Rough Estimation of Instruction Rate

Anshul Gupta

28/06/2014

Motive

Calculate the instruction rate of execution using TIC traces.

Overview

5 traces with TIC instrumentation of LU benchmark, class C and 16 processes were acquired on graphene cluster.
Their **COMP 4b** and **COMP 4c** global average values and involved processes were parsed using the *scripts/Expand.java* code with this format.
See *org/analysis.org*

```
setwd('..')
library(ggplot2)
dataset.tic.lu.C.16.comp.4b.I<-read.table("datasets/tic.lu.C.16.comp.4b.I", header=TRUE)
dataset.tic.lu.C.16.comp.4b.II<-read.table("datasets/tic.lu.C.16.comp.4b.II", header=TRUE)
dataset.tic.lu.C.16.comp.4b.III<-read.table("datasets/tic.lu.C.16.comp.4b.III", header=TRUE)
dataset.tic.lu.C.16.comp.4b.IV<-read.table("datasets/tic.lu.C.16.comp.4b.IV", header=TRUE)
dataset.tic.lu.C.16.comp.4b.V<-read.table("datasets/tic.lu.C.16.comp.4b.V", header=TRUE)
dataset.tic.lu.C.16.comp.4c.I<-read.table("datasets/tic.lu.C.16.comp.4c.I", header=TRUE)
dataset.tic.lu.C.16.comp.4c.II<-read.table("datasets/tic.lu.C.16.comp.4c.II", header=TRUE)
dataset.tic.lu.C.16.comp.4c.III<-read.table("datasets/tic.lu.C.16.comp.4c.III", header=TRUE)
dataset.tic.lu.C.16.comp.4c.IV<-read.table("datasets/tic.lu.C.16.comp.4c.IV", header=TRUE)
dataset.tic.lu.C.16.comp.4c.V<-read.table("datasets/tic.lu.C.16.comp.4c.V", header=TRUE)
```

The variables are named as `d.<A>.<B>.4b.4c.<X>`,
where `<X>` is the TIC dataset,
`<A>` is the class,
and `<B>` is the number of processes.
Each `d.<A>.<B>.4b.4c.<X>` has 6 fields:

- eventID = Event ID
- opcode = Opcode of the event
- subblock = Subblock ID of the event
- events = Number of elements involved in the event
- COMP.4b.avg = Average of COMP 4b
- COMP.4c.avg = Average of COMP 4c

```
d.C.16.4b.4c.I<-cbind(dataset.tic.lu.C.16.comp.4b.I[,1:4], dataset.tic.lu.C.16.comp.4b.I$avg/1e6, datase
d.C.16.4b.4c.II<-cbind(dataset.tic.lu.C.16.comp.4b.II[,1:4], dataset.tic.lu.C.16.comp.4b.II$avg/1e6, dat
d.C.16.4b.4c.III<-cbind(dataset.tic.lu.C.16.comp.4b.III[,1:4], dataset.tic.lu.C.16.comp.4b.III$avg/1e6,
d.C.16.4b.4c.IV<-cbind(dataset.tic.lu.C.16.comp.4b.IV[,1:4], dataset.tic.lu.C.16.comp.4b.IV$avg/1e6, dat
d.C.16.4b.4c.V<-cbind(dataset.tic.lu.C.16.comp.4b.V[,1:4], dataset.tic.lu.C.16.comp.4b.V$avg/1e6, datase

colnames(d.C.16.4b.4c.I)[5:6]<-c("COMP.4b.avg", "COMP.4c.avg")
colnames(d.C.16.4b.4c.II)[5:6]<-c("COMP.4b.avg", "COMP.4c.avg")
colnames(d.C.16.4b.4c.III)[5:6]<-c("COMP.4b.avg", "COMP.4c.avg")
colnames(d.C.16.4b.4c.IV)[5:6]<-c("COMP.4b.avg", "COMP.4c.avg")
colnames(d.C.16.4b.4c.V)[5:6]<-c("COMP.4b.avg", "COMP.4c.avg")
```

Get the average instruction rate

Having COMP 4b averages and COMP 4c averages, instruction rate is calculated by:
`instruction rate = mean(COMP.4c.average / COMP.4b.average)`

```
inst_rate<-c(mean(d.C.16.4b.4c.I$COMP.4c.avg/d.C.16.4b.4c.I$COMP.4b.avg),
     mean(d.C.16.4b.4c.II$COMP.4c.avg/d.C.16.4b.4c.II$COMP.4b.avg),
     mean(d.C.16.4b.4c.III$COMP.4c.avg/d.C.16.4b.4c.III$COMP.4b.avg),
     mean(d.C.16.4b.4c.IV$COMP.4c.avg/d.C.16.4b.4c.IV$COMP.4b.avg),
     mean(d.C.16.4b.4c.V$COMP.4c.avg/d.C.16.4b.4c.V$COMP.4b.avg))

names(inst_rate)<-c('dataset1','dataset2','dataset3','dataset4','dataset5')
print(inst_rate)
```

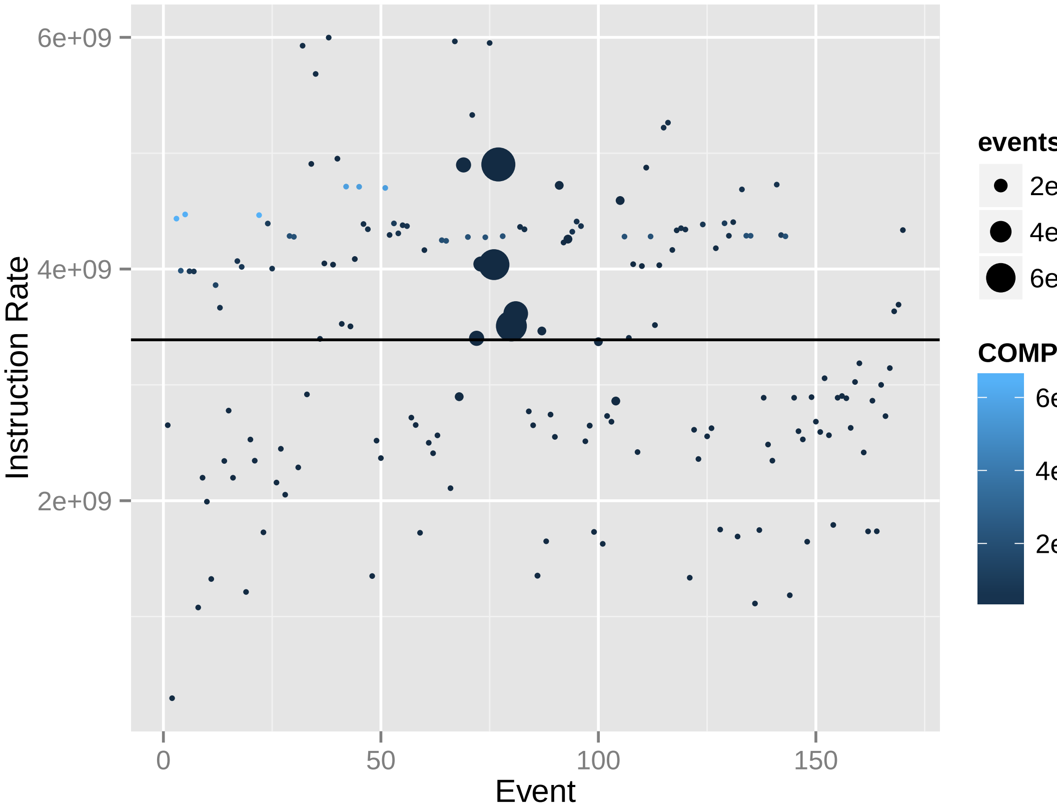
```
##  dataset1  dataset2  dataset3  dataset4  dataset5
## 3.389e+09 3.416e+09 3.414e+09 3.419e+09 3.408e+09
```

Plots

On x-axis, we have the eventID_subblock.
On y-axis, we have the Instruction Rate (COMP.4c.avg/COMP.4b.avg).
The size of each point represents the number of elements involved in the current event_subblock.
Color represents the average number of instructions (COMP.4c.avg).

