PREDICTION OF PROCESS PERFORMANCE USING ML

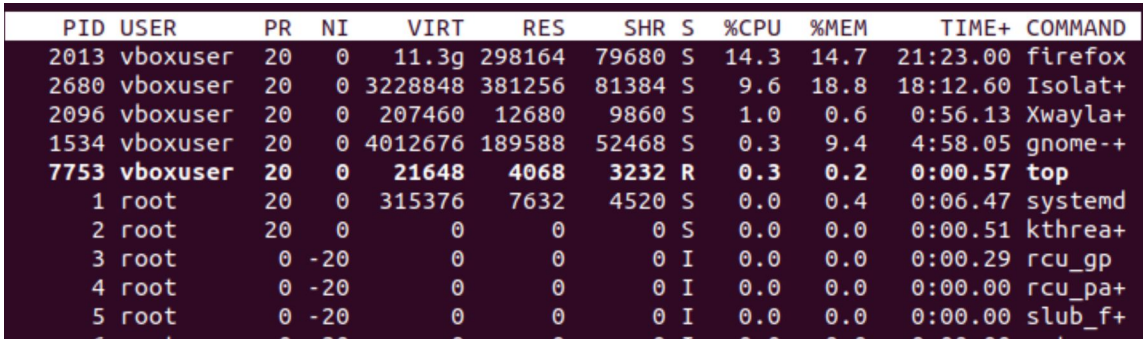
Jiaqian Huang

About Idea / topic (1)

1. Inspired by two papers named *“On the Use of ML for Blackbox System Performance Prediction”* and *“Applying Machine Learning Techniques to Improve Linux Process Scheduling”*
2. Important and optimal to predict the process performance
    a. Early detection of performance issues
    b. Improved resource utilization
    c. Better capacity planning
    d. Reduced downtime
    e. Improved user experience
    f. Cost savings

About Idea / topic (2)

1. Process performance can be measured by:
    a. CPU usage, memory usage, disk I/O, and network I/O.



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2013	vboxuser	20	0	11.3g	298164	79680	S	14.3	14.7	21:23.00	firefox
2680	vboxuser	20	0	3228848	381256	81384	S	9.6	18.8	18:12.60	Isolat+
2096	vboxuser	20	0	207460	12680	9860	S	1.0	0.6	0:56.13	Xwayla+
1534	vboxuser	20	0	4012676	189588	52468	S	0.3	9.4	4:58.05	gnome-+
7753	**vboxuser**	**20**	**0**	**21648**	**4068**	**3232**	**R**	**0.3**	**0.2**	**0:00.57**	**top**
1	root	20	0	315376	7632	4520	S	0.0	0.4	0:06.47	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.51	kthrea+
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.29	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_pa+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_f+

APPROACH

1. Collect large amount of data
2. Train machine learning models
3. Predict the performance

Steps to approach

1. Write a linux kernel module in C language
    a. Log process-related data such as CPU usage, memory usage, disk I/O, and network I/O

```
[ 2559.174759] proclog: loading out-of-tree module taints kernel.
[ 2559.178474] proclog: module verification failed: signature and/or required k
ey missing - tainting kernel
[ 2559.195497] Process logger module loaded
[ 2559.195511] Process: systemd (pid: 1)
[ 2559.195513] Process: kthreadd (pid: 2)
[ 2559.195515] Process: rcu_gp (pid: 3)
[ 2559.195516] Process: rcu_par_gp (pid: 4)
[ 2559.195518] Process: slub_flushwq (pid: 5)
```

```c
static void log_processes(void) {
    struct task_struct *task;
    for_each_process(task) {
        printk(KERN_INFO "Process: %s (pid: %d)\n", task->comm, task->pid);
    }
}
```

Steps to approach

1. Build ML models in Python
    a. Preprocess data
        i. Linear assumption, remove noise and collinearity, gaussian distributions, rescale inputs
    b. Use `sklearn` for ML models
        i. Linear regression
        ii. Random forest
2. Present the prediction
    a. Data visualization such as D3.js (TBD)

Critical thinking

1. Is the prediction made by the data I collect general enough?
2. Does our approach make it simple to predict the performance in ML?
3. Does our model provide accurate prediction of performance?

Contribution

1. Project design - JH
2. Linux Kernel Module - JH
3. ML Models - JH
4. Data preprocessing - JH
5. Data visualization (TBD) - JH

Sources and Tutorials

1. https://www.usenix.org/conference/nsdi21/presentation/fu
2. https://ieeexplore.ieee.org/document/4085157

1. http://tldp.org/LDP/lkmpg/2.6/html/index.html
2. https://scikit-learn.org/stable/modules/linear_model.html

Q&A