Trace Analysis

by

- 2K18/MC/122 Vedank Goyal
- 2K18/MC/103 Sarthak Singh

In [97]:
```
import pandas as pd
import requests
from io import StringIO
```

Reading the Synchronized Trace Set

Reading the trace file from google drive link

In [109]:
```
orig_url='https://drive.google.com/file/d/1AofJk6yCrtjkgTieXbq8a1EM-SqWHK5X/view?usp=sharing'

file_id = orig_url.split('/')[-2]
dwn_url='https://drive.google.com/uc?export=download&id=' + file_id
url = requests.get(dwn_url).text
csv_raw = StringIO(url)
df = pd.read_csv(csv_raw)
df
```

Out[109]:

	Trace	Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source	packet_in	packet_out
0	C2	12:37:12.219 698 599	channel0_0	0	irq_handler_entry	irq=21, name=snd_intel8x0	0	20	NaN	NaN	NaN	NaN
1	C2	12:37:12.219 703 416	channel0_0	0	irq_handler_exit	irq=21, ret=0	0	20	NaN	NaN	NaN	NaN
2	C2	12:37:12.219 737 468	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
3	C2	12:37:12.219 766 031	channel0_0	0	sched_switch	prev_comm=jbd2/sda5-8, prev_tid=182, prev_prio...	182	20	182.0	NaN	NaN	NaN
4	C1	12:37:12.220 362 105	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
5	C1	12:37:12.220 405 216	channel0_0	0	sched_switch	prev_comm=Xorg, prev_tid=868, prev_prio=20, pr...	868	20	868.0	NaN	NaN	NaN
6	C2	12:37:12.222 184 127	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
7	C2	12:37:12.222 191 967	channel0_0	0	sched_switch	prev_comm=kworker/0:3, prev_tid=435, prev_prio...	435	20	435.0	NaN	NaN	NaN
8	C1	12:37:12.225 181 464	channel0_0	0	irq_handler_entry	irq=20, name=vboxguest	0	20	NaN	NaN	NaN	NaN
9	C1	12:37:12.225 217 559	channel0_0	0	irq_handler_exit	irq=20, ret=1	0	20	NaN	NaN	NaN	NaN

Converting Timestamp relative to the first event

In [110]:
```
def timestamp(ts):
    time = ts.split('.')[1]
    time = time.split()
    digits = []
    for i in range(3):
        digit = [char for char in time[i]]
        for j in range(3):
            digits.append(digit[j])
    digits = [int(i) for i in digits]

    ans = 0
    mult = 1
    for i in range(9):
        ans = ans + mult*digits[9-i-1]
        mult *= 10
    return ans
df['Timestamp'] = df['Timestamp'].apply(timestamp)
initial = df['Timestamp'][0]
df['Timestamp'] = df['Timestamp'].apply(lambda sub:sub-initial)
df
```

Out[110]:

	Trace	Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source	packet_in	packet_out
0	C2	0	channel0_0	0	irq_handler_entry	irq=21, name=snd_intel8x0	0	20	NaN	NaN	NaN	NaN
1	C2	4817	channel0_0	0	irq_handler_exit	irq=21, ret=0	0	20	NaN	NaN	NaN	NaN
2	C2	38869	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
3	C2	67432	channel0_0	0	sched_switch	prev_comm=jbd2/sda5-8, prev_tid=182, prev_prio...	182	20	182.0	NaN	NaN	NaN
4	C1	663506	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
5	C1	706617	channel0_0	0	sched_switch	prev_comm=Xorg, prev_tid=868, prev_prio=20, pr...	868	20	868.0	NaN	NaN	NaN
6	C2	2485528	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN
7	C2	2493368	channel0_0	0	sched_switch	prev_comm=kworker/0:3, prev_tid=435, prev_prio...	435	20	435.0	NaN	NaN	NaN
8	C1	5482865	channel0_0	0	irq_handler_entry	irq=20, name=vboxguest	0	20	NaN	NaN	NaN	NaN
9	C1	5518960	channel0_0	0	irq_handler_exit	irq=20, ret=1	0	20	NaN	NaN	NaN	NaN

Creating previous process id column

In [111]:
```
def get_prev(content):
    ids = content.split(',')
    prev_id = ids[1].split('=')
    if prev_id[0] == ' prev_tid':
        return int(prev_id[1])
df['prev id'] = df['Contents'].apply(get_prev)
df
```

Out[111]:

