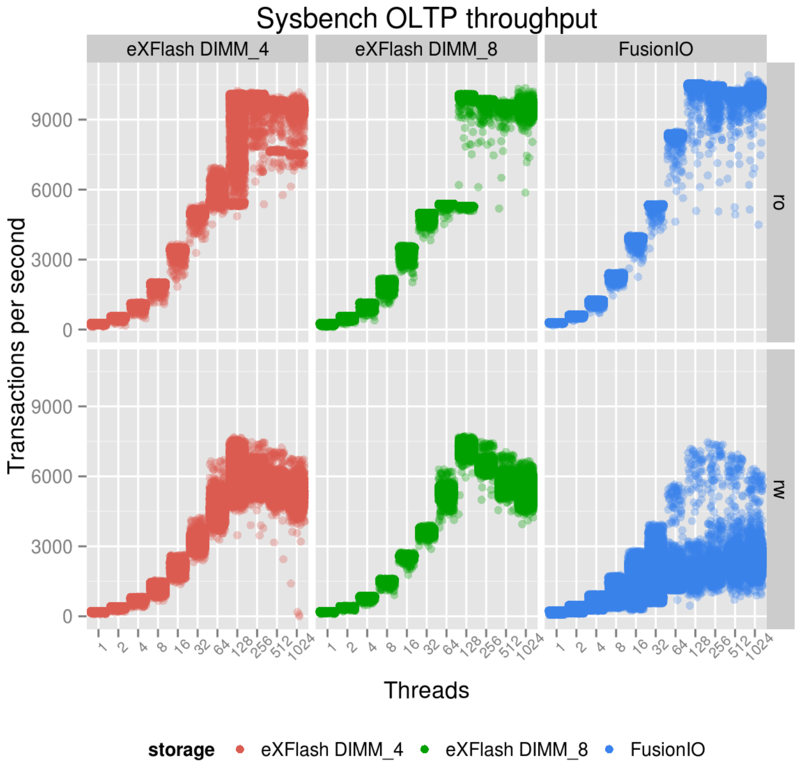
PERCONA

Visualizing benchmark data with R and ggplot2

Peter Boros
Principal Architect @ Percona

FOSDEM 2015

In this talk...



How to make this graph

In 17 easy steps

Step 0: preprocessing the data

```
1,eXFlash DIMM_4,rw,32,sysbench_tps,2717.27
1,eXFlash DIMM_4,rw,32,sysbench_resp_time,15.86
2,eXFlash DIMM_4,rw,32,sysbench_tps,3104.05
2,eXFlash DIMM_4,rw,32,sysbench_resp_time,13.69
3,eXFlash DIMM_4,rw,32,sysbench_tps,3221.99

colnames(sysbench_oltp)<-
c("time","storage","ro_rw","threads","metric","value")
```

Something should transform raw data to this.

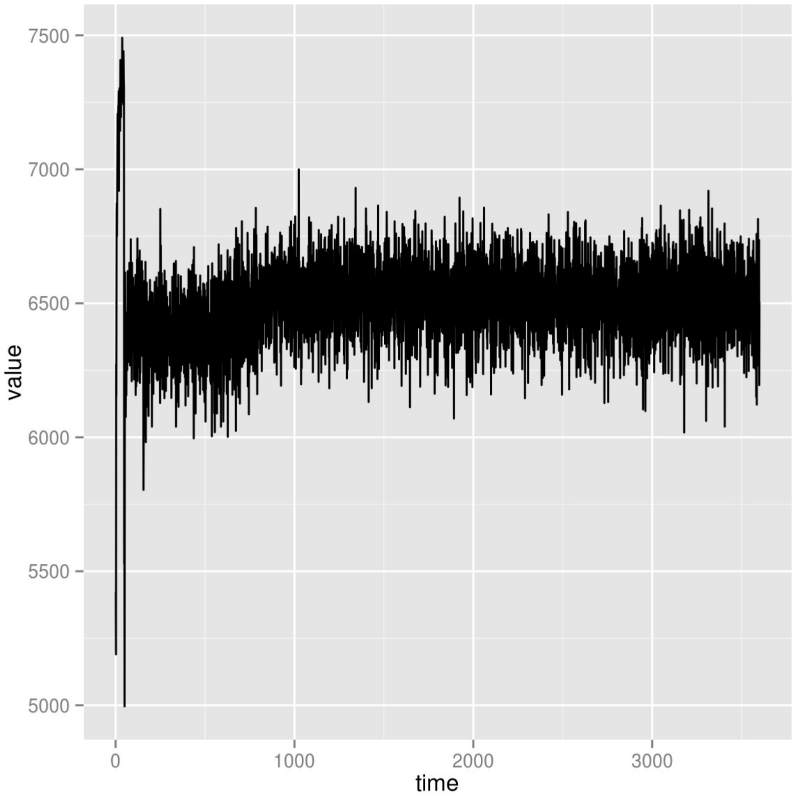
Step 0: reading and subsetting

```
sysbench_oltp<-
read.table("r_ggplot2_benchmark_visualization/sysbench_simp
le.txt",sep=",",as.is=T,header=F)

colnames(sysbench_oltp)<-
c("time","storage","ro_rw","threads","metric","value")

sysbench_tps<-subset(sysbench_oltp,metric=="sysbench_tps" &
threads=="256" & ro_rw=="rw" & storage=='eXFlash DIMM_8')
sysbench_tps$value<-as.numeric(sysbench_tps$value)
sysbench_tps_summ<-
ddply(sysbench_tps,c("storage","threads","ro_rw"),
                         summarize,sd_throughput=sd(value),
                         mean_throughput=mean(value),

t95th_percentile_throughput=quantile(value, 0.95),
                         max_throughput=max(value))
```

