Thinking and Practice on Linux System Performance

Barry Song <baohua@linuxep.com>

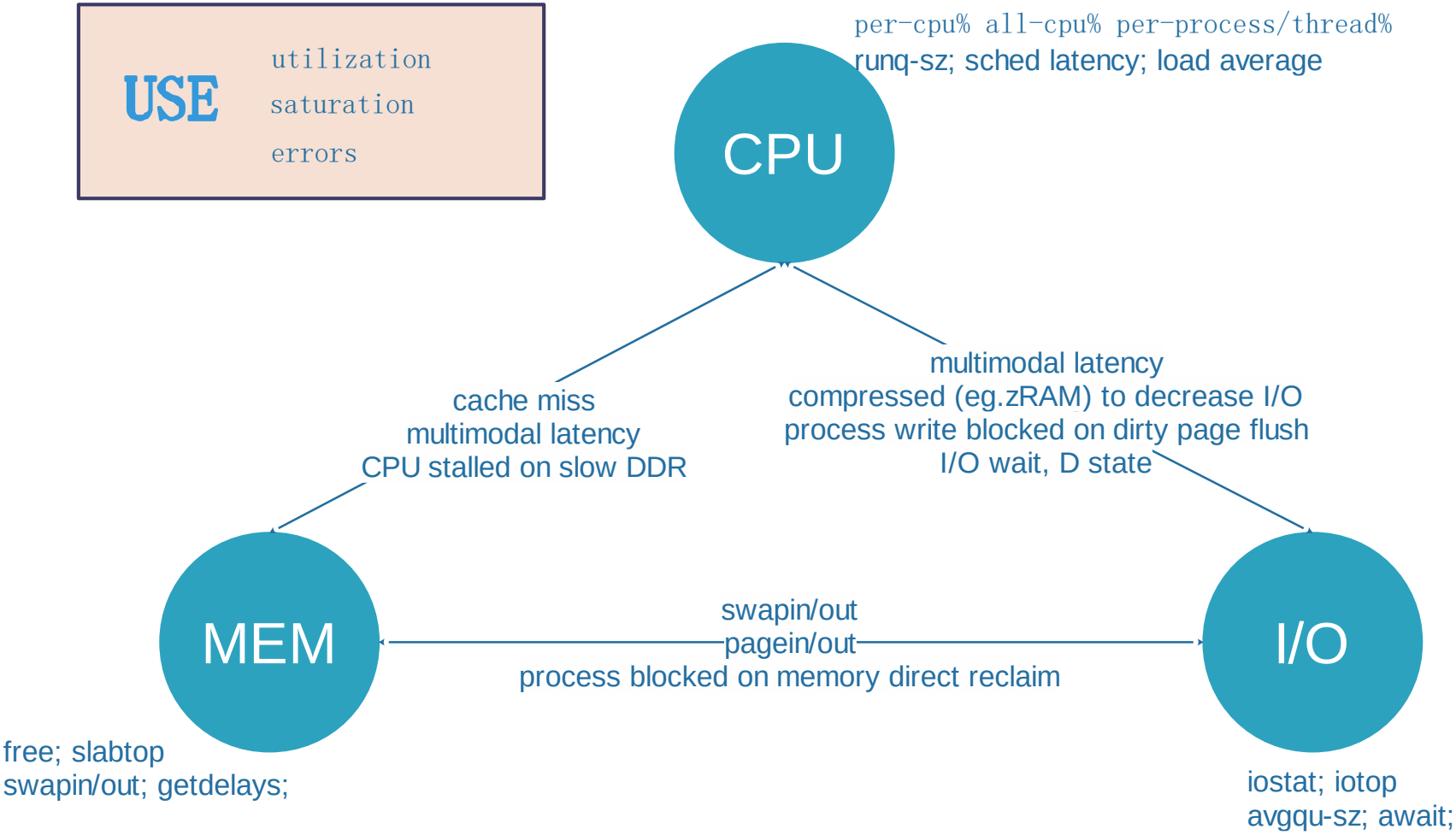
http://www.linuxep.com

LEP(Linux Easy Profiling)

An open-sourced all-in-one toolbox
for Linux/Android performance profiling & visualization

