Distributed Profiler

High level descrition of the current implementation and

Contents

[1 Introduction 3](#_Toc528594967)

[2 Objectives 3](#_Toc528594968)

[3 Use case high level description 3](#_Toc528594969)

[3.1 Calculate Profiles 4](#_Toc528594970)

[4 Architecture 5](#_Toc528594971)

[5 Profile calculation 5](#_Toc528594972)

[5.1 Data handling in Reporter and T&P 5](#_Toc528594973)

[5.1.1 Reporter data aggregation 6](#_Toc528594974)

[5.1.2 T&P data processing 6](#_Toc528594975)

[5.2 Profile Value 8](#_Toc528594976)

[5.3 Mean, variance and standard deviation 9](#_Toc528594977)

[5.3.1 Overflow considerations 10](#_Toc528594978)

[5.4 Profiling external values 11](#_Toc528594979)

# Introduction

SQM (Service Quality Manager) is a Service Monitoring tool from Nokia that operates in NOC (Network Operation Center) and SOC (Service Operation Center) levels. As it is a powerful and generic rule evaluation / correlation engine it is used also in other solutions to fulfill specific use cases as in Nokia Performance Manager tool where it is used for Thresholder and Profiler functionality.

Currently SQM is moving from a huge Java Enterprise Application to a microservice oriented deployment, part of the system is already redesigned, but some components are still part of a big monolith “service”.

# Objectives

The goal of this thesis is to study the best way to evolve the current profiler component to a high scalable microservice oriented component. By the end the profiler should be able to scale horizontally with a linear scalability. So the scope of this document will only be focused in the profiler component and functionality.

# Use case high level description

An overview of main T&P use cases can be found in the next diagram.



T&P Use Cases

## Calculate Profiles

Actor : None, triggered automatically when thresholds are evaluated

Profile value is updated and made available for formula evaluation

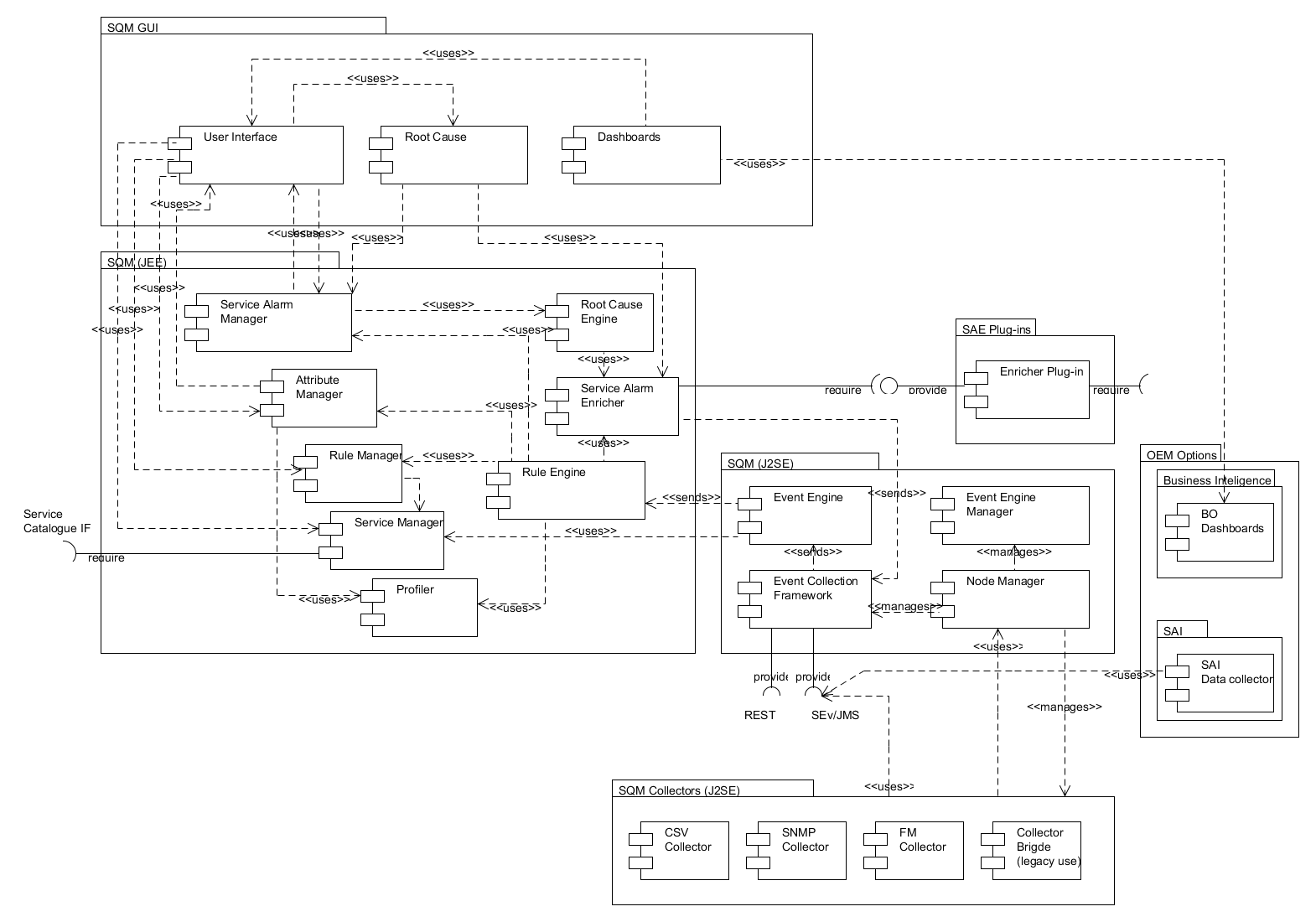
Description:

* Selected KPI values are send as samples to profile engine
  + Each of the KPIs referred in formulas inside the identifier “prof” is selected to be profiled
* Profile is updated and ready to be used in new evaluations

# Architecture

The current version (monolith application) is designed as an enterprise application deployed in JBoss Application server, along with some others standalone Linux services.

Following is the internal view, where it can be easily seen that the profiler component is already well decoupled from the rest of the components.

**

# Profile calculation

## Data handling in Reporter and T&P

This chapter describes how the PM data is aggregated by Reporter and subsequently interpreted by T&P for profiling.

### Reporter data aggregation

The Reporter module is responsible for aggregating PM data, according to the reports a user defines. It is over these reports that T&P thresholds and profiles are calculated.

The Reporter module will aggregate the incoming data according to the report’s configuration. The first step is to decide what is the aggregation period, which can be Raw (no aggregation), Hourly, Daily, etc. This period, *P*, will affect the report’s granularity and scheduling frequency.

Every period *P*, Reporter will run the report, and gather all performance data from the selected PIs, from the selected NEs, and aggregate them into a single value for the period *P*.

So for example, if a set of PIs are selected, and the granularity is one day, then once a day, Reporter will run a job (the report) to aggregate all of that day’s data into a single value for that NE. At the week’s end, we will have 7 samples for each selected PI.

The exact formulas that Reporter uses to aggregate the PM data for a period are not discussed here; please see [[REPARCH](#REPARCH)].

### T&P data processing

It is over the Reporter data output that T&P will run to calculate thresholds and profiles.

The key point to understand here is how the data is interpreted by T&P. As stated before, the various reports created by the user will produce a series of samples. Those samples are ***always*** organized on a weekly basis.

Taking the example from the previous section, we would have, for a given PI, seven samples per week. At the end of, for example, four weeks, we would have 28 data points.

If we would plot these points on a graph sequentially, they would appear like this:

Figure 4: Four week data plotted sequentially

As we can see, this isn’t very useful for data analysis. We can spot a trend, but using this sequential display doesn’t make it easy.

Telecommunications performance data usually exhibits weekly patterns. This is normal, since people’s behavior is patterned according to the general weekly organization of time.

Therefore, T&P will profile the data relative to the same period in the previous week, like so:

Figure 5: Four week data, plotted week-on-week

In this way it is much easier to spot the general weekly trend.

This is also the way that T&P processes data. When calculating the various profile values for a PI / KPI, T&P will always process the data by taking the homologous period’s value from last week. If we display the above charts’ underlying data on a table:

Table 8: PI data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Day** | **Week 1** | **Week 2** | **Week 3** | **Week 4** |
| 1 | 83 | 63 | 73 | 87 |
| 2 | 92 | 82 | 102 | 99 |
| 3 | 68 | 56 | 43 | 45 |
| 4 | 99 | 115 | 127 | 146 |
| 5 | 99 | 95 | 84 | 78 |
| 6 | 89 | 75 | 61 | 60 |
| 7 | 7 | 4 | 4 | 7 |

Then the profile calculation is done for each ***row*** of this table, like so:

Table 9: PI data plus profile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Day** | **Week 1** | **Week 2** | **Week 3** | **Week 4** | **Profile** | **Deviation** |
| 1 | 83 | 63 | 73 | 87 | 77,61 | 10,75 |
| 2 | 92 | 82 | 102 | 99 | 95,25 | 8,88 |
| 3 | 68 | 56 | 43 | 45 | 50,26 | 11,52 |
| 4 | 99 | 115 | 127 | 146 | 127,16 | 19,82 |
| 5 | 99 | 95 | 84 | 78 | 86,38 | 9,69 |
| 6 | 89 | 75 | 61 | 60 | 67,83 | 13,67 |
| 7 | 7 | 4 | 4 | 7 | 8,39 | 7,63 |

Therefore, on the sections below, please remember that the calculations are performed as data comes in for each period *P* of the week, as defined in the report.

## Profile Value

The profile of a certain KPI / PI is a value calculated by taking into account the values the KPI / PI takes over time. It is an Exponentially Weighted Moving Average (EWMA) of the KPI’s value. The actual formula used is the standard online EWMA formula:

So the current profile’s value () is calculated using the previous value and the current KPI value (), also called the current *sample*. The value of *α* used is 0,25. This value is known as the forgetfulness factor – higher alpha values give more weight to recent values (i.e., the formula “forgets” past samples quicker), lower values will take more past samples into account.

The number of past samples taken into account (called the window size in moving averages) can be approximated by the equation

where *N* is the window size. Solving for *N* with , we have *N = 7*; this means we take into account approximately the last 7 samples of a KPI’s value.

This value is available to threshold formulas as the prof function, as outlined in section 2.1.

## Mean, variance and standard deviation

The variance of a KPI is simply the classical statistical variance of the KPI’s value over time. The standard deviation also follows from the classical statistical definition, and is therefore the square root of the variance.

The classical formula for the sample variance is:

Were is a sample; is the sample mean, defined as

where *n* is the number of samples.

The sample variance is used instead of the population variance, since the KPI values we receive from Reporter are in fact samples, not the entire sequence of values as produced by the NE. Therefore using the sample variance is much more appropriate.

From the sample variance we can get the (sample) standard deviation, which is simply .

As we can see from the formulas above, their calculation would require Profiler to keep some number of samples (and, to be accurate, all samples) collected. The formula can be replaced by a similar on-line calculation formula[[1]](#footnote-1):

As in the profile calculation above, the new value is calculated using the previous variance value and the current sample value.

It turns out that the easiest way to translate this into an algorithm is to maintain a running (on-line) sum of squares difference

with the variance as

As before, simply taking the square root of this value will yield the sample standard deviation. The following pseudo-code implements the formula above, also maintaining a running calculation of the mean:

def online\_variance(data):

n = 0

mean = 0

M2 = 0

for x in data:

n = n + 1

delta = x - mean

mean = mean + delta/n

M2 = M2 + delta\*(x - mean) # This expression uses the new value of mean

variance = M2/(n - 1)

return variance

### Overflow considerations

The above algorithm maintains an ever-growing variable, M2. We can see that its value never decreases, since it accumulates the sum of differences throughout the lifetime of the calculation (i.e., as long as the profile is calculated).

The variables are kept in the database, using a column data type of NUMBER, with no capacity qualifiers. On Oracle databases, if no precision and scale qualifiers are given, the maximum numbers the data type will hold are up to 38 digits, with no fractional part, which is equivalent to the range of .

Since NetAct Base imposes a minimum data sampling period of one minute, we will have at most, for a year’s worth of samples, samples.

Using this number we can see how many samples could be gathered for a given time period. From there, we estimate the maximum per-iteration average increase of M2, by simply dividing the maximum capacity by the number of samples, as outlined in the following table:

Table 10: Maximum values for the M2 variable

|  |  |  |  |
| --- | --- | --- | --- |
| **Years** | **# Samples** | **Maximum M2** | **Maximum difference** |
| 1 | 524160 | 1,91E+32 | 1,38E+16 |
| 2 | 1048320 | 9,54E+31 | 9,77E+15 |
| 3 | 1572480 | 6,36E+31 | 7,97E+15 |
| 4 | 2096640 | 4,77E+31 | 6,91E+15 |
| 5 | 2620800 | 3,82E+31 | 6,18E+15 |
| 10 | 5241600 | 1,91E+31 | 4,37E+15 |
| 20 | 10483200 | 9,54E+30 | 3,09E+15 |
| 100 | 52416000 | 1,91E+30 | 1,38E+15 |
| 1000 | 524160000 | 1,91E+29 | 4,37E+14 |

We can also then estimate the maximum difference that samples can take from the mean. Remember that M2’s formula is

We can approximate the increase, for the purposes of this exercise, by noting that

and so, we can take the square root of “Maximum M2”, and get an estimate of the (average) maximum difference to the mean supported; this is shown in column “Maximum difference”.

So, for example, for 20 year’s worth of data, we cannot increase M2, on average, by more than on each calculation round; otherwise we risk exceeding the variable’s capacity.This M2 value, in turn, implies a certain maximum difference from the mean the sample can have, which in this case is .

As we can see, even if we consider 1000 years of data at the minimum platform sample interval, we can add to M2, every minute, at the staggering rate of , implying that that minute’s incoming sample would have deviated from the current mean by about 430 billion (430 trillion US scale).

With this data in mind we can safely assume that the maximum range of the variable’s data type will never be reached in the projected useful lifetime of the system.

## Profiling external values

Apart from profiling data originated from a report, T&P can also profile data from an external source. This data is fed via the profiling interface (described below), where the values, granularity and timestamp are provided. Profiler will take these values and apply the rules described above, and store them in the database. The profiling interface then allows the external system to retrieve the calculated profiles.

1. See Donald E. Knuth (1998). The Art of Computer Programming, volume 2: Seminumerical Algorithms, 3rd edn., p. 232. Boston: Addison-Wesley and B. P. Welford (1962)."Note on a method for calculating corrected sums of squares and products". Technometrics 4(3):419–420 [↑](#footnote-ref-1)