PEOPLESOFT: PROPERLY INSTRUMENTED FOR PERFORMANCE TUNING?

David Kurtz, Go-Faster Consultancy Ltd.

As a consultant who specialises in performance tuning, the recently introduced PeopleSoft Performance Monitor has been of particular interest to me. It has provided a method to collect various performance related metrics from the PeopleSoft technology without the need to resort to esoteric scripts. It enables the user to determine how much time was executing which part of the application, and in which tier of the technology.

Since version 6 of the Oracle RDBMS, it has been possible to enable SQL*Trace (event 10046) on a session since. Additional code in the Oracle kernel writes information to a trace file about every SQL statement, parsed and executed, every fetch operation and execution plan, and how long each operation took. The trace can be enhanced to additionally record time spent waiting on an ever-growing list of database events, and the values of bind variables referenced in the SQL statements. Hence, the DBA can determine exactly what is going on and how long it is taking.

As a young naïve DBA I used to believe that all you had to do to improve the performance of the database was measure the buffer cache hit ratio, and if it fell below 90% then it was time to add more memory to the block buffer cache. Later, I learnt that performance tuning is all about the time lost when a user waits for the application. SQL*Trace files can be processed with either Oracle's TKPROF, Trace Analyser or a third party profiler in order to identify the SQL operation, or database event that took the most time. However, over the years as multi-tier web-orientated architectures have become prevalent, applications spend less time in the database and more time executing code in an application server or java servlet (or both in the case of PeopleSoft). Therefore, I increasingly see Oracle reporting significant quantities of time spent on the event 'SQL*Net message from client'. DBAs often discarded this as a mere 'idle event', but the database might be idle because the middleware is busy, in which case this is time lost to a user that affects performance. It is ironic that as time-based analysis of performance issues has become generally recognised as the most (if not the only) effective technique, the tools that Oracle provide to investigate database performance issues have been made less effective in resolving system performance issues.

PeopleSoft have instrumented every part of their Enterprise PeopleTools technology, from release 8.44. As each operator action generates activity, this can be recorded as a number of transactions. These are recorded in the database. It is now possible to determine exactly which operator executes which piece of code and how long it took to execute. You can think of the PeopleSoft Performance Monitor as event 10046 for the application.

PeopleSoft used its own existing PeopleTools technology to collect and store the performance data. A new servlet was added to the web server to receive performance metrics from the instrumented processes, and a new server process was added to the application server to store that data in the database. Each of the instrumented processes connects over HTTP to the monitoring servlet.