	Trace	Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source	packet_in	packet_out	prev id
0	C2	0	channel0_0	0	irq_handler_entry	irq=21, name=snd_intel8x0	0	20	NaN	NaN	NaN	NaN	NaN
1	C2	4817	channel0_0	0	irq_handler_exit	irq=21, ret=0	0	20	NaN	NaN	NaN	NaN	NaN
2	C2	38869	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0
3	C2	67432	channel0_0	0	sched_switch	prev_comm=jbd2/sda5-8, prev_tid=182, prev_prio...	182	20	182.0	NaN	NaN	NaN	182.0
4	C1	663506	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0
5	C1	706617	channel0_0	0	sched_switch	prev_comm=Xorg, prev_tid=868, prev_prio=20, pr...	868	20	868.0	NaN	NaN	NaN	868.0
6	C2	2485528	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0
7	C2	2493368	channel0_0	0	sched_switch	prev_comm=kworker/0:3, prev_tid=435, prev_prio...	435	20	435.0	NaN	NaN	NaN	435.0
8	C1	5482865	channel0_0	0	irq_handler_entry	irq=20, name=vboxguest	0	20	NaN	NaN	NaN	NaN	NaN
9	C1	5518960	channel0_0	0	irq_handler_exit	irq=20, ret=1	0	20	NaN	NaN	NaN	NaN	NaN

Creating next process id column

In [112]:
```
def get_next(content):
    ids = content.split(',')
    try:
        next_id = ids[5].split('=')[1]
        return int(next_id)
    except:
        return None
df['next id'] = df['Contents'].apply(get_next)
df
```

Out[112]:

	Trace	Timestamp	Channel	CPU	Event type	Contents	TID	Prio	PID	Source	packet_in	packet_out	prev id	next id
0	C2	0	channel0_0	0	irq_handler_entry	irq=21, name=snd_intel8x0	0	20	NaN	NaN	NaN	NaN	NaN	NaN
1	C2	4817	channel0_0	0	irq_handler_exit	irq=21, ret=0	0	20	NaN	NaN	NaN	NaN	NaN	NaN
2	C2	38869	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0	182.0
3	C2	67432	channel0_0	0	sched_switch	prev_comm=jbd2/sda5-8, prev_tid=182, prev_prio...	182	20	182.0	NaN	NaN	NaN	182.0	0.0
4	C1	663506	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0	868.0
5	C1	706617	channel0_0	0	sched_switch	prev_comm=Xorg, prev_tid=868, prev_prio=20, pr...	868	20	868.0	NaN	NaN	NaN	868.0	0.0
6	C2	2485528	channel0_0	0	sched_switch	prev_comm=swapper/0, prev_tid=0, prev_prio=20,...	0	20	NaN	NaN	NaN	NaN	0.0	435.0
7	C2	2493368	channel0_0	0	sched_switch	prev_comm=kworker/0:3, prev_tid=435, prev_prio...	435	20	435.0	NaN	NaN	NaN	435.0	0.0
8	C1	5482865	channel0_0	0	irq_handler_entry	irq=20, name=vboxguest	0	20	NaN	NaN	NaN	NaN	NaN	NaN
9	C1	5518960	channel0_0	0	irq_handler_exit	irq=20, ret=1	0	20	NaN	NaN	NaN	NaN	NaN	NaN

Dropping unecessary columns

In [113]:
```
traces = df.drop(['CPU','Channel','Prio','PID','Source','Contents'],axis=1)
traces
```

Out[113]:

	Trace	Timestamp	Event type	TID	packet_in	packet_out	prev id	next id
0	C2	0	irq_handler_entry	0	NaN	NaN	NaN	NaN
1	C2	4817	irq_handler_exit	0	NaN	NaN	NaN	NaN
2	C2	38869	sched_switch	0	NaN	NaN	0.0	182.0
3	C2	67432	sched_switch	182	NaN	NaN	182.0	0.0
4	C1	663506	sched_switch	0	NaN	NaN	0.0	868.0
5	C1	706617	sched_switch	868	NaN	NaN	868.0	0.0
6	C2	2485528	sched_switch	0	NaN	NaN	0.0	435.0
7	C2	2493368	sched_switch	435	NaN	NaN	435.0	0.0
8	C1	5482865	irq_handler_entry	0	NaN	NaN	NaN	NaN
9	C1	5518960	irq_handler_exit	0	NaN	NaN	NaN	NaN

Creating link_horizontal() function

In [114]:
```
def link_horizontal(cpu,task,ts,l):

    processes = graph[cpu]

    if not task in processes:
        processes[task] = []
        processes[task].append({
            'right_label':l,
        })
    processes[task][len(processes[task])-1]['right_label'] = l

    processes[task].append({
        'val':ts,
        'right_label':None,
    })
    return len(processes[task])-1
```

Creating link_vertical() function

In [115]:
```
def link_vertical(from_cpu,to_cpu,from_process,to_process,from_vertex,to_vertex,l):

    graph[to_cpu][to_process][to_vertex]['up']={
        'cpu':from_cpu,
        'process':from_process,
        'vertex':from_vertex,
        'label':l
    }

    graph[from_cpu][from_process][from_vertex]['down']={
        'cpu':to_cpu,
        'process':to_process,
        'vertex':to_vertex,
        'label':l
    }
```