Dataset 1

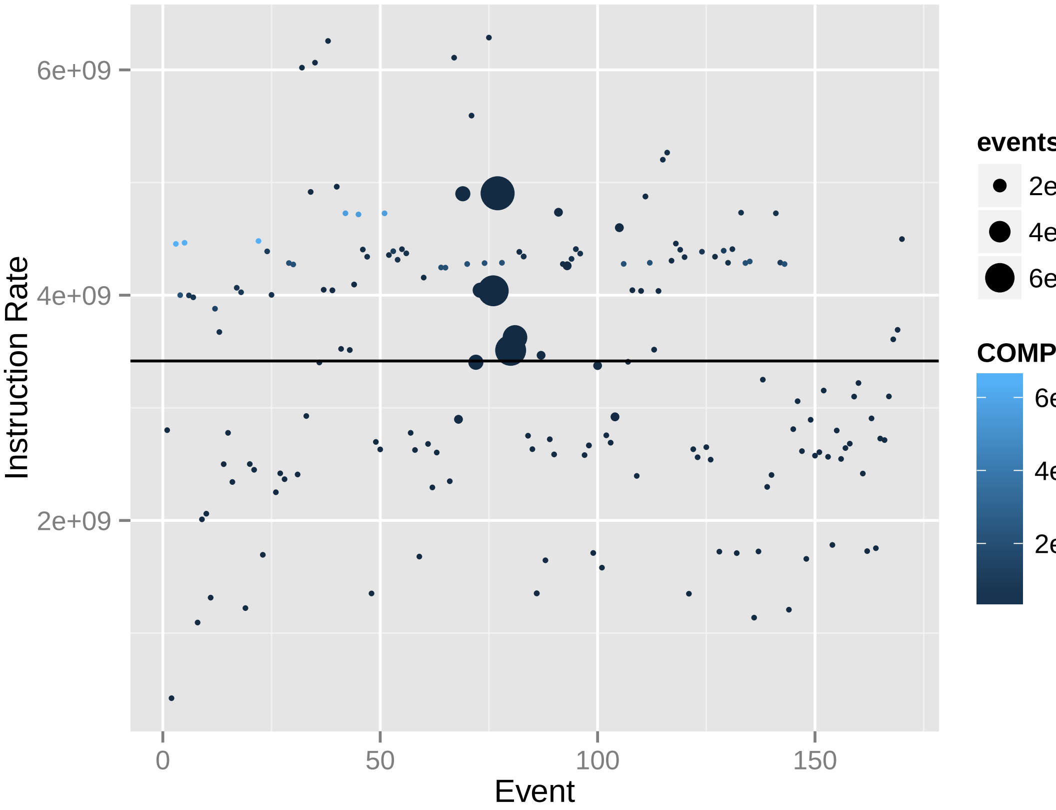
```
ggplot(d.C.16.4b.4c.I, aes(x=1:nrow(d.C.16.4b.4c.I),
                           y=COMP.4c.avg/COMP.4b.avg,
                           color=COMP.4c.avg,
                           size=events)) +
  geom_point() + xlab("Event") + ylab("Instruction Rate") +
  geom_hline(yintercept=inst_rate[1])
```