Life is hard
So is performance profiling

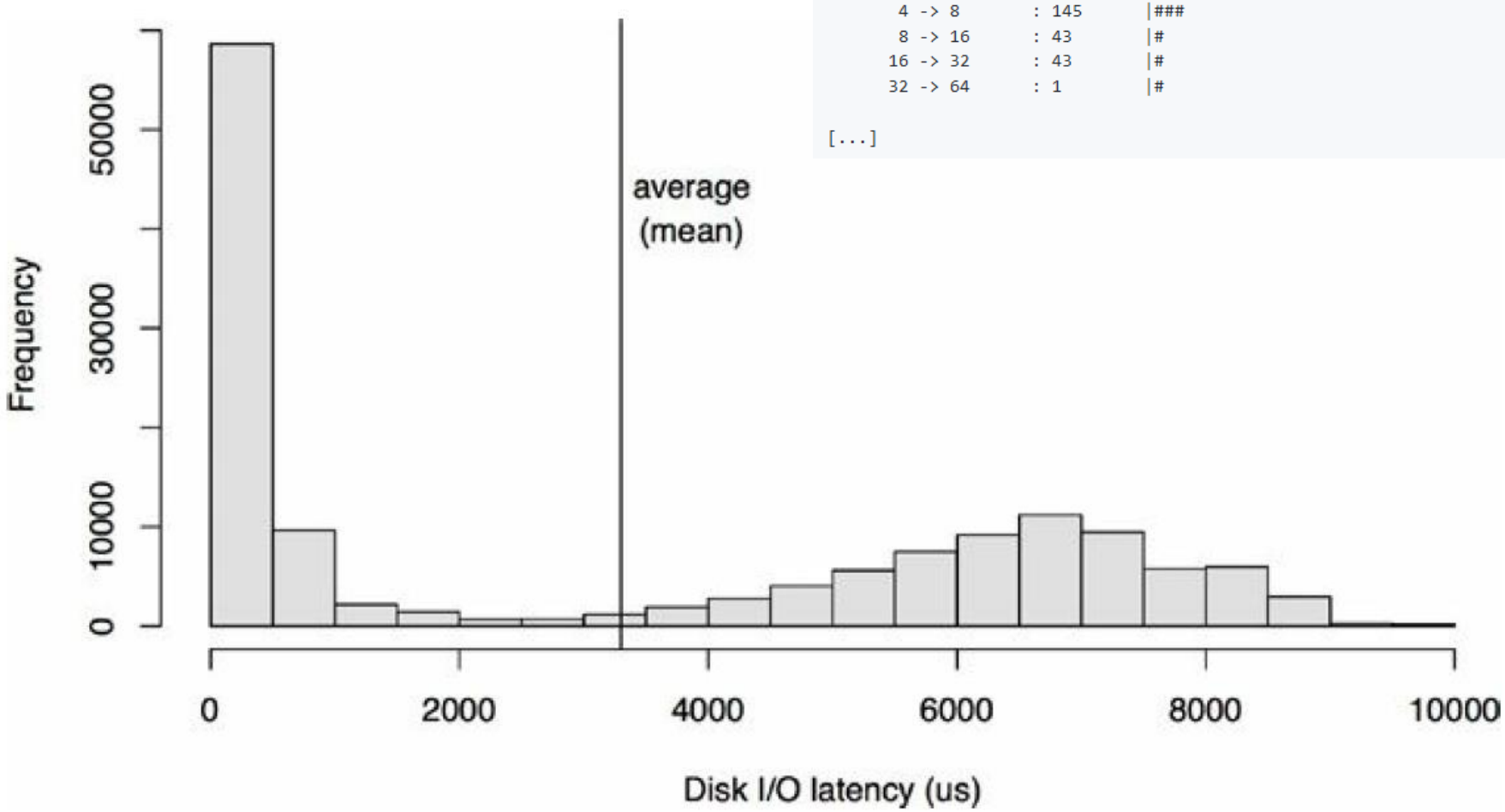
CPU、Memory和I/O



Typical Latency Profiles - Multimodal

Multimodal Distribution

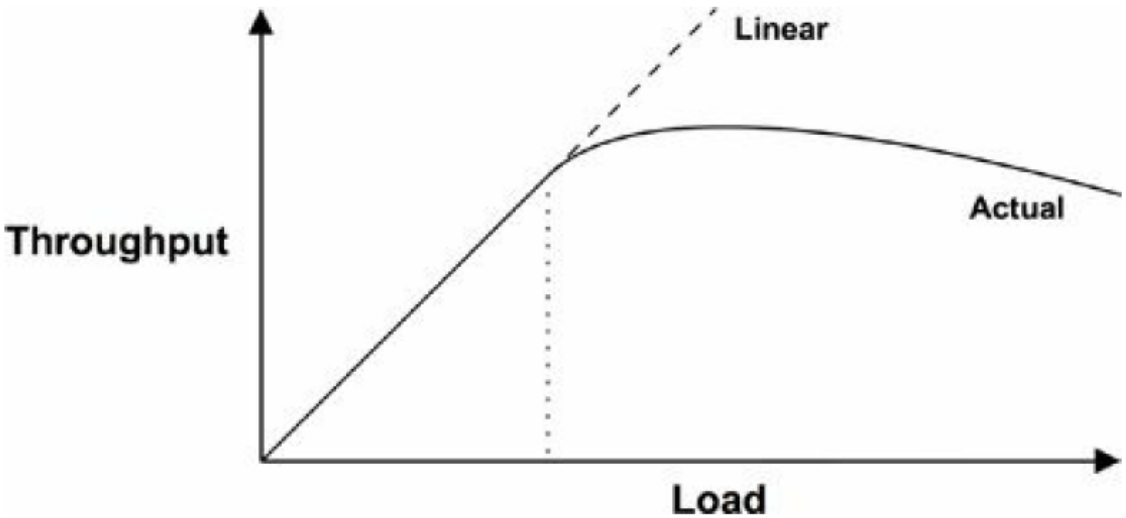
```
# ./iolatency -Q
Tracing block I/O. Output every 1 seconds. Ctrl-C to end.

  >=(ms) .. <(ms)   : I/O      |Distribution                         |
       0 -> 1       : 1913     |######################################|
       1 -> 2       : 438      |#########                             |
       2 -> 4       : 100      |##                                    |
       4 -> 8       : 145      |###                                   |
       8 -> 16      : 43       |#                                     |
      16 -> 32      : 43       |#                                     |
      32 -> 64      : 1        |#                                     |

[...]
```