Creating current_task() function

In [116]:
```
def current_task(cpu,IRQ,e,current_tasks):
    if e=='irq_handler_entry':
        return IRQ[cpu][len(IRQ[cpu])-1]
    return current_tasks[cpu]
```

Initializing parameters

- **CPU :** The list of CPU's involved in the Experiment
- **pids :** The list of processes involved in the Experiment
- **initial_tasks :** The initial task for each CPUs in the Experiment
- **IRQ :** It holds the interrupt requests for each CPU in stack
- **PKT :** It holds the list of packets transferred during the Experiment
- **graph :** The Execution Graph
- **current_tasks :** The current task for each CPUs in the Experiment

In [117]:
```
CPU = list(traces['Trace'].unique())

pids = list(traces['TID'].unique())

initial_tasks = {}
for i in range(len(vms)) :
    for j in range(len(traces)):
        if traces.loc[j,'Trace'] == vms[i]:
            initial_tasks[vms[i]] = traces.loc[j,'TID']
            break
IRQ = {}
PKT = []

graph = {}
for i in range(len(CPU)):
    graph[CPU[i]] = {}
    IRQ[CPU[i]] = []

current_tasks = initial_tasks
```

Construction of the Graph

In [118]:
```
for i in range(len(traces)):

    e = traces.loc[i,'Event type']
    now = traces.loc[i,'Timestamp']
    cpu = traces.loc[i,'Trace']

    if e == 'sched_switch' :
        v = link_horizontal(cpu,traces.loc[i,'prev id'],now,'running')
        v = link_horizontal(cpu,traces.loc[i,'next id'],now,'preempted')
        current_tasks[cpu] = traces.loc[i,'TID']
    elif e == 'sched_ttwu' :
        target = traces.loc[i,'TID']
        source = current_task(cpu,IRQ,e,current_tasks)
        v1 = link_horizontal(cpu,target,now,'blocked')
        v2 = link_horizontal(cpu,source,now,'running')
        link_vertical(cpu,cpu,target,source,v1,v2,'wake-up')
    elif e == 'irq_handler_entry':
        IRQ[cpu].append(traces.loc[i,'TID'])
        v = link_horizontal(cpu,current_task(cpu,IRQ,e,current_tasks),now,'None')
    elif e == 'irq_handler_exit':
        IRQ[cpu].pop()
    elif e == ' inet_sock_local_out':
        tx = link_horizontal(cpu,current_task(cpu,IRQ,e,current_tasks),now,'running')
        PKT.append((traces.loc[i,'packet_out'],tx))
    elif e == ' inet_sock_local_in':
        tx = None
        for j in len(PKT):
            if PKT[j].index(0) == traces.loc[i,'packet_in']:
                (packet,tx) = PKT.pop(j)
                break
        rx = link_horizontal(cpu,current_task(cpu,IRQ,e,current_tasks),now,'running')
        link_vertical(cpu,cpu,traces.loc[i,'TID'],traces.loc[i,'TID'],tx,rx,'network')

graph
```

Out[118]:
```
{'C2': {0: [{'right_label': 'None'},
   {'val': 0, 'right_label': 'running'},
   {'val': 38869, 'right_label': 'preempted'},
   {'val': 67432, 'right_label': 'running'},
   {'val': 2485528, 'right_label': 'preempted'},
   {'val': 2493368, 'right_label': None}],
  182.0: [{'right_label': 'preempted'},
   {'val': 38869, 'right_label': 'running'},
   {'val': 67432, 'right_label': None}],
  435.0: [{'right_label': 'preempted'},
   {'val': 2485528, 'right_label': 'running'},
   {'val': 2493368, 'right_label': None}]},
 'C1': {0.0: [{'right_label': 'running'},
   {'val': 663506, 'right_label': 'preempted'},
   {'val': 706617, 'right_label': 'None'},
   {'val': 5482865, 'right_label': None}],
  868.0: [{'right_label': 'preempted'},
   {'val': 663506, 'right_label': 'running'},
   {'val': 706617, 'right_label': None}]}}
```

Printing the Execution Graph

In [151]:
```
for CPU in graph:
    print(CPU)
    processes = graph[CPU]
    #print("|",end =" ")
    for pid in processes:
        print("|--Process",int(pid),end =" ")
        nodes = processes[pid]
        print("-->",end =" ")
        for ts in range(1,len(nodes)):
            if not nodes[ts]['right_label'] == None:
                print(nodes[ts]['val'],"--",nodes[ts]['right_label'],"-->",end =" ")
        print()
```

```
C2
|--Process 0 --> 0 -- running --> 38869 -- preempted --> 67432 -- running --> 2485528 -- preempted -->
|--Process 182 --> 38869 -- running -->
|--Process 435 --> 2485528 -- running -->
C1
|--Process 0 --> 663506 -- preempted --> 706617 -- None -->
|--Process 868 --> 663506 -- running -->
```

In [ ]:

In [ ]: