BigFix FillDB Performance:
The Canary in the Coal Mine

Authors: Mark Leitch, Aram Eblighatian @ HCL Technologies

Coal miners historically would bring canaries in cages down into the mines as a way to warn them of danger. In the presence of methane or carbon dioxide, the canary would die at levels that would not be lethal to the miners. In this way, the canaries would safeguard the health of the miners.

In a similar manner, FillDB is the single best indicator of the health of your BigFix Root Server. If FillDB is healthy, the likelihood is extremely good that you have a robust and well performing BigFix Server. We will describe a mechanism to provide deep understanding of FillDB performance, and thus a good indication to the overall health of your BigFix Server.

Why is FillDB so important?

FillDB is the service that processes the incoming data from all of the BigFix Agents (aka the endpoints) throughout your environment.

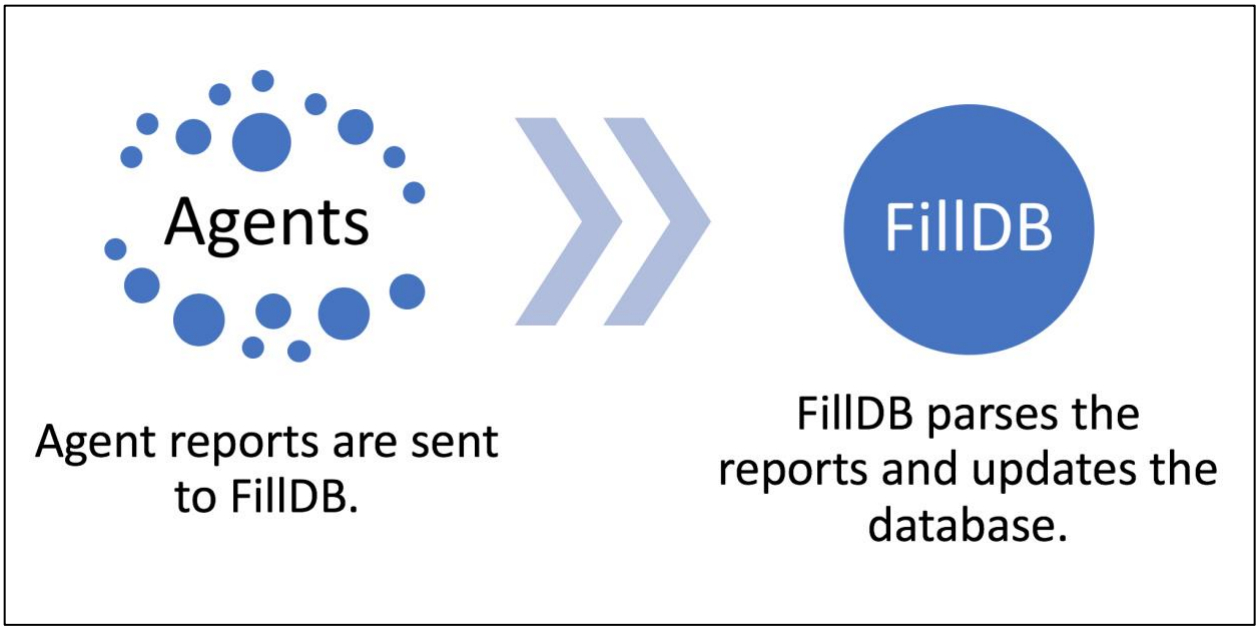

Figure 1: FillDB Processing

FillDB is so important as through this processing it will exercise the following elements.

- Database Management Server (DBMS) performance.
- IO performance, including the DBMS and the FillDB incoming Agent data buffer directory.
- CPU and memory capability of the relevant server components.
- Where applicable, virtualization hypervisor capability.

How to understand FillDB performance?

The first step to understanding FillDB is done by enabling the FillDB performance log (instructions are here). Once you do that, you will have a large and fairly cryptic log. What can you do with this? We provide a FillDB performance log analyzer (aka MXFillDBPerf). The log analyzer is available via the BigFix Performance Toolkit (see here). The syntax follows.

```
usage: MXFillDBPerf [-h] --input FILE [--format {table,csv,json}] [--stats]

BigFix FillDB Performance Analyzer

optional arguments:
  -h, --help                                    Show this help message and exit.
  --input FILE, -i FILE                         The FillDB performance log to be processed.
  --format {table,csv,json}, -f {table,csv,json} The format to use to display the results.
  --stats, -s                                   Generate statistics for the results?
```

Figure 2: MXFillDBPerf Options

A sample invocation of the utility follows. The invocation includes generated statistics, that provide information on the log sample and the utility processing time. It also includes information on the FillDB report parsing and database insertion threads (the default value for each of these is "3").

```
MXFillDBPerf -i myperformancelog.txt -s

FillDB Object                        Count    Time (ms)    Rate/s
-------------------------------  --------  -----------  --------
Fixlet results:                  36204204      2177866     16624
action results:                    247782       120095      2063
short property results:           3080911       296564     10389
long property results:              40935        94549       433
computer administrators:            39359      1732965        23
computer roles:                     39359       124716       316
computer sequences:                138947       105576      1316
computer properties:               138947       545383       255
computer sites subscription:       245610       285136       861
Parallel DB Update (Short Batch):  146401      2445576        60
Parallel DB Update:                 18576       168672       110
Batch Rate:                        164977      2614248        63
Parallel Parsing:                  164977        86602      1905

Statistic                   Value
--------------------------  -------------------------
Lines processed:            285361
Duration (s):               4.94
Throughput (lines/s):       57765.38
FillDB start time:          2017-05-09 09:53:40+02:00
FillDB end time:            2017-05-09 10:52:39+02:00
FillDB duration (hh:mm:ss): 0:58:59
FillDB parsing threads:     3
FillDB database threads:    3
```

Figure 3: MXFillDBPerf Sample Invocation

How do I know if FillDB performance is good?

Okay, this sounds great and everything, but how do I know if these numbers are good? The most critical numbers are the following.

- "Parallel DB Update (Short Batch)" and "Parallel DB Update": The Agent reports are managed in batches with an associated message processing rate. This is a good measure of IO, CPU, and DBMS processing.
- "Batch Rate": The aggregate of the above two updates.
- "Parallel Parsing": The efficiency of the parallel parsing threads. This is a good measure of IO.

In addition to these summary counters, additional counters are provided for transaction rates for the specific database target tables.

This is still a whole bunch of data. How do we know it is any good? To facilitate this, a hidden "health check" option is provided. The option is hidden as it is merely a rule of thumb and there are reasons why even poor values may be acceptable.

```
MXFillDBPerf -i myperformancelog.txt -s -c

FillDB Object                        Count    Time (ms)    Rate/s  Health Check
---------------------------------  --------  -----------  --------  --------------
Fixlet results:                   36204204      2177866     16624  665%: Great
action results:                     247782       120095      2063  413%: Great
short property results:            3080911       296564     10389  2078%: Great
long property results:               40935        94549       433  043%: Poor
computer administrators:             39359      1732965        23  002%: Poor
computer roles:                      39359       124716       316  032%: Poor
computer sequences:                 138947       105576      1316  132%: Great
computer properties:                138947       545383       255  026%: Poor
computer sites subscription:        245610       285136       861  043%: Poor
Parallel DB Update (Short Batch):   146401      2445576        60  120%: Good
Parallel DB Update:                  18576       168672       110  220%: Great
Batch Rate:                         164977      2614248        63  126%: Good
Parallel Parsing:                   164977        86602      1905  381%: Great

Statistic                   Value
--------------------------  -------------------------
Lines processed:            285361
Duration (s):               9.99
Throughput (lines/s):       28564.66
FillDB start time:          2017-05-09 09:53:40+02:00
FillDB end time:            2017-05-09 10:52:39+02:00
FillDB duration (hh:mm:ss): 0:58:59
FillDB parsing threads:     3
FillDB database threads:    3
```

Figure 4: MXFillDBPerf Sample Health Check

In this example, you can see the set of four core performance metrics are in very good shape. However, there is a wide disparity for individual target tables. For instance, while we're seeing great results for Fixlets, actions, and short properties, several other data object types show poor results (e.g. long

property results, computer administrators, and roles). This can be the result of the sample size (larger samples generally perform better), index maintenance issues, or resource contention.

How can I go faster?

In the event you would like FillDB to go faster, there are a few options available.

1. Ensure the base health of the system is in line with the BigFix capacity planning and maintenance guides. In addition, periodic inspection of the FillDB Agent report buffer directory is recommended to ensure it is not backing up. If desired, HCL services may be engaged to perform a health check. If interested in a health check, feel free to contact the authors of this paper.
2. Vertically and horizontally scale system resources, especially CPU and IO.
3. If system resources are available, increase the parsing and database insertion threads for FillDB. Note this is highly dependent on system capability, and you will reach a point where increasing threads will actually slow you down due to thread resource contention.

Further Reading

In the event further reading is desired on BigFix and FillDB performance, the following technical resources are available.

BigFix Platform Documentation: URL
BigFix FillDB Performance Logging: URL
BigFix Configuration Settings: URL
BigFix Capacity Planning Guide, Maintenance Guide, and Performance Toolkit: URL