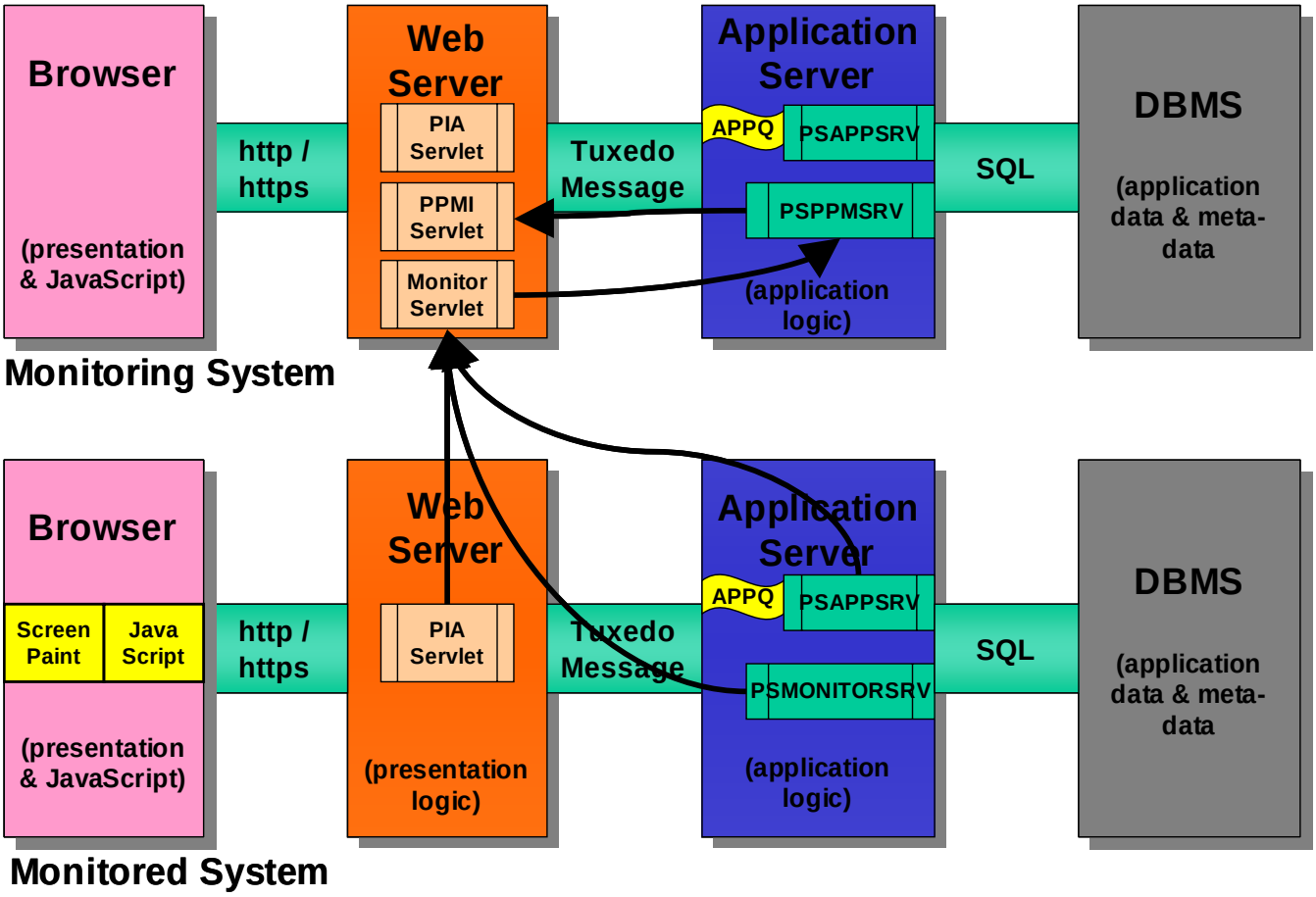


Figure 1. PeopleTools Performance Monitor Architecture

Although a system can monitor itself, the recommended configuration is for one PeopleTools system to be configured to monitor another. Thus the overhead of storing performance data is not borne by the monitored system. To configure the Performance Monitor is merely necessary to give the monitored system the URL of the monitor servlet in the monitoring system.

Performance Metrics are separated into Transactions and Events.

Events are instantaneous metrics that can be sampled periodically, such as CPU and memory utilisation, or are collected in response to a certain conditions, such as the time-out of an ad-hoc query.

Transactions are a part of the activity that occurs when a user does something in the application that causes an interaction with the middleware. Therefore, they also have a duration. One interaction will produce a hierarchy of many transactions; this is called a Performance Monitoring Unit (PMU). The Performance Monitor can be configured to sample PMUs across the system, say 1 in every 40, so that a picture of the whole system can be built up, without creating excessive load. It is possible to enable the Component Performance Trace, which collects all the PMUs for a user's session.

The Performance Monitor is supplied with a number of analytic pages that perform queries on the performance data back from the database. For example, Figure 2 shows the system summary page from which you can drill into individual server processes.

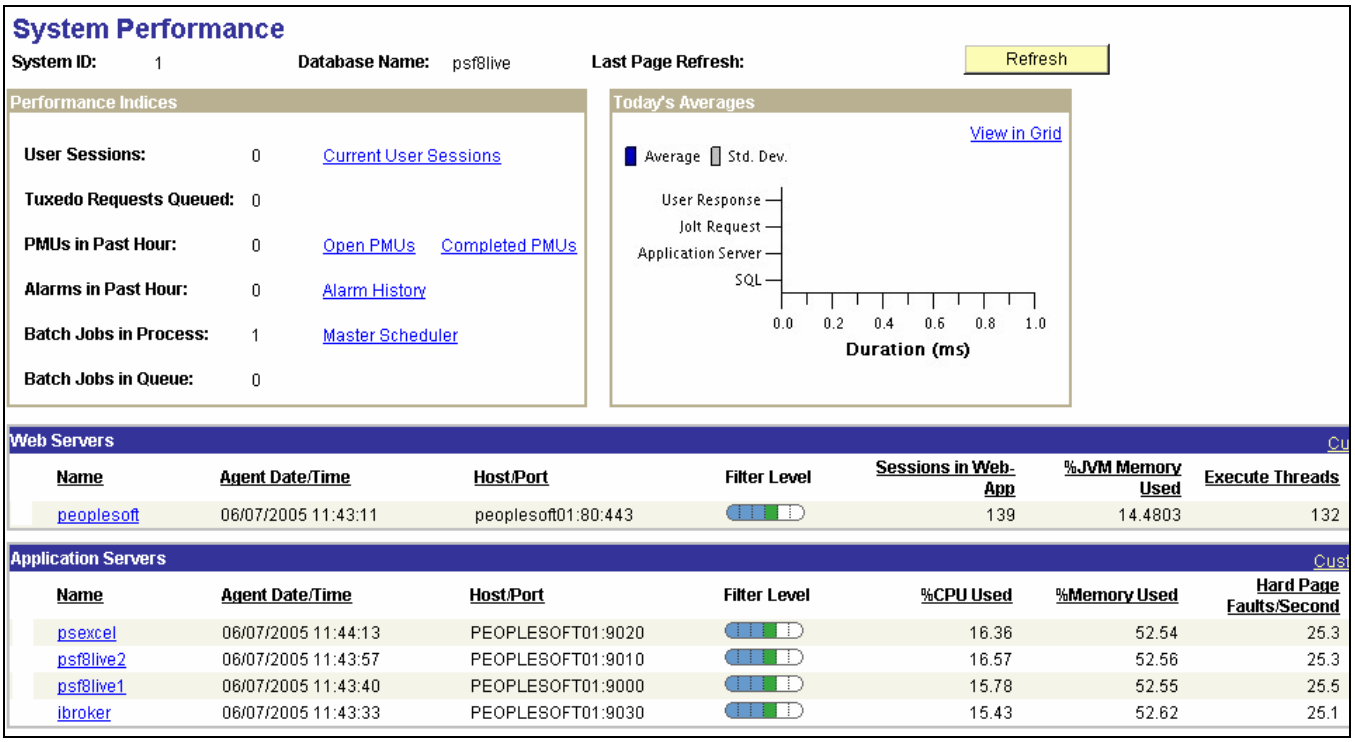
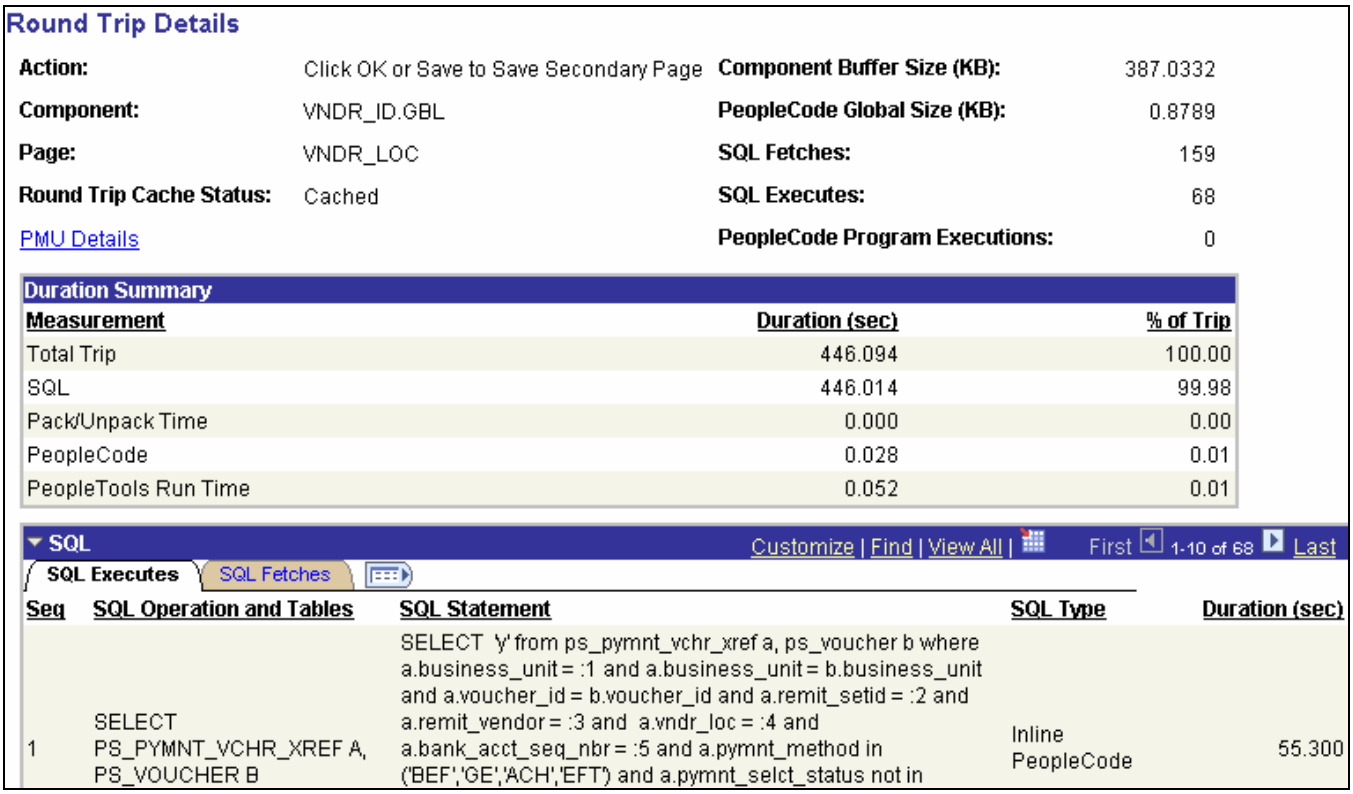


Figure 2: System Performance Summary Page

These are standard PeopleTools components that are developed in the usual way. The analytics delivered in PeopleTools 8.45 are considerably more sophisticated. However, since the takeover of PeopleSoft by Oracle this area is unlikely to develop further in PeopleTools, although, there is therefore nothing to prevent users from developing the own custom analytic pages. For example, the chart in Figure 3 is from a component that identifies the top 10 transactions by cumulative execution time. Clearly one transaction needs to be looked at closely.