Typical Throughput Profile

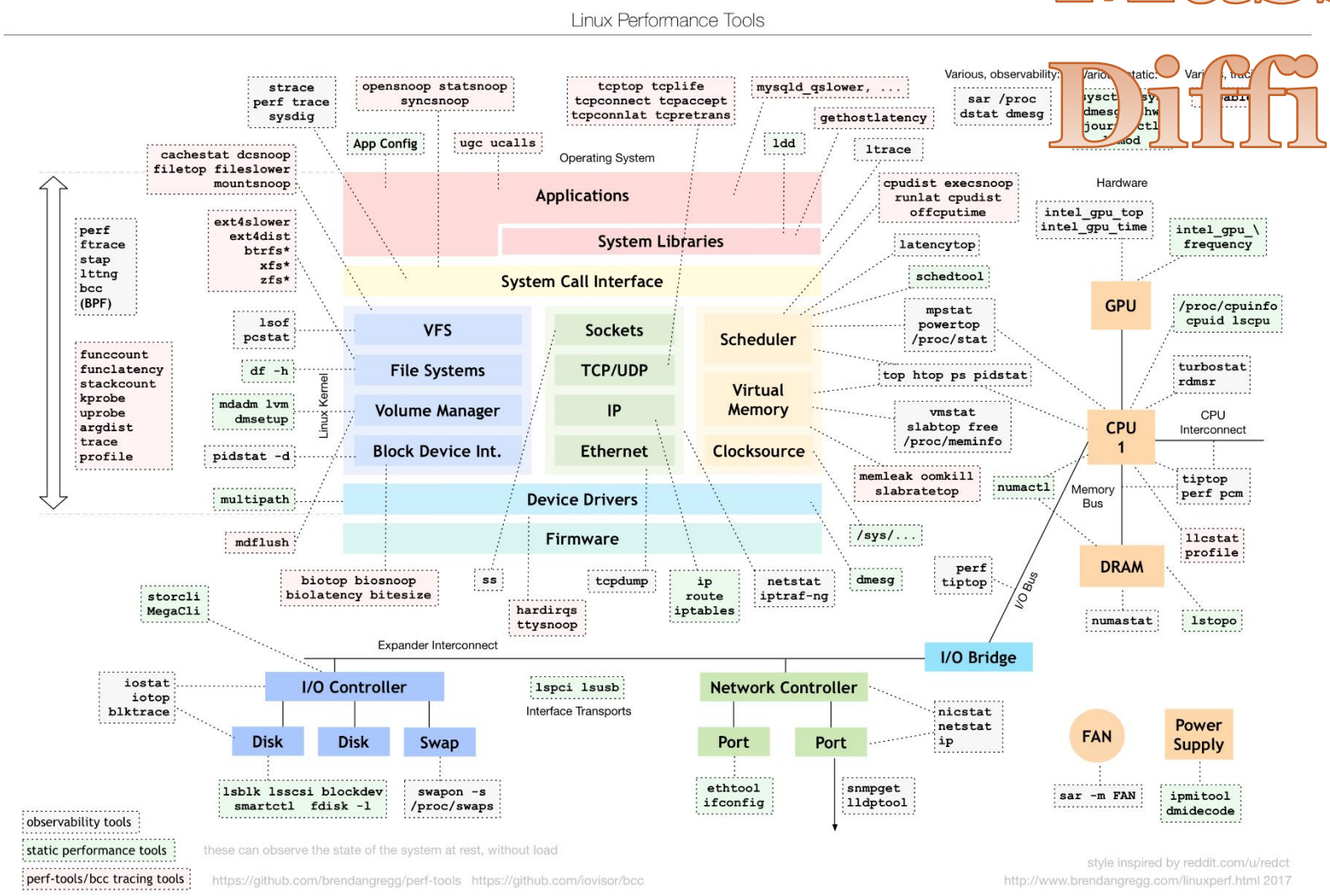
Linear increasment -> *knee point/ non-linear increasment -> decrease*

Enough resources -> *resource contention ->* increased contention



Linux performance tools

Massive Difficult



Show me

Linux experts

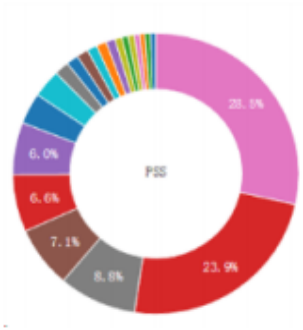
Help me

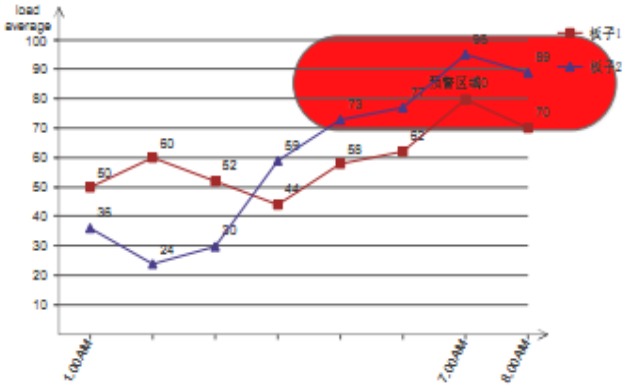
Familiar with Linux

Save me

Not familiar with Linux

PID	USER	PR	NI
3005	baohua	20	0
1979	root	20	0
1771	root	0	-20
3425	baohua	20	0
3595	**baohua**	**20**	**0**
3045	baohua	20	0
1	root	20	0
2	root	20	0
3	root	20	0





