Troubleshooting “like a boss” con Sysdig

…

Ovvero come fare tracing a basso livello e vivere felici.

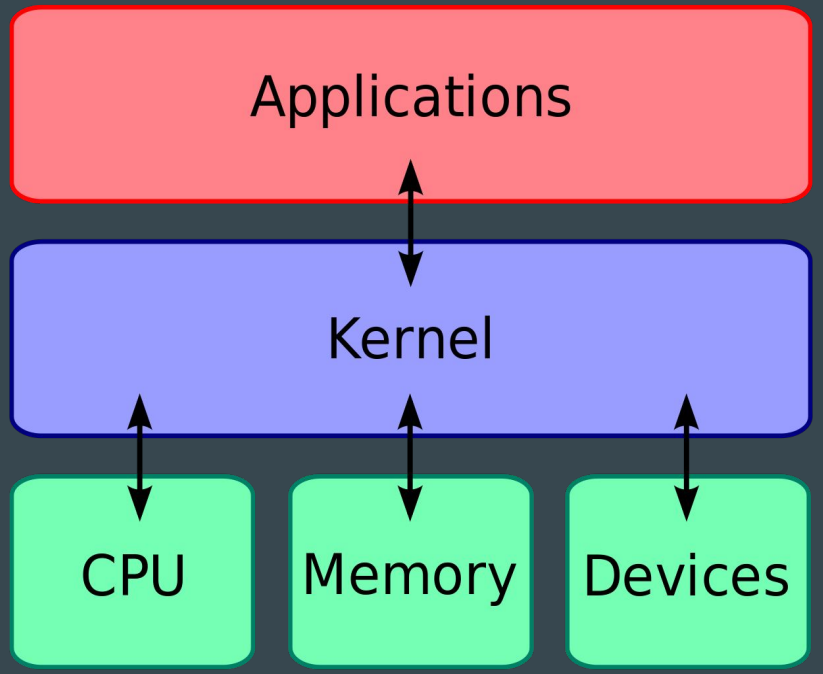
Pasquale De Rose 24/10/2015

Il Kernel

Il Kernel è la base su cui è costruito un Sistema Operativo, ogni operazione di input/output e di gestione dell'hardware è gestita da quest'ultimo.

Alcuni Esempi:

- Scrittura e lettura su supporti (Hard Disk, Dvd).
- Processare l'input dell'utente (tastiere, mouse, touchscreen).
- Gestisce le periferiche collegate al computer.
- Permette la connessione alla rete.



Pasquale De Rose 24/10/2015

Allora Linux?

Linux è un **Kernel Monolitico Open Source** inizialmente programmato da Linus Torvalds a partire dai primi anni novanta per sopperire alle troppe mancanze di Minix.

A differenza di quanto si possa pensare Linux non è tutto il sistema operativo ma solo il kernel del sistema operativo.

Pasquale De Rose 24/10/2015

Esistono Altri Kernel?

Certo ecco qualche esempio:

- La famiglia BSD ({Free|Net|Open|Dragonfly}BSD)
- Haiku
- Inferno
- Darwin (la base di Os X)
- Windows NT

E se il Kernel e le applicazioni non vanno d'accordo?

Un Kernel ha milioni di righe di codice, svolge numerosissimi compiti e soprattutto, non si lamenta quando lo stressiamo.