Figure 3: Part of Top Component Page

Figure 4 shows the analysis of a Component Performance Trace of the part of the application with the problematic transaction shown in Figure 3. It shows that during the trace 99.98% of the response time is waiting for SQL execution. This case is clearly a SQL problem. You can see the SQL statements and the values of the bind variables order by execution time, starting with the longest.



Round Trip Details

Action:	Click OK or Save to Save Secondary Page	**Component Buffer Size (KB):**	387.0332
Component:	VNDR_ID.GBL	**PeopleCode Global Size (KB):**	0.8789
Page:	VNDR_LOC	**SQL Fetches:**	159
Round Trip Cache Status:	Cached	**SQL Executes:**	68
PMU Details		**PeopleCode Program Executions:**	0

Duration Summary

Measurement	Duration (sec)	% of Trip
Total Trip	446.094	100.00
SQL	446.014	99.98
Pack/Unpack Time	0.000	0.00
PeopleCode	0.028	0.01
PeopleTools Run Time	0.052	0.01

SQL Customize | Find | View All | First 1-10 of 68 Last

SQL Executes | SQL Fetches

Seq	SQL Operation and Tables	SQL Statement	SQL Type	Duration (sec)
1	SELECT PS_PYMNT_VCHR_XREF A, PS_VOUCHER B	SELECT 'y' from ps_pymnt_vchr_xref a, ps_voucher b where a.business_unit = :1 and a.business_unit = b.business_unit and a.voucher_id = b.voucher_id and a.remit_setid = :2 and a.remit_vendor = :3 and a.vndr_loc = :4 and a.bank_acct_seq_nbr = :5 and a.pymnt_method in ('BEF','GE','ACH','EFT') and a.pymnt_selct_status not in	Inline PeopleCode	55.300

Figure 4: Analysis of Component Performance Trace

You have to cut and paste the SQL into a database tool, or just SQL*Plus, in order to get the execution plan. PeopleSoft doesn't do this for you because that is database platform specific (and of course you will probably be reviewing performance data on a different database).

You can also look at PMUs in the trace. They can be presented as a hierarchy or tree (see Figure 5). The entries are listed in the order they executed. You can see how much time each transaction took, and what other transactions where a part of it.

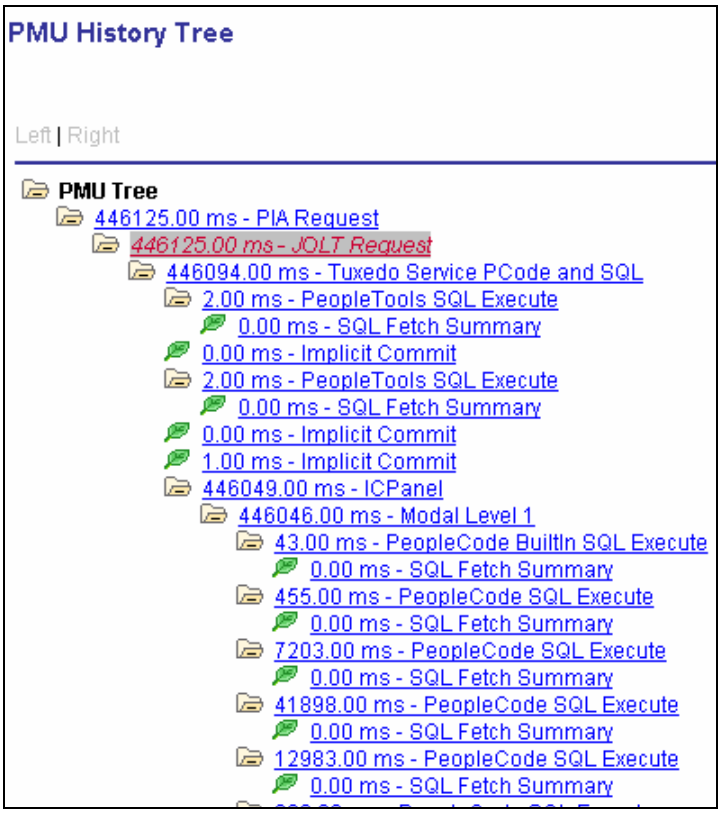


Figure 5