A

All-in-one

B

Visualization

C

Interactive

D

Historical Record

E

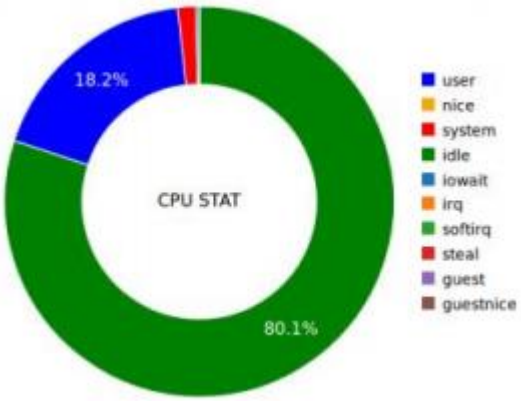
Change/Trendency

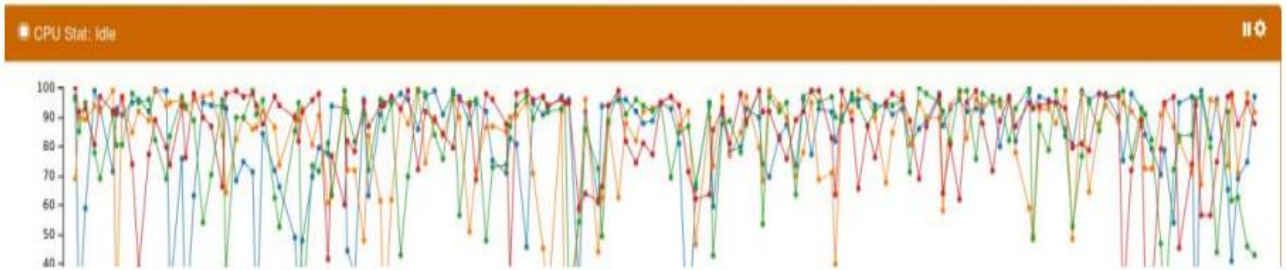
F

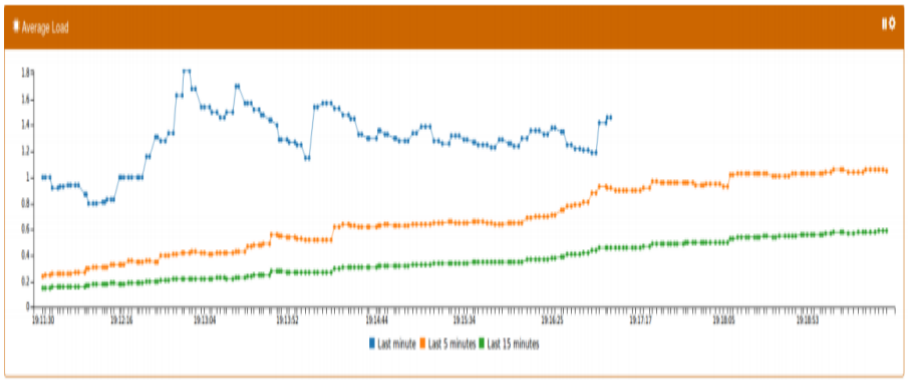
Alarm/Dignose

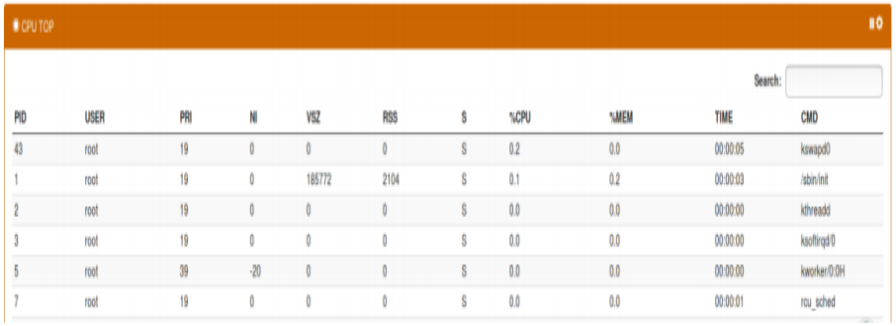
Expert Knowledge Base

CPU Perspective







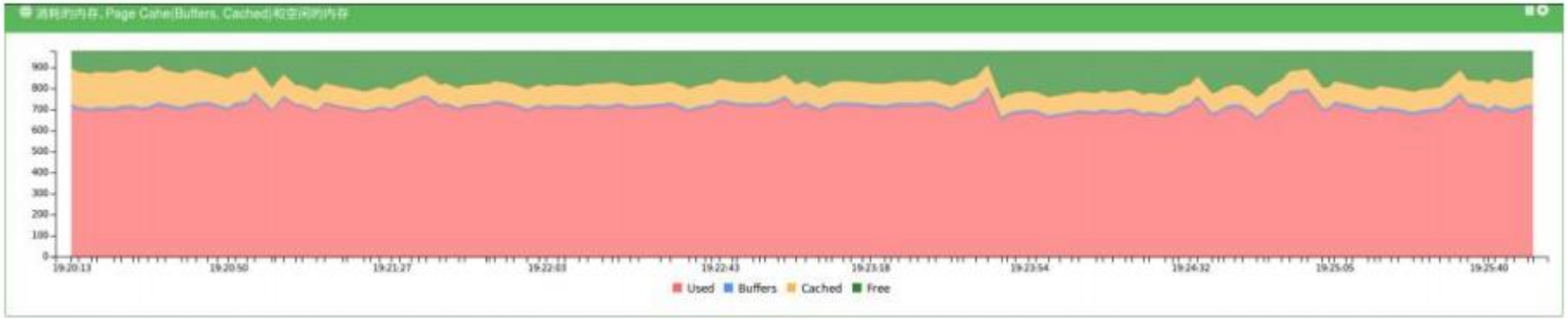


CPU TOP

Search:

PID	USER	PRI	NI	VSZ	RSS	S	%CPU	%MEM	TIME	CMD
43	root	19	0	0	0	S	0.2	0.0	00:00:05	kswapd0
1	root	19	0	185772	2104	S	0.1	0.2	00:00:03	/sbin/init
2	root	19	0	0	0	S	0.0	0.0	00:00:00	kthreadd
3	root	19	0	0	0	S	0.0	0.0	00:00:00	ksoftirqd/0
5	root	39	-20	0	0	S	0.0	0.0	00:00:00	kworker/0:0H
7	root	19	0	0	0	S	0.0	0.0	00:00:01	rcu_sched

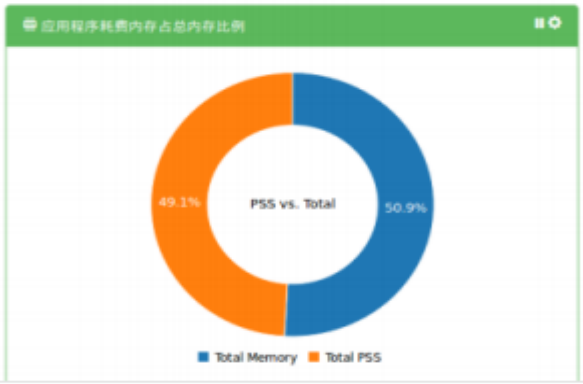
MEM Perspective



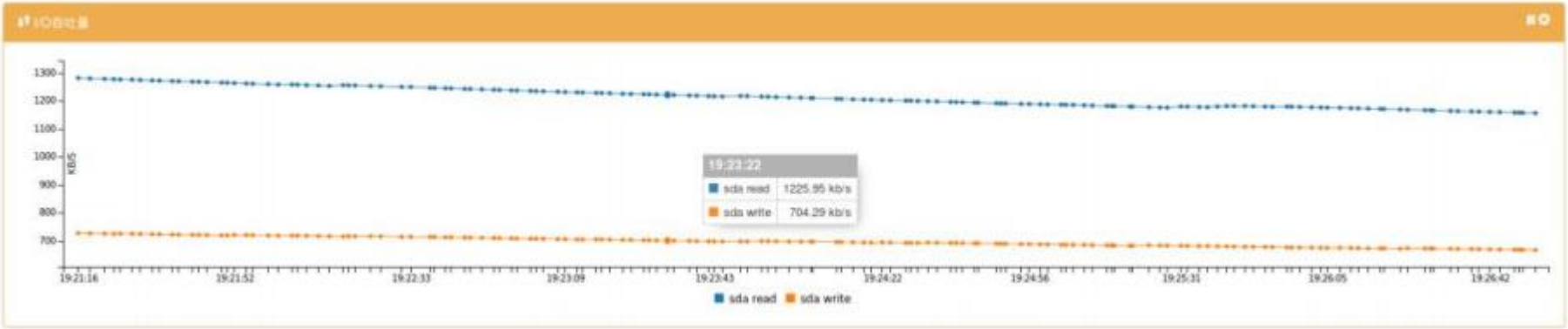
内存消耗(单位: MB)

Search:

PID	VSS	RSS	PSS	USS	CMDLINE
14493	2638712	364712	351430	343036	/usr/lib/firefox/firefox
11766	403400	33808	24439	17060	/usr/lib/xorg/Xorg
13089	669044	17736	14340	14080	compiz
11720	655404	11264	9728	9712	/usr/bin/dockerd
13591	49088	7508	6468	6412	./lepd
14480	135760	9140	6914	6316	uwsgi
14482	135744	8872	6718	6144	uwsgi
14478	139948	8340	6201	5692	uwsgi



IO Perspective



IO Top Table

Search:

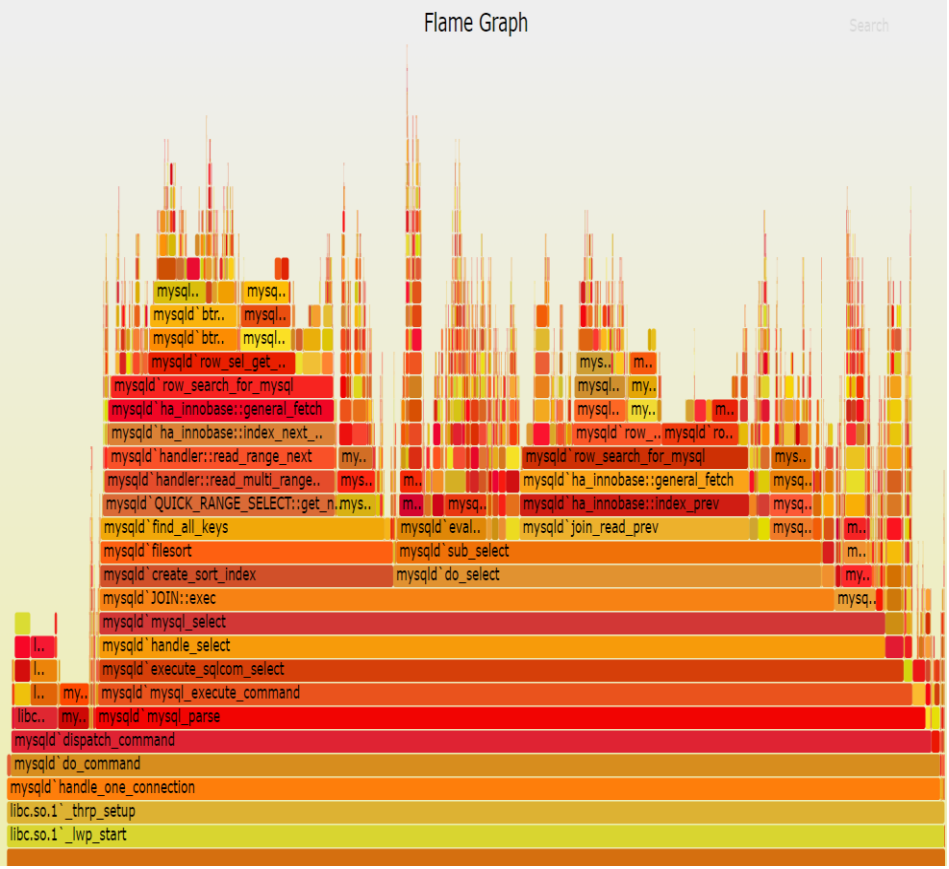
TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO	COMMAND
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	/sbin/init auto noprompt
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthread]
3	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/]
5	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/0:0]
7	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_sche]
8	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_b]

Perf Perspective

基于Symbol的时间分布 (perf top)

Search:

Command	Overhead	Shared Object	Symbol
malloc	53.53%	malloc	[.] main
swapper	32.87%	[kernel.kallsyms]	[k] native_safe_halt
malloc	12.50%	[kernel.kallsyms]	[k] clear_page
malloc	0.15%	[kernel.kallsyms]	[k] free_pages_prepare
malloc	0.13%	[kernel.kallsyms]	[k] get_page_from_freelist
malloc	0.10%	[kernel.kallsyms]	[k] _cond_resched
malloc	0.08%	[kernel.kallsyms]	[k] clear_huge_page
malloc	0.05%	[kernel.kallsyms]	[k] __do_page_fault
malloc	0.05%	[kernel.kallsyms]	[k] _raw_spin_lock
swapper	0.05%	[kernel.kallsyms]	[k] __do_softirq
malloc	0.03%	[kernel.kallsyms]	[k] page_add_new_anon_rmap
swapper	0.03%	[kernel.kallsyms]	[k] refresh_cpu_vm_stats
free	0.02%	[kernel.kallsyms]	[k] do_exit
free	0.02%	[kernel.kallsyms]	[k] page_add_file_rmap
free	0.02%	ld-2.19.so	[.] 0x0000000000009e2a
free	0.02%	libc-2.19.so	[.] 0x000000000007eae8
gapd	0.02%	[kernel.kallsyms]	[k] __fdget_raw
gapd	0.02%	[kernel.kallsyms]	[k] futex_wait
malloc	0.02%	[kernel.kallsyms]	[k] __do_softirq
malloc	0.02%	[kernel.kallsyms]	[k] lru_cache_add