Step 0: reading and subsetting

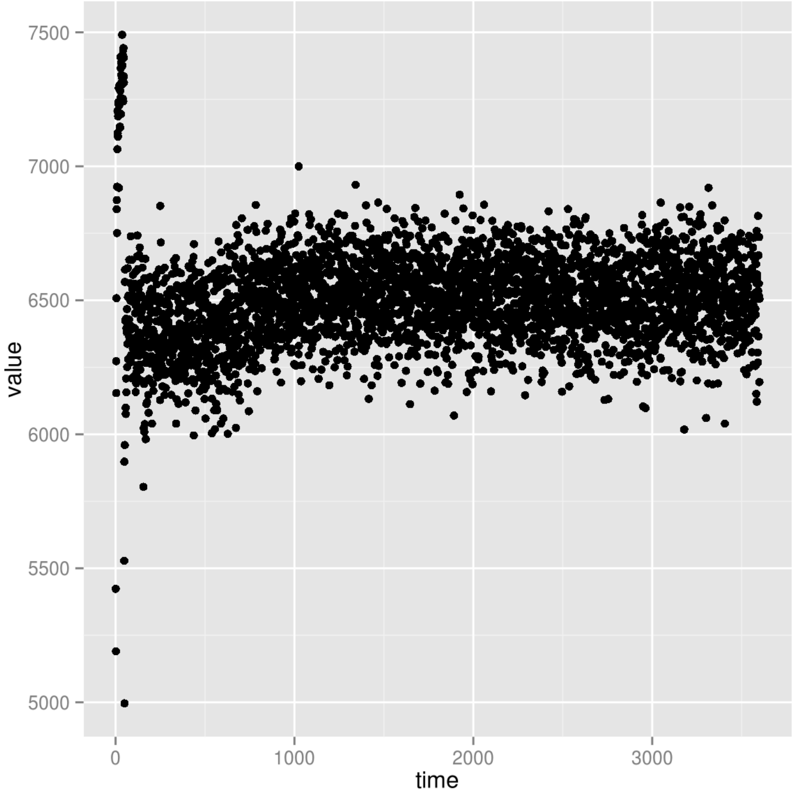
```
sysbench_oltp<-
read.table("r_ggplot2_benchmark_visualization/sysbench_simp
le.txt",sep=",",as.is=T,header=F)

colnames(sysbench_oltp)<-
c("time","storage","ro_rw","threads","metric","value")

sysbench_tps<-subset(sysbench_oltp,metric=="sysbench_tps" &
threads=="256" & ro_rw=="rw" & storage=='eXFlash DIMM_8')
sysbench_tps$value<-as.numeric(sysbench_tps$value)
sysbench_tps_summ<-
ddply(sysbench_tps,c("storage","threads","ro_rw"),
      summarize,sd_throughput=sd(value),
      mean_throughput=mean(value),
      t95th_percentile_throughput=quantile(value, 0.95),
      max_throughput=max(value))
```

Step 1: Initial graph from a single run



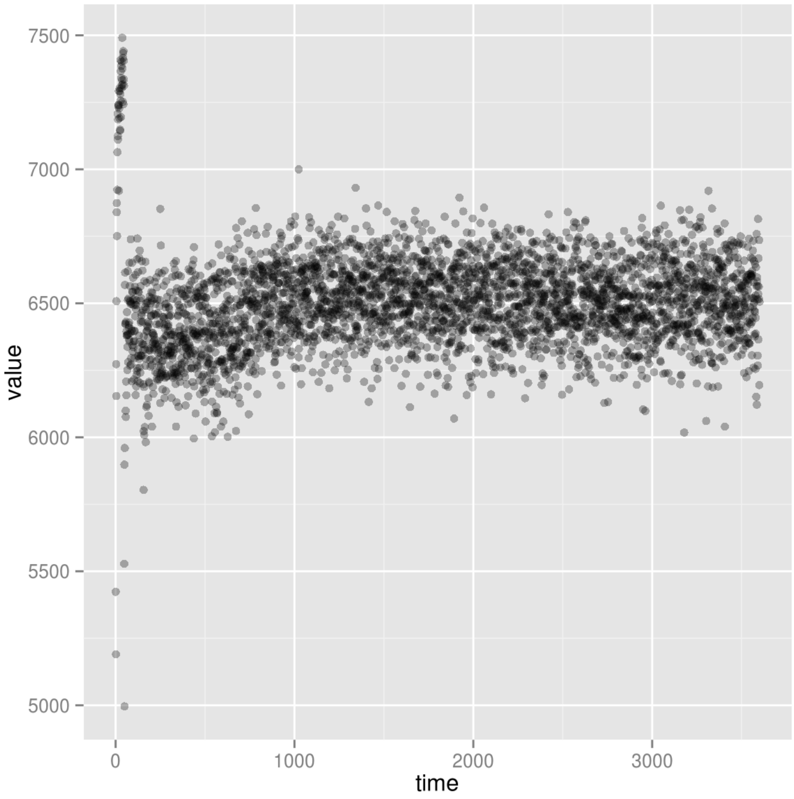
```
tps_graph<-
ggplot(sysbench_tps)

tps_graph<-
tps_graph+aes(x=time,y=val
ue)

tps_graph<-
tps_graph+geom_line()

tps_graph
```

Step 2: Use jitter plot



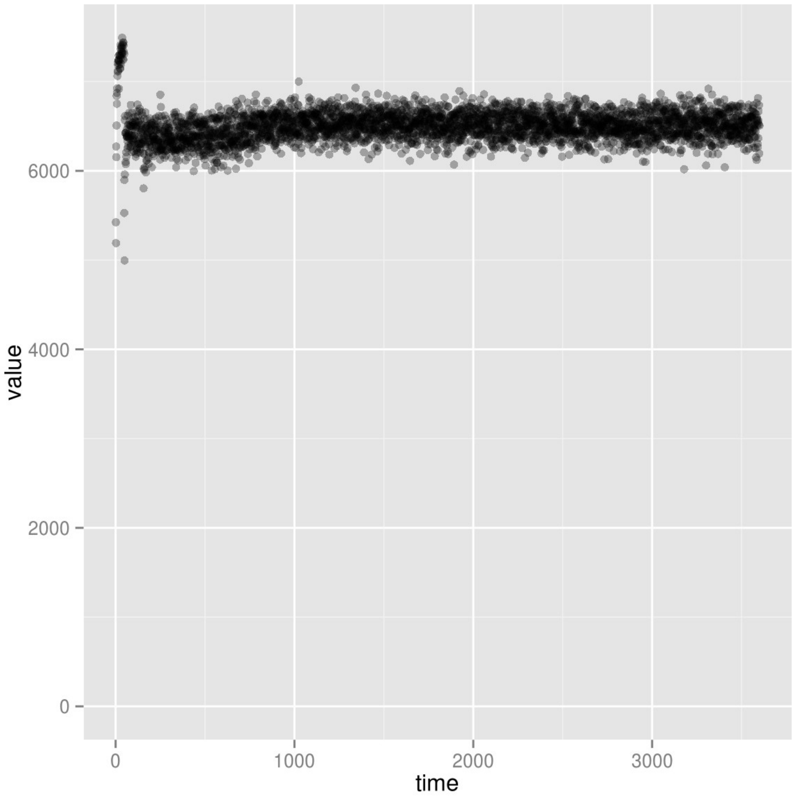
```
tps_graph<-
ggplot(sysbench_tps)

tps_graph<-
tps_graph+aes(x=time,y=val
ue)

tps_graph<-
tps_graph+geom_jitter()

tps_graph
```

Step 3: Use transparent dots



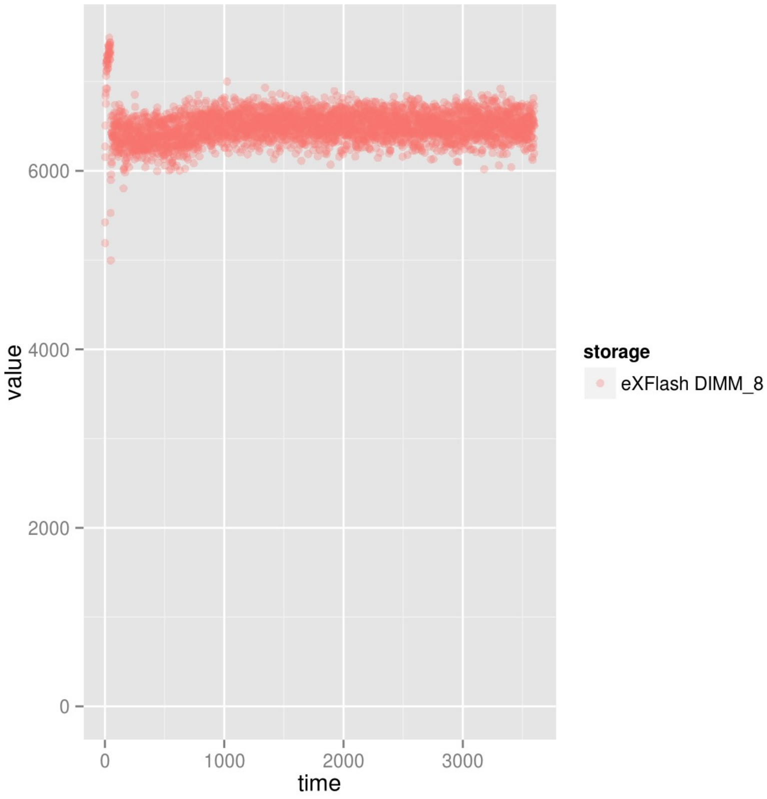
```
tps_graph<-
ggplot(sysbench_tps)

tps_graph<-
tps_graph+aes(x=time,y=val
ue)

tps_graph<-
tps_graph+geom_jitter(alph
a=0.3)

tps_graph
```

Step 4: Start the y axis from 0



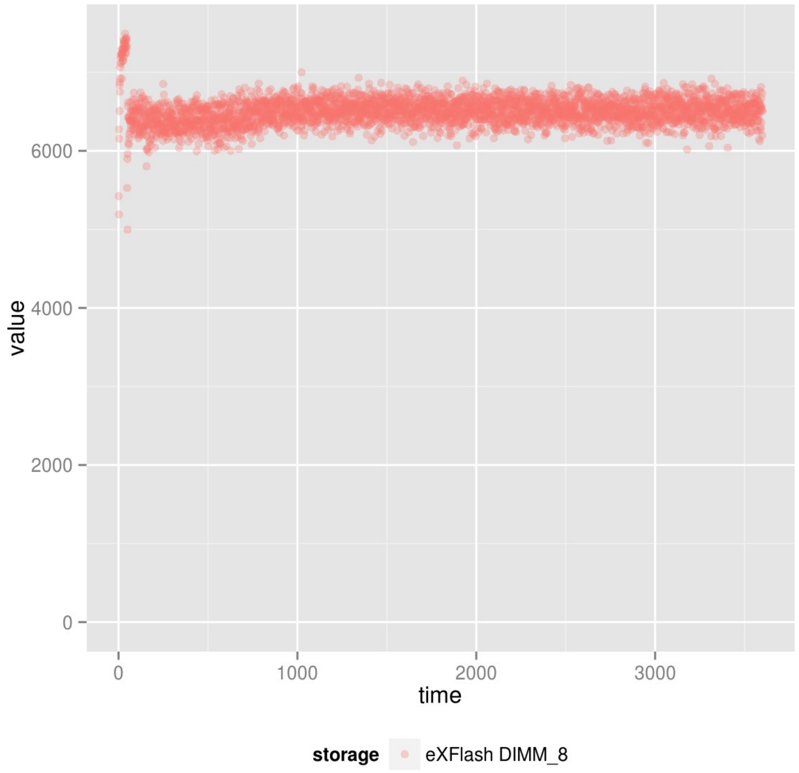
```
tps_graph<-ggplot(sysbench_tps)

tps_graph<-tps_graph+aes(x=time,y=value)

tps_graph<-tps_graph+geom_jitter(alpha=0.3)

tps_graph<-tps_graph+expand_limits(y=0)

tps_graph
```

Step 5: Specify geom and color



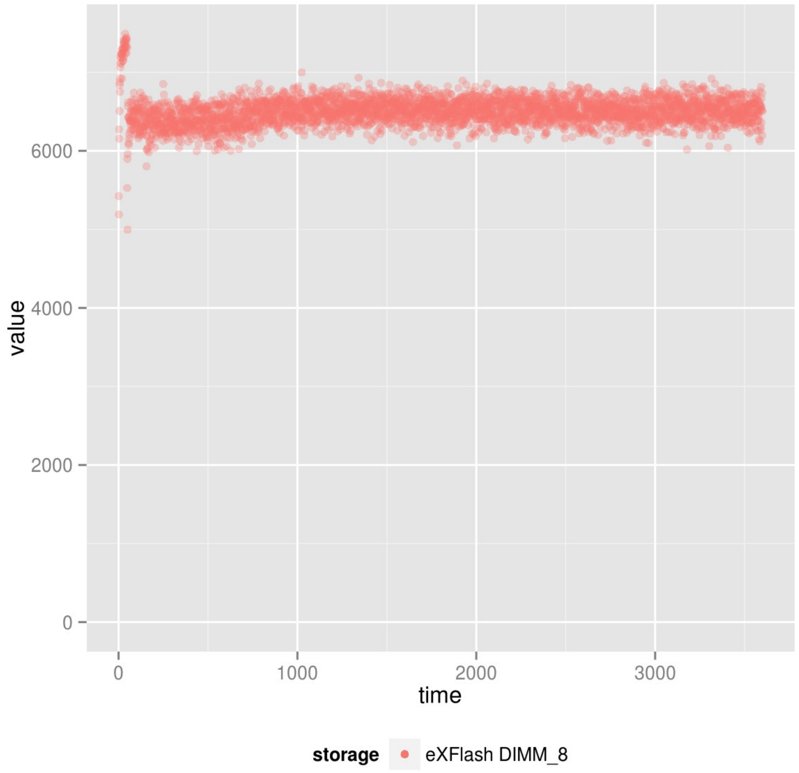
```
tps_graph<-tps_graph
+aes(x=time,y=value,
geom=storage,
color=storage)
```

Step 6: Put the legend to the bottom



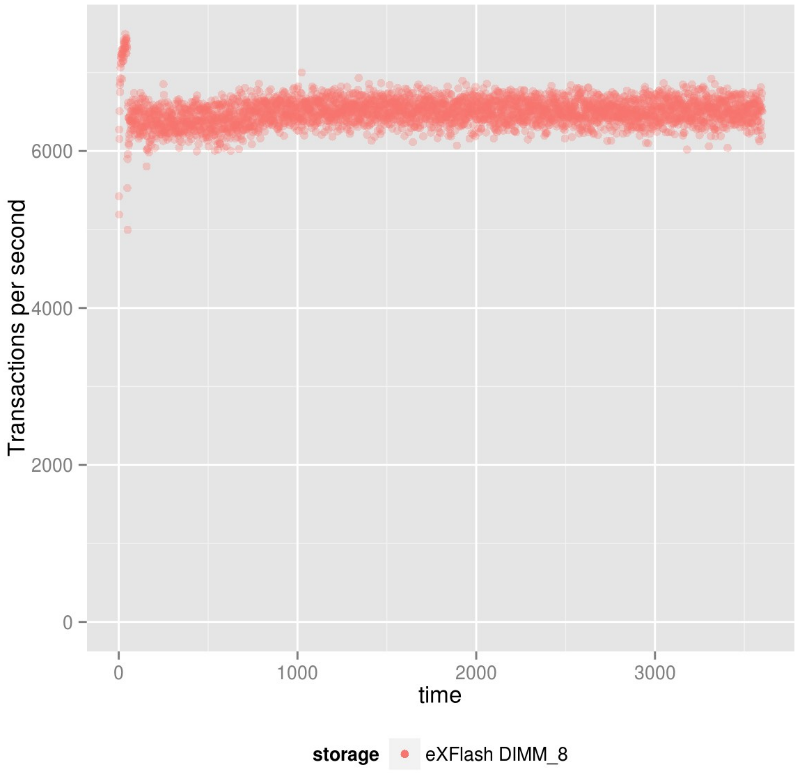
```
tps_graph<-
tps_graph
+theme(legend.position="bottom")
```

Step 7: No transparency in the legend



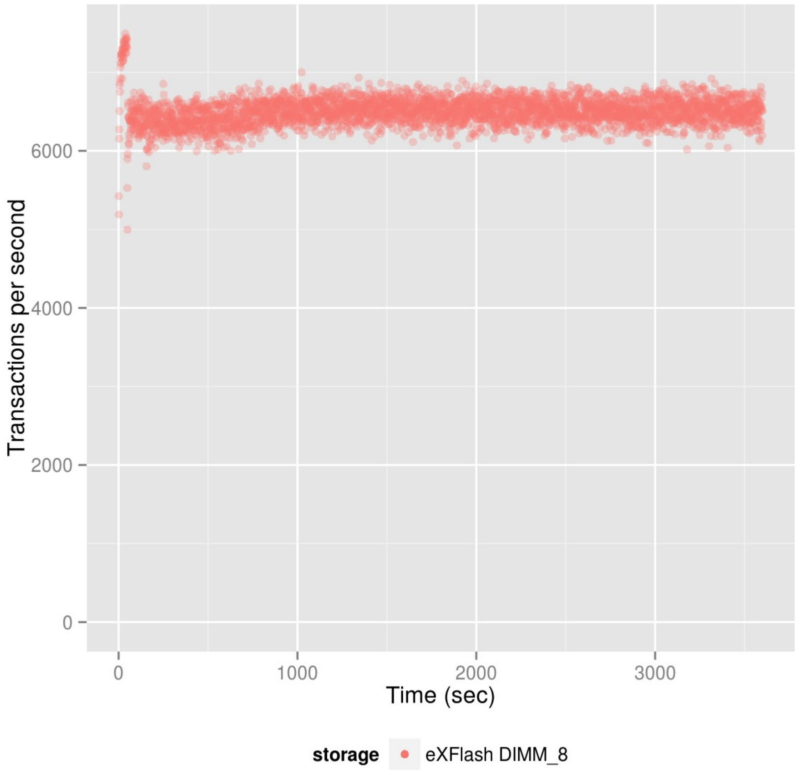
```
tps_graph<-
tps_graph
+guides(colour=
guide_legend

(override.aes=

list(alpha=1,
fill=NA)))
```

Step 8: Setting label of the y axis



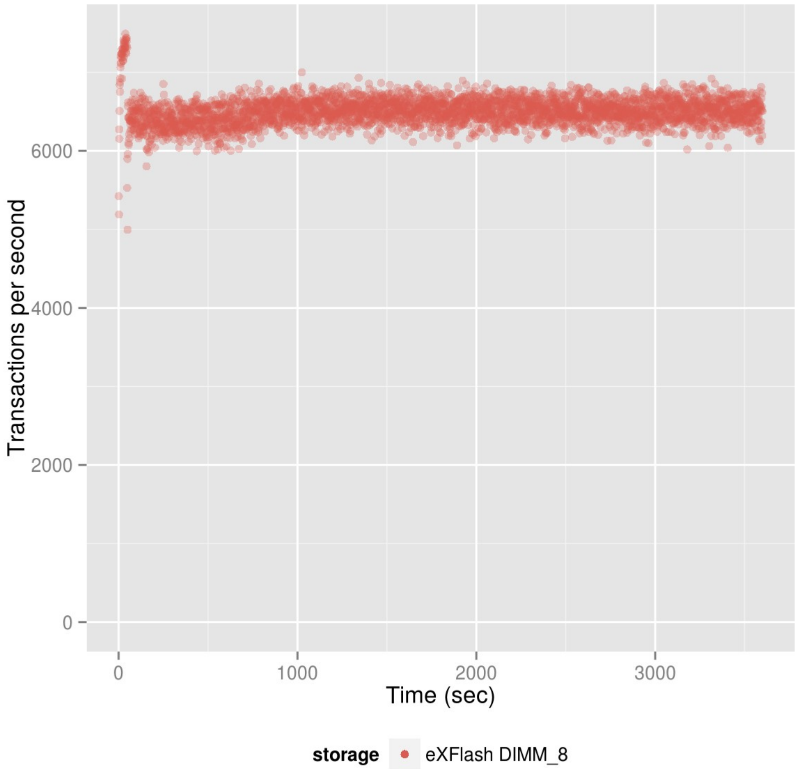
```
tps_graph<-
tps_graph
+ylab("Transact
ions per
second")
```

Step 9: Setting label of the x axis



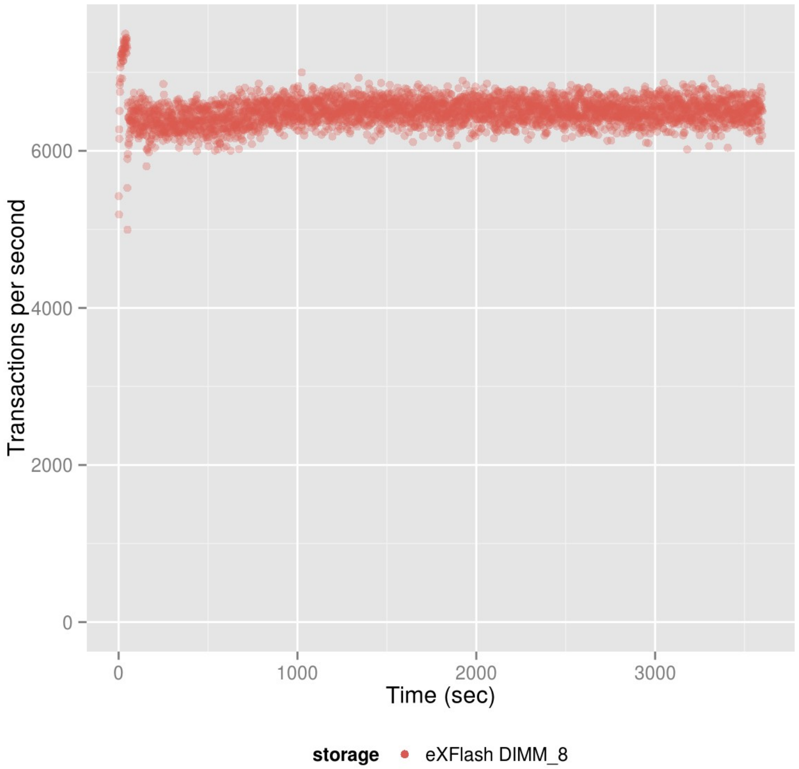
```
tps_graph<-
tps_graph
+xlab("Time
(sec)")
```

Step 10: More vibrant colors



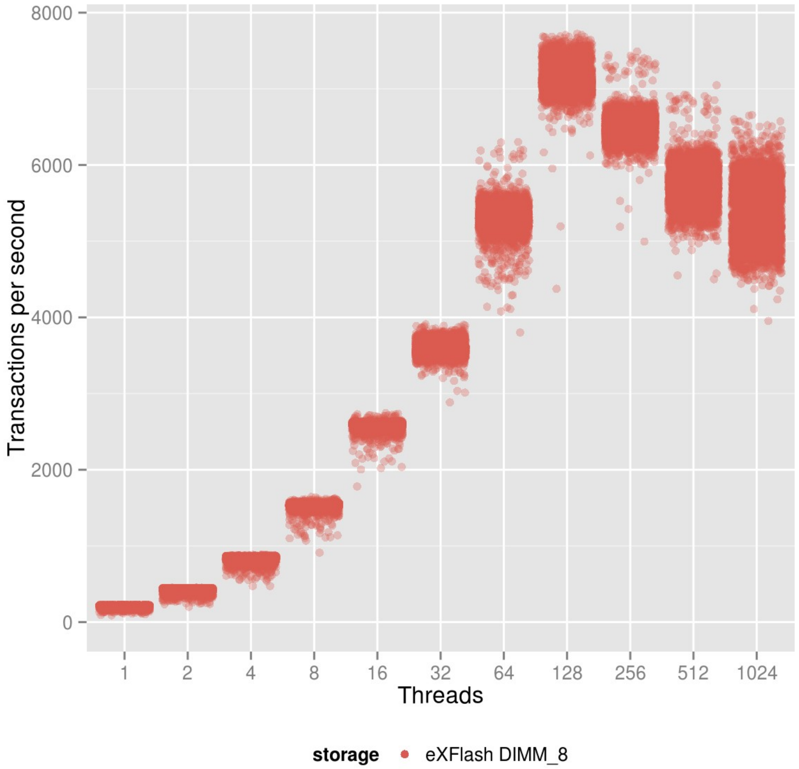
```
tps_graph<-
tps_graph
+scale_color_hue
(l=55,name="stor
age")
```

Step 11: Legend's background white



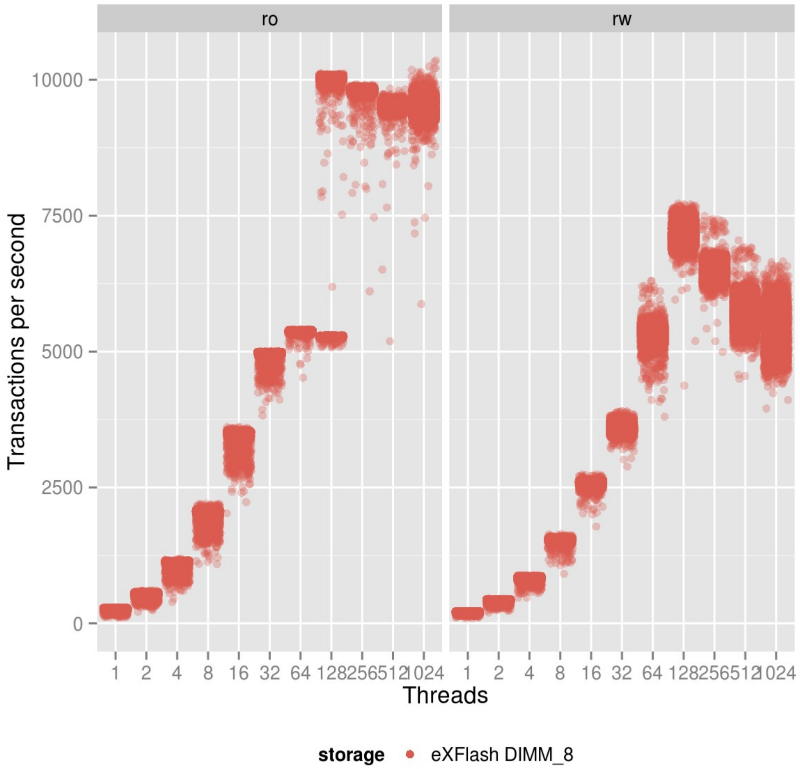
```
tps_graph<-
tps_graph
+theme(
legend.key=
element_rect(
fill="white"))
```

Step 12: Change aesthetics of the x axis



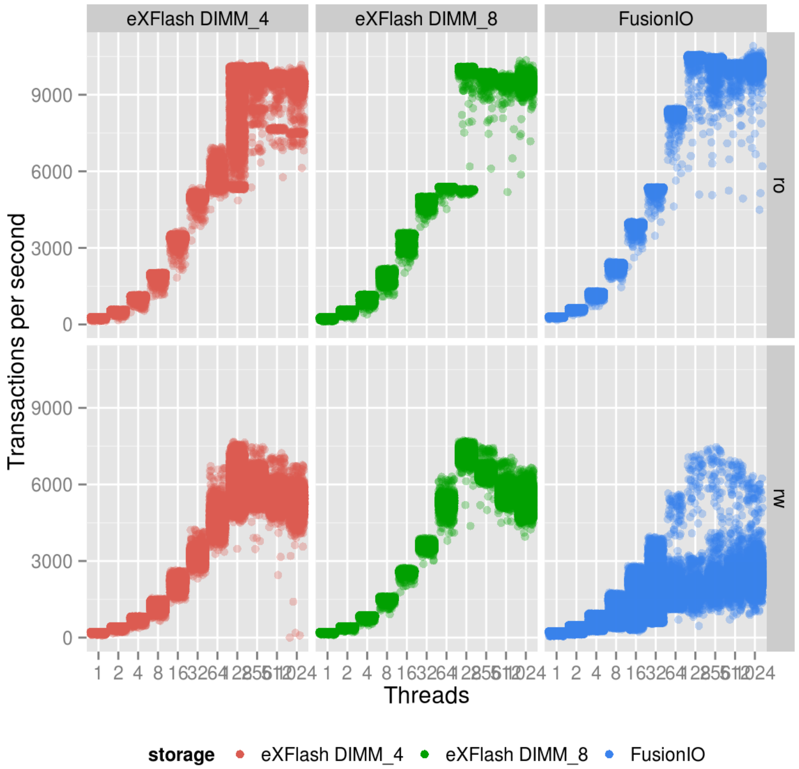
- tps_graph<- tps_graph
- +aes(
- **x=factor(threads),**
- y=value,
- geom=storage,
- colour=storage)

Step 13: Faceting, side by side comparison



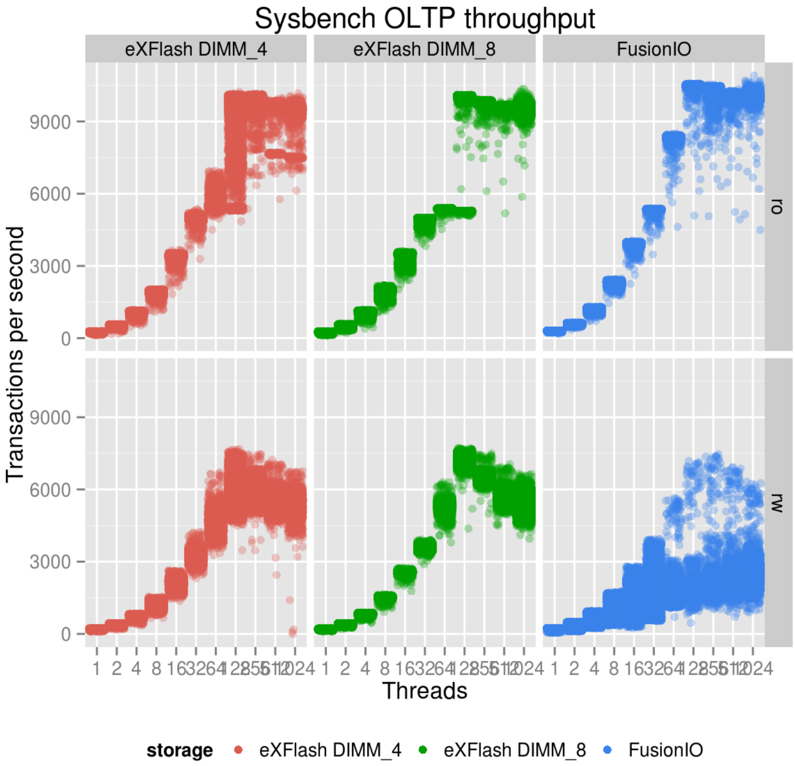
```
tps_graph<-
tps_graph
+facet_grid(
. ~ ro_rw)
```

Step 14: More faceting



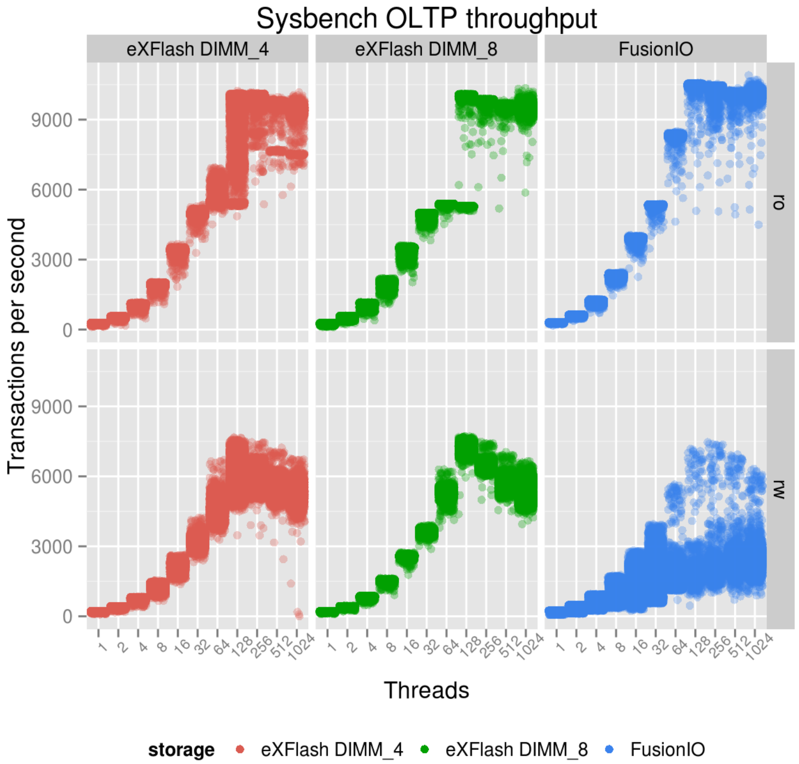
```
tps_graph<-
tps_graph
+facet_grid(
ro_rw ~
storage)
```

Step 15: Set the graph title



```
tps_graph<-
tps_graph
+ggtitle(
"Sysbench OLTP
throughput")
```

Step 16: Fix x axis text



```
tps_graph<-
tps_graph
+theme(
axis.text.x =
element_text(
size =
rel(0.8),
angle = 45))
```

Saving the plot to file

```
ggsave(plot = tps_graph,
"r_ggplot2_benchmark_visualizatio
n/ex16.png",
dpi=200, scale=1, height=6,
width=6, type = "cairo-png")
```

All the R on one slide

```
tps_graph<-ggplot(sysbench_tps)
tps_graph<-
tps_graph+aes(x=factor(threads),y=value,geom=storage,colour=storage)
tps_graph<-tps_graph+facet_grid(ro_rw ~ storage)
tps_graph<-tps_graph+geom_jitter(alpha=0.3)
tps_graph<-tps_graph+expand_limits(y=0)
tps_graph<-tps_graph+theme(legend.position="bottom")
tps_graph<-tps_graph+xlab("Threads")
tps_graph<-tps_graph+ylab("Transactions per second")
tps_graph<-tps_graph+ggtitle("Sysbench OLTP throughput")
tps_graph<-tps_graph+theme(legend.key=element_rect(fill="white"))
tps_graph<-tps_graph+guides(colour=guide_legend(override.aes=list(alpha=1,
fill=NA)))
tps_graph<-tps_graph+scale_color_hue(l=55,name="storage")
tps_graph<-tps_graph+theme(axis.text.x = element_text(size = rel(0.8), angle
= 45))
tps_graph

ggsave(plot = tps_graph, "r_ggplot2_benchmark_visualization/ex16.png",
dpi=200, scale=1, height=6, width=6, type = "cairo-png")
```

All the examples and sample data on github

Thanks!