Exploration of HDFS performance without stressing the system

Louis-Claude Canon
April 6, 2012

To avoid previous problem, a sleep of 3 seconds was introduced between each operation, so that the system can clean all blocks that need to be deleted. Additionally, the block is a multiple of 4 MiB up to 24 times this amount (96 MiB). All the other parameters have default values, except the replication which is set to 1 by the client. No failure was detected.

1 Preliminary analysis

```r
> D <- read.table("explo2_summary")
> dim(D)
[1] 530 13

> names(D) <- c("servers", "clients", "strip_size", "file_size",
+ "consecutive", "MPI_rank", "file_size", "consecutive", "write_start",
+ "write_end", "write_close", "read_start", "read_end")
> duration <- cbind(D[, c("strip_size", "file_size")], write = D[, "write_close"] - D[, "write_start"], read = D[, "read_end"] -
+ D[, "read_start"])
> pairs(~log10(strip_size) + log10(file_size) + log10(write) +
+ log10(read), data = duration, cex = 0.1)
```
2 Write operation

```r
> plot(duration[, "file_size"], duration[, "write"], log = "xy",
+       cex = 0.1)
> duration_small <- duration[duration[, "strip_size"] == 2^22,
+ ]
> duration_large <- duration[duration[, "strip_size"] >= 2^26,
+ ]
> dim(duration_small)
[1] 113  4
> dim(duration_large)
[1]  68  4

> points(write ~ file_size, data = duration_small, cex = 0.3, col = 2)
> points(write ~ file_size, data = duration_large, cex = 0.3, col = 3)
> (latency <- median(duration[duration[, "file_size"] < 1e+05,
+       "write")]))
```
The block size has a clear impact on the performance. Let’s fit the data depending on the number of blocks (the assumption is that a latency is paid for each block).

```r
> fit <- lm(write ~ file_size * strip_size * I(1/file_size) * I(1/strip_size),
+   data = duration)
> library(xtable)
> xtable(summary(fit), display = c("s", "g", "f", "f", "g"))
```

Three effects seems to be significant: file size (obviously), number of blocks (as expected) and, to a lesser extend, the product of the sizes (unexplained). We redo the fit with only the two effects we are interesting in.

```r
> fit <- lm(write ~ file_size + I(file_size/strip_size), data = duration)
> xtable(summary(fit), display = c("s", "g", "f", "f", "g"))
```

```r
> 1/coefficients(fit)[2]
```

file_size
77268210
The model is $77268210 \times \text{file\_size} + 0.04012167 \times \frac{\text{file\_size}}{\text{strip\_size}}$, which means that the latency paid by each block is around 40 ms, which is not far from the minimum time for each transfer (around 30 ms). A uniform sampling of the number of block with a maximum block size close to 1 GB could improve this model. Also, the maximum bandwidth seems to be close to 80 MB/s.

### Read operation

```r
> plot(duration[, "file_size"], duration[, "read"], log = "xy",
+ cex = 0.1)
> points(read ~ file_size, data = duration_small, cex = 0.3, col = 2)
> points(read ~ file_size, data = duration_large, cex = 0.3, col = 3)
> (latency <- median(duration[duration[, "file_size"] < 1e+05, + "read"]))
[1] 0.0030551
```
> fit <- lm(read ~ file_size * strip_size * I(1/file_size) * I(1/strip_size),
+     data = duration)
> xtable(summary(fit), display = c("s", "g", "f", "f", "g"))

|                           | Estimate | Std. Error | t value | Pr(>|t|) |
|---------------------------|----------|------------|---------|----------|
| (Intercept)               | 0.004798 | 0.0070     | 0.68    | 0.4948   |
| file_size                 | 9.957e-09| 0.0000     | 221.20  | 0        |
| strip_size                | 1.601e-10| 0.0000     | 1.32    | 0.189    |
| I(1/file_size)            | 0.0008351| 0.0356     | 0.02    | 0.9813   |
| I(1/strip_size)           | 3.06e+04 | 39113.0191 | 0.78    | 0.4343   |
| file_size:strip_size      | 4.138e-18| 0.0000     | 5.84    | 9.414e-09|
| strip_size:I(1/file_size) | -3.548e-10| 0.0000    | -0.53   | 0.5948   |
| file_size:I(1/strip_size) | 0.0145   | 0.0002     | 59.60   | 8.495e-235|
| I(1/file_size):I(1/strip_size)| -6.048e+04| 189846.6291| -0.32  | 0.7502   |

> fit <- lm(read ~ file_size + I(file_size/strip_size), data = duration)
> xtable(summary(fit), display = c("s", "g", "f", "f", "g"))
| Estimate     | Std. Error | t value | Pr(>|t|) |
|--------------|------------|---------|---------|
| (Intercept)  | 0.01076    | 0.0022  | 4.83    | 1.826e-06 |
| file_size    | 1.02e-08   | 0.0000  | 564.83  | 0         |
| I(file_size/strip_size) | 0.01339     | 0.0002  | 86.56   | 1.015e-313 |

The model is $98010423 \times \text{file\_size} + 0.01339325 \times \frac{\text{file\_size}}{\text{strip\_size}}$. The latency for each block is lower than for the writes, but much higher than the latency for reading small messages. The maximum bandwidth is close to the peak one.

4 Conclusion

Proceed to 2 clients, 2 servers and a uniform sampling of the number of blocks.