vs Instruction Rate Dataset 1.pdf

Dataset 2

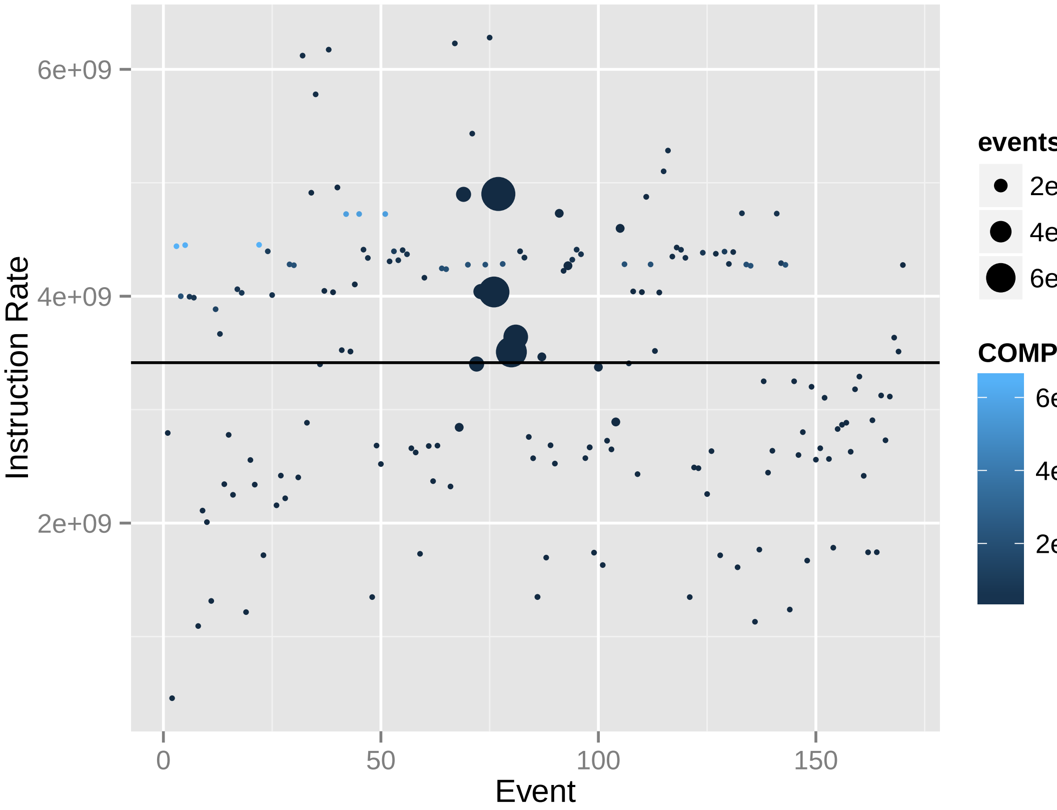
```
ggplot(d.C.16.4b.4c.II, aes(x=1:nrow(d.C.16.4b.4c.II),
                           y=COMP.4c.avg/COMP.4b.avg,
                           color=COMP.4c.avg,
                           size=events)) +
  geom_point() + xlab("Event") + ylab("Instruction Rate") +
  geom_hline(yintercept=inst_rate[2])
```



vs Instruction Rate Dataset 2.pdf

Dataset 3

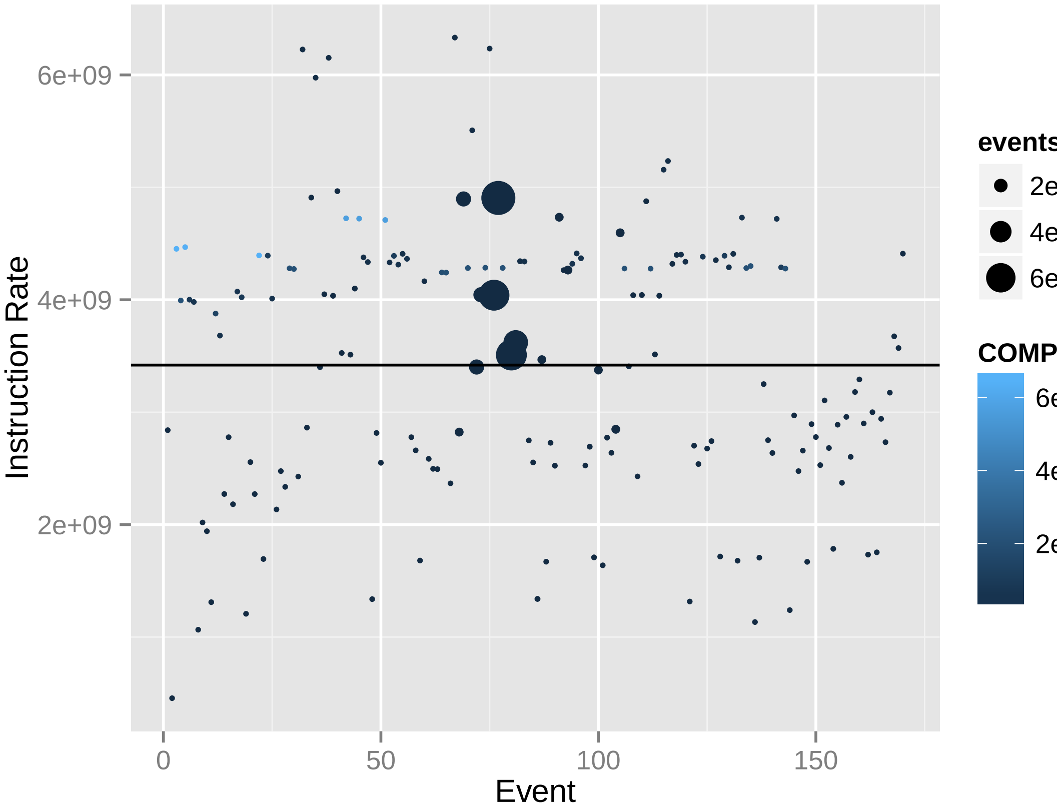
```
ggplot(d.C.16.4b.4c.III, aes(x=1:nrow(d.C.16.4b.4c.III),
                            y=COMP.4c.avg/COMP.4b.avg,
                            color=COMP.4c.avg,
                            size=events)) +
  geom_point() + xlab("Event") + ylab("Instruction Rate") +
  geom_hline(yintercept=inst_rate[3])
```



vs Instruction Rate Dataset 3.pdf

Dataset 4

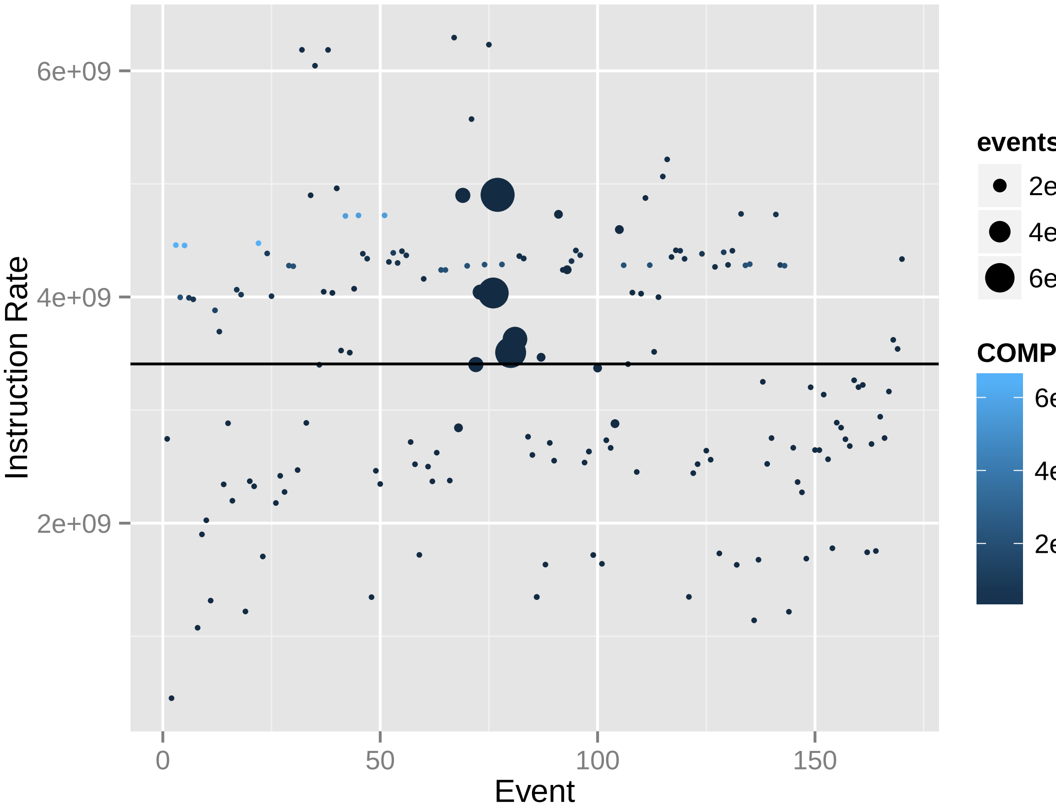
```
ggplot(d.C.16.4b.4c.IV, aes(x=1:nrow(d.C.16.4b.4c.IV),
                            y=COMP.4c.avg/COMP.4b.avg,
                            color=COMP.4c.avg,
                            size=events)) +
  geom_point() + xlab("Event") + ylab("Instruction Rate") +
  geom_hline(yintercept=inst_rate[4])
```



vs Instruction Rate Dataset 4.pdf

Dataset 5

```
ggplot(d.C.16.4b.4c.V, aes(x=1:nrow(d.C.16.4b.4c.V),
                           y=COMP.4c.avg/COMP.4b.avg,
                           color=COMP.4c.avg,
                           size=events)) +
  geom_point() + xlab("Event") + ylab("Instruction Rate") +
  geom_hline(yintercept=inst_rate[5])
```



vs Instruction Rate Dataset 5.pdf

Results

Instruction rate is around **3.4e9 Instructions per sec**

Conclusions

It is clear from the plots that the executions are quite stable as CPU bursts on each execution occur at same places.