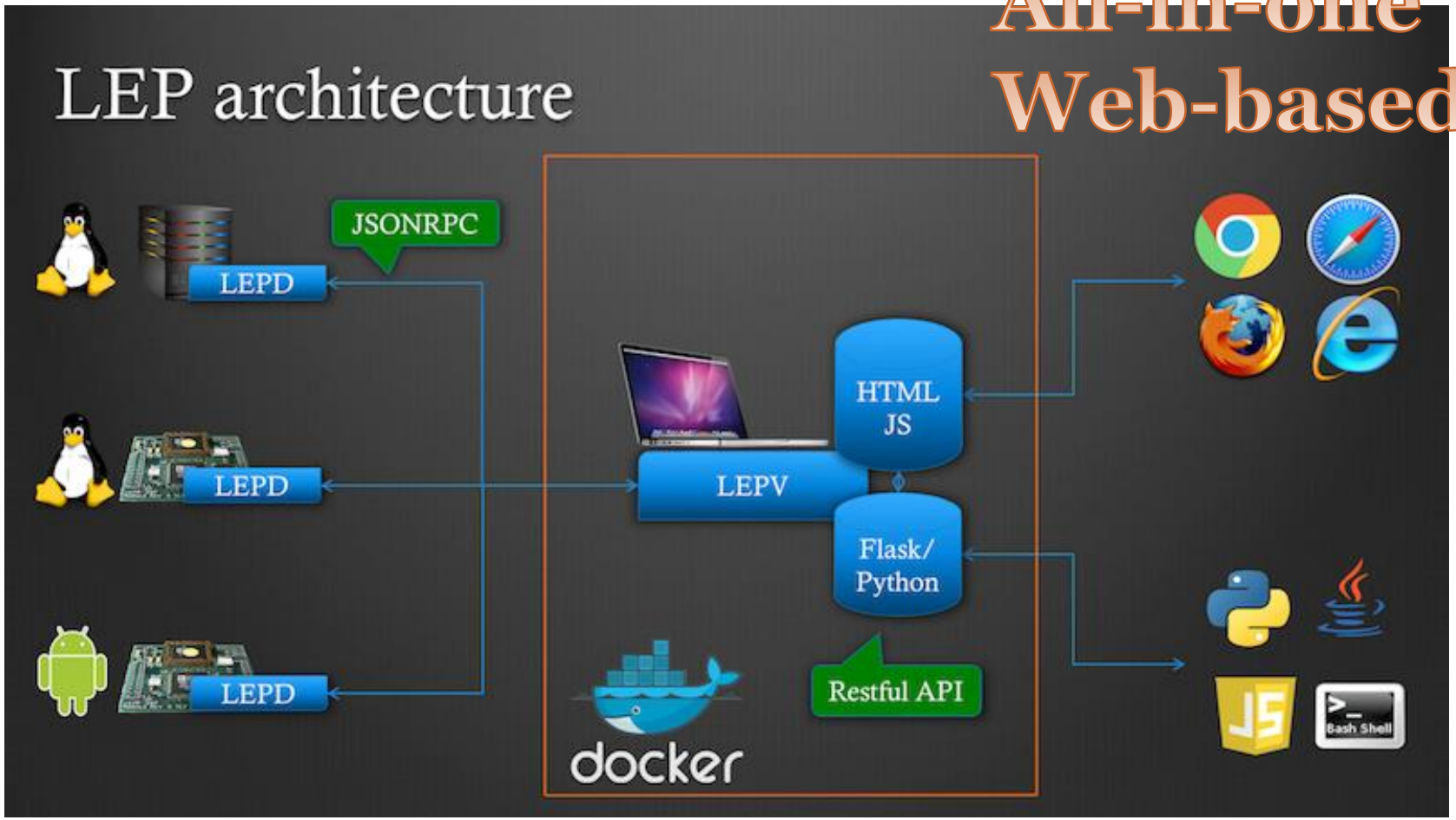


Cgroup Perspective

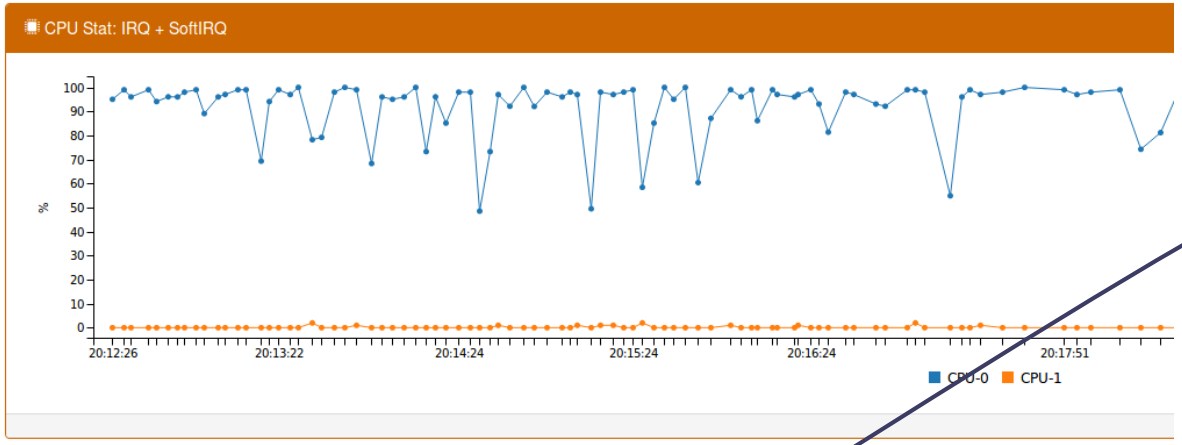
- Demonstrate subsystem(cpu,mem,io,network)
- Demonstrate hierarchy
- Demonstrate their accounting

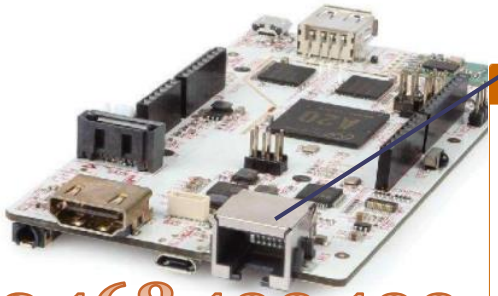
LEP(Linux Easy Profiling)

Easy
All-in-one
Web-based

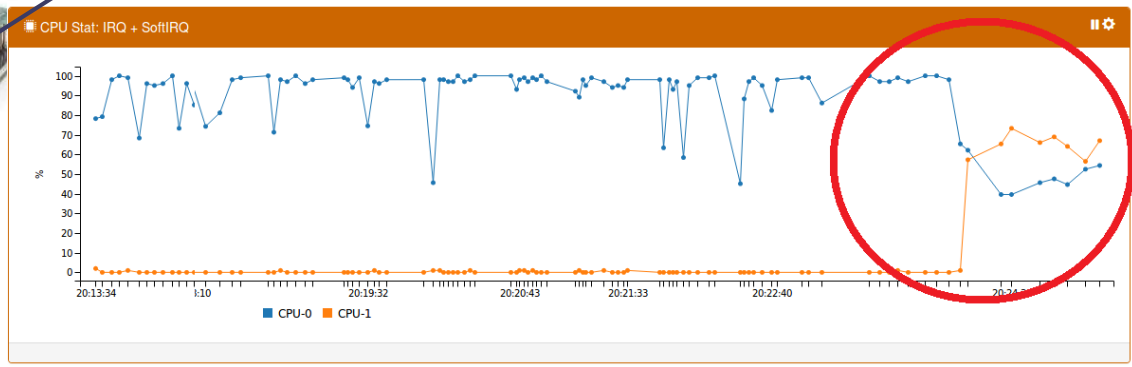


demo: improve network throughput using LEP





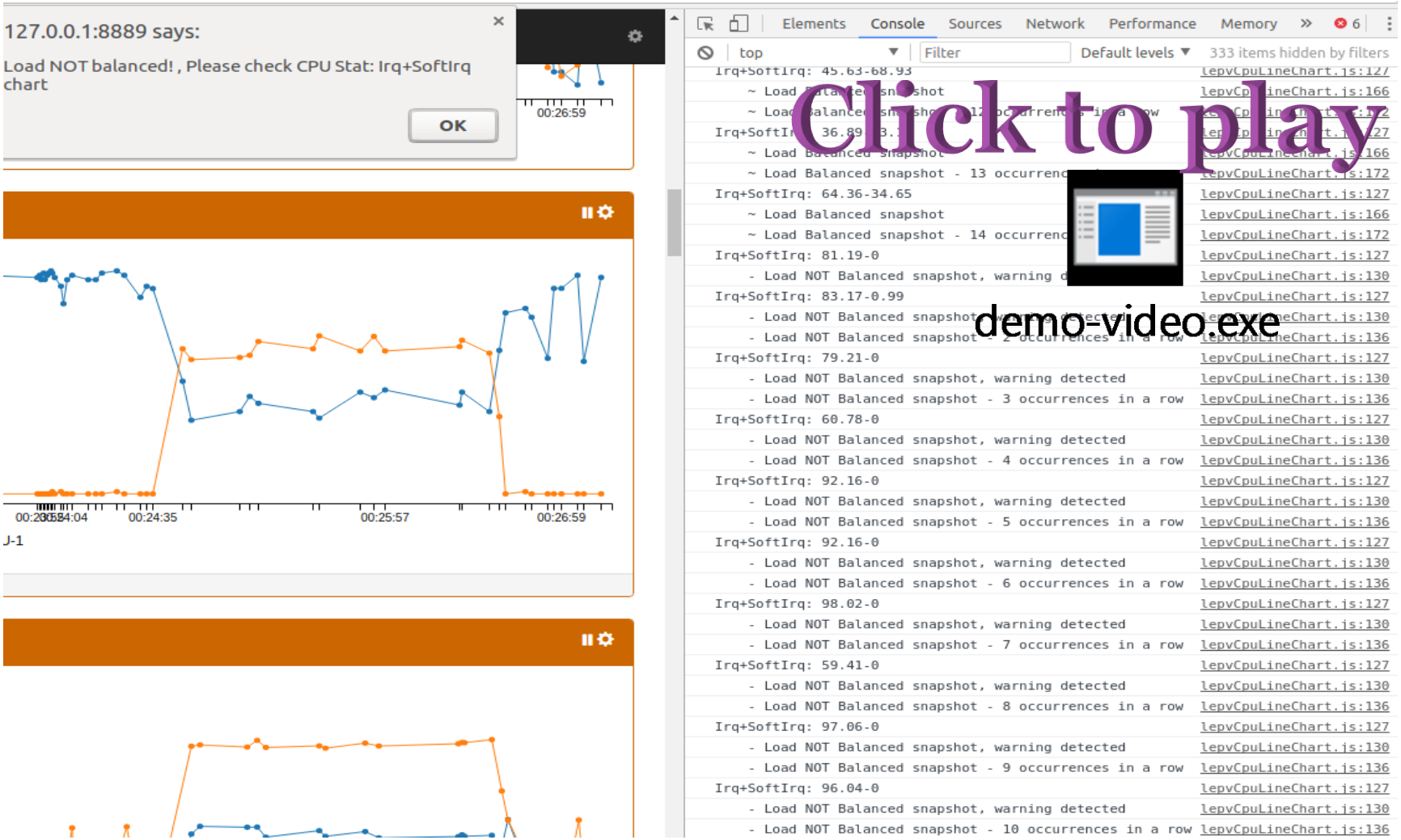
192.168.100.100



[Related Doc. 《LEP与负载均衡—— 以PCDUINO实际案例来使用LEP提高网络带宽》](http://mp.weixin.qq.com/s/FQKFRcMNDk6VE7ihnnvBPw)

http://mp.weixin.qq.com/s/FQKFRcMNDk6VE7ihnnvBPw

A snapshot of LEPV for the demo



ToDo

Deeply data analysis and system status alert
Database
Scenerios-based profiling
Latency analysis(off-cpu io, scheduler runqueue, memory reclaim)
QA/Test

More info

- Homepage： http://www.linuxep.com/
- Code ： https://github.com/linuxep

Thanks