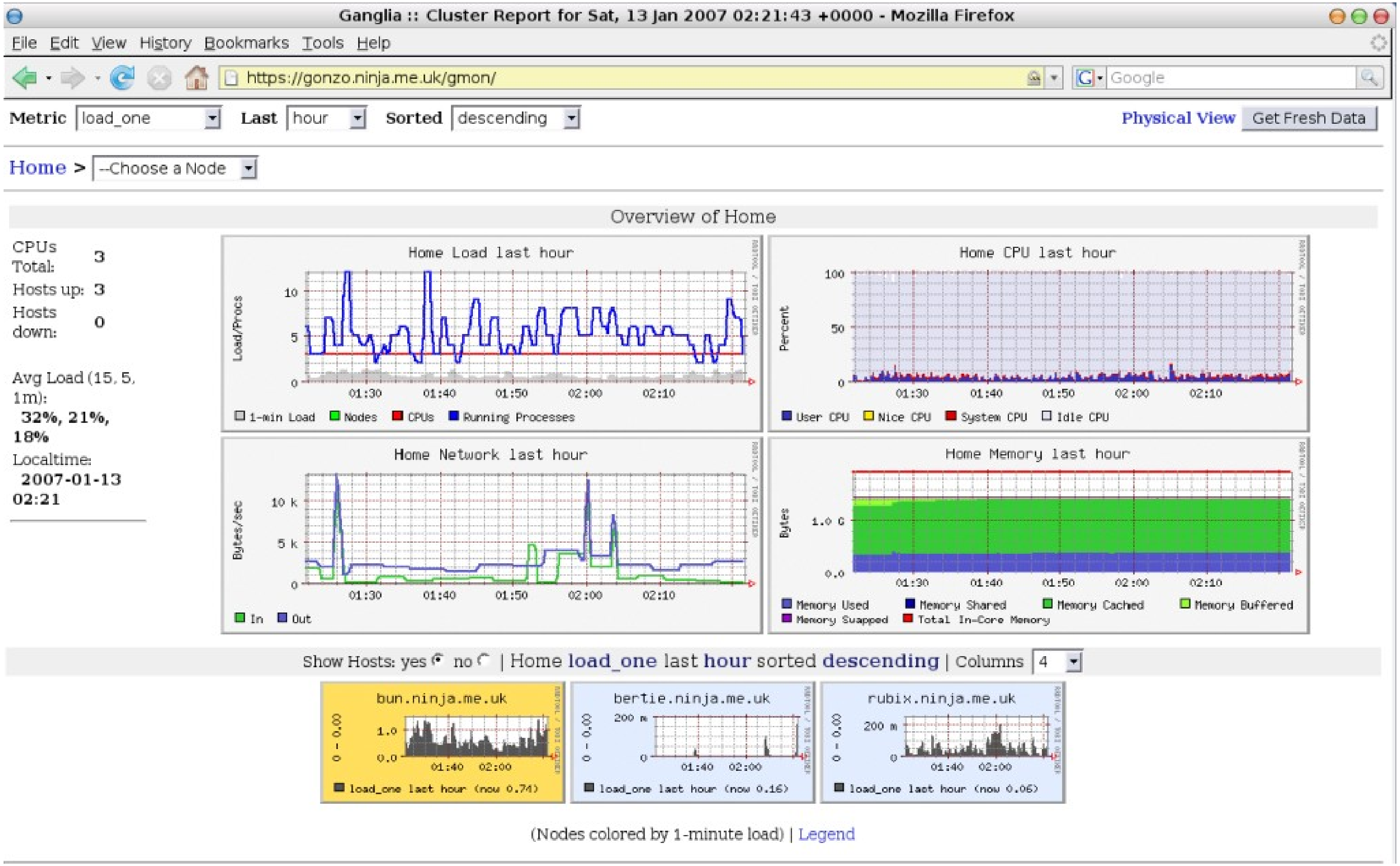
Why I Chose UML

- I didn’t want another server – I had 7 before consolidation efforts and for financial, space and relationship reasons no extra computers allowed!
- UML provided me:
  - A mechanism for running Debian Testing on my server along side Sarge.
  - A GPL solution using mature code,
  - Did not require changes to the host server OS or hardware (although I did use the SKAS patch).
  - No performance impact on the other duties of the host server.

Building, Installing and Running

1. Prepare the host system:
   - Apt-get the UML tools and configure networking.
2. Build a UML kernel,
   - UML is included in the mainline 2.6 kernel.
3. Install a guest operating system,
   - There are many options for this. I created a file to contain the guest OS. I used the debootstrap script to install Debian onto this file system.
4. Boot UML.
   -
     ```
     ./linux ubd0=sarge.disk ubd1=swap.disk
     con0=fd:0,fd:1 con=pts eth0=tuntap,tap0
     mem=128M umid=gonzo
     ```
5. Automate starting the machine.

gonzo.ninja.me.uk



Ganglia :: Cluster Report for Sat, 13 Jan 2007 02:21:43 +0000 - Mozilla Firefox

https://gonzo.ninja.me.uk/gmon/

Metric load_one Last hour Sorted descending

Home > --Choose a Node

Overview of Home

CPUs Total: 3
Hosts up: 3
Hosts down: 0

Avg Load (15, 5, 1m):
32%, 21%, 18%

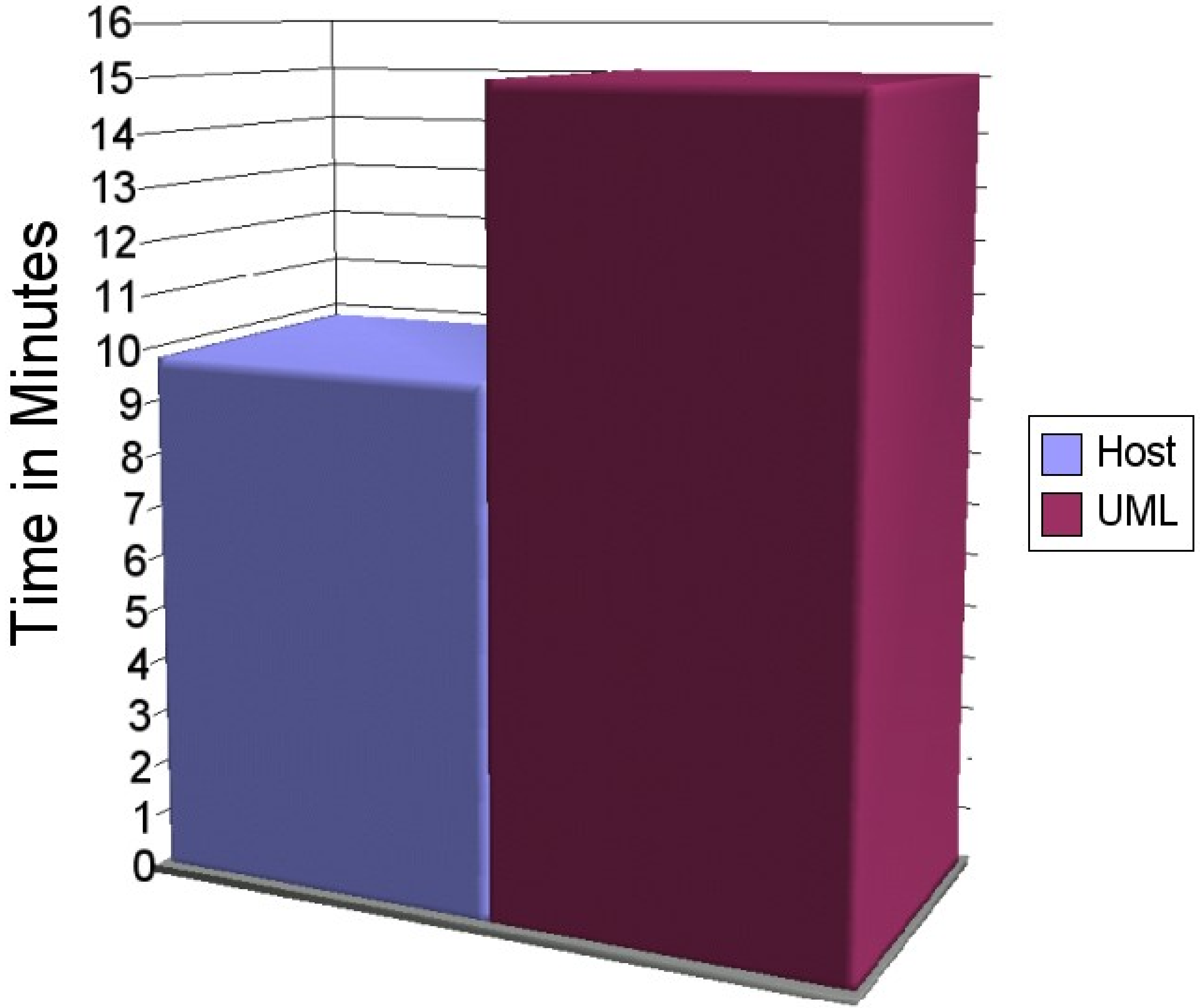
Localtime:
2007-01-13 02:21

Show Hosts: yes ● no ○ | Home **load_one** last **hour** sorted **descending** | Columns 4

(Nodes colored by 1-minute load) | Legend

Distributed Systems Group

Unscientific Performance Evaluation



Time to compile the Linux 2.6.18.4 kernel

Links

- UML project page:

  http://user-mode-linux.sourceforge.net/

- Running Debian inside of Debian with User-Mode Linux :

  http://eggdrop.ch/texts/uml/

- Bytemark (UML) Hosting:

  http://www.bytemark.co.uk/

- The UML book:

  User Mode Linux, Jeff Dike. Prentice Hall PTR, 2006 , ISBN-10: 0131865056.