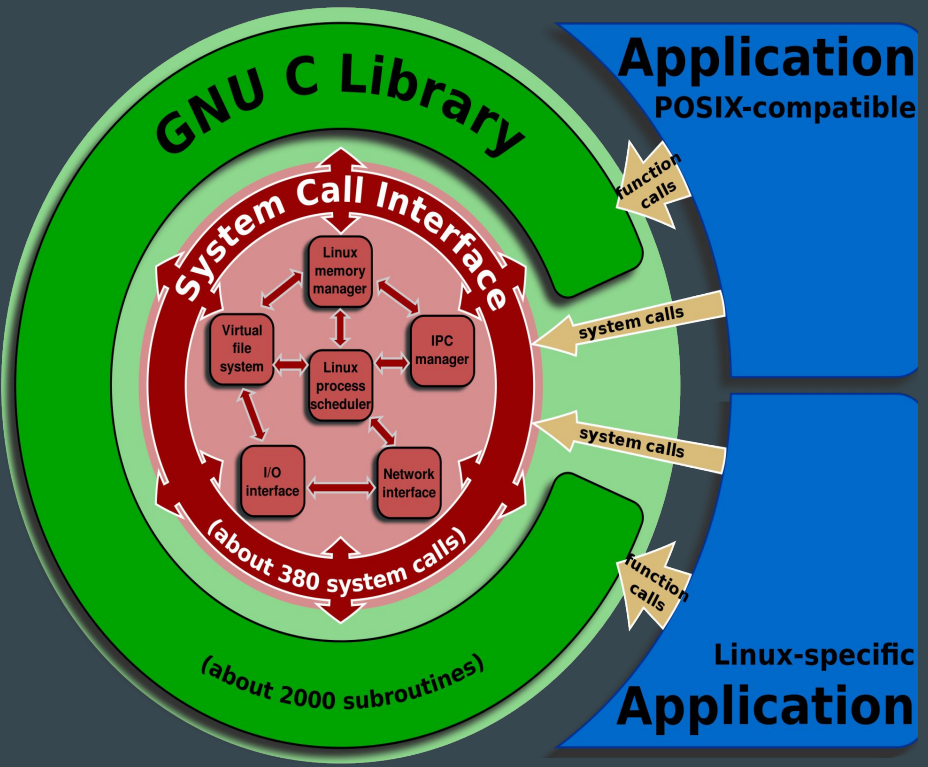
Spesso però la mediazione fra Kernel e Applicazioni può avere problemi portando il sistema operativo ad essere instabile, sopratutto se l'applicazione lavora a basso livello.

Pasquale De Rose 24/10/2015

La Soluzione?

Pasquale De Rose 24/10/2015

Le Chiamate di Sistema

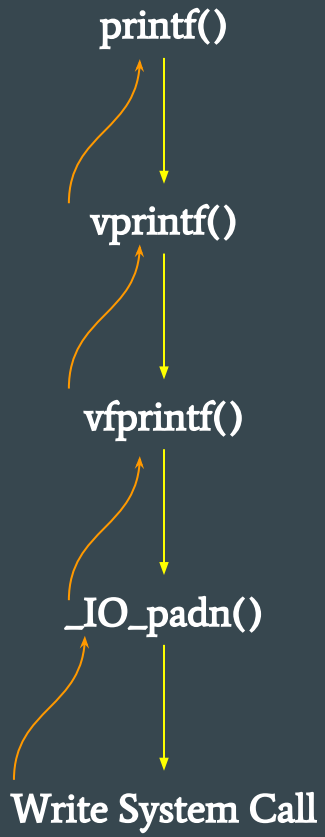


La comincazione fra applicazioni a livello utente e il Kernel avviene attraverso le chiamate di sistema, queste richiedono al Kernel di svolgere un determinato servizio.

Nel Kernel Linux tutto ruota intorno alla Gnu C Library, questa libreria si occupa di interfacciarsi con il kernel attraverso le chiamate di sitema.

Pasquale De Rose 24/10/2015

Un esempio Pratico di Chiamata Di Sistema



Pasquale De Rose 24/10/2015

Cosa fà SysDig?

SysDig è un framework che ci permette di analizzare ogni aspetto delle operazioni svolte dal nostro **Kernel** catturando e analizzando ogni chiamata di sistema che viene fatta.

SysDig altro non è che l'unione di tool come:

- **Strace**
- **tcpdump**
- **htop**
- **iftop**
- **lsof**

Più tanta altra roba figa (scripting in Lua)

Pasquale De Rose 24/10/2015

I Componenti di SysDig

- **Modulo Del Kernel** -> Permette di tracciare le chiamate di sistema, è necessario caricarlo se si vuole usare SysDig.

- **SysDig** -> Tool da riga di comando e principale modo per interfacciarsi al programma

- **CSysDig** -> Tool grafico

Pasquale De Rose 24/10/2015

Facciamo Qualche Esempio

Pasquale De Rose 24/10/2015

Analziamo Il Nostro Sistema

Stampiamo tutte le informazioni del nostro kernel:

sudo sysdig

Pasquale De Rose 24/10/2015

Mmmm…. Troppe informazioni, meglio essere più precisi.

Pasquale De Rose 24/10/2015

Recuperiamo solo le informazioni riguardanti Firefox

```
sudo sysdig proc.name=firefox
```

- **proc.name ->** parametro che specifica il processo da analizzare

Pasquale De Rose 24/10/2015

Ancora più precisi!

Recuperiamo tutte le informazioni del processo firefox nelle prossime 100 operazioni

```
sudo sysdig -n 100 -v proc.name=firefox
```

- n -> il numero delle operazioni su cui operare
- v -> modalità verbosa

Pasquale De Rose 24/10/2015

I Chisel

I **Chisel** sono script scritti in Lua e interpretati da **SysDig** che ci permettono, usando delle funzioni integrate nel **framework**, di recuperare e operare su vari aspetti del nostro sistema.

Pasquale De Rose 24/10/2015

Recuperiamo la lista dei Chisel che possiamo usare con SysDig

sudo sysdig -cl

- cl -> Stampa la lista dei Chisel disponibili

Pasquale De Rose 24/10/2015

Stampiamo le operazioni di lettura e scrittura dei processi

Recuperiamo i processi di scrittura e di lettura nelle prossime 100 operazioni.

sudo **sysdig** -c echo_fds -n 100

- n -> il numero delle operazioni su cui operare
- c -> specifica il Chisel da usare

Pasquale De Rose 24/10/2015

Sempre più precisi!

Recuperiamo i processi di scrittura e di lettura di Firefox nelle prossime 100 operazioni.

sudo **sysdig** -c echo_fds -n 100 -j proc.name=firefox

- c -> specifica il Chisel da usare
- n -> il numero delle operazioni su cui operare
- j -> stampa l'output in un file Json
- proc.name -> parametro che specifica il processo da analizzare

Pasquale De Rose 24/10/2015

E se l'interfaccia testuale vi sembra poco

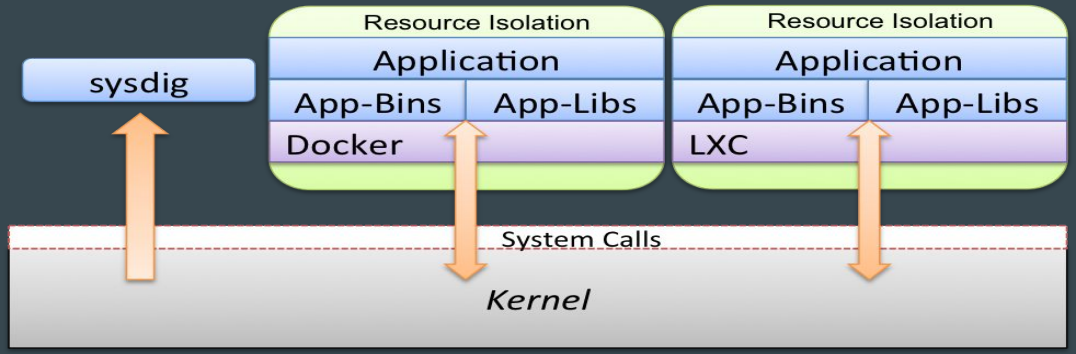
Stampiamo lo spettrogramma che ci mostra la latenza della CPU.

sudo sysdig -c spectrogram

- c -> specifica il Chisel da usare

Pasquale De Rose 24/10/2015

Lavoriamo con i Container



SysDig nasce come sistema di monitoring per container, quindi ogni comando può essere eseguito su container **Docker** e **LXC.**

sudo **sysdig** -c topprocs_cpu container.name=client

- c -> specifica il Chisel da usare
- container.name -> specifica il nome del container

Pasquale De Rose 24/10/2015

Un esempio di sorgente di un Chisel

Pasquale De Rose 24/10/2015

Per Concludere

- Articolo su Docker e la virtualizzazione http://hlcs.it/2015/02/24/hacklab-x-anno-2-edizione-speciale-1/
- Seminario su Docker al TAG → https://hlcs.it/2015/06/18/introduzione-a-docker-tag-cs/
- Blog Personale → https://j-lemon.github.io/Punk_Overflow/
- Sito dell'HLCS → http://hlcs.it
- Sito ufficiale SysDig → http://www.sysdig.org/
- Twitter → @JamesSlimLemon

Grazie per l'attenzione!

Pasquale De Rose 24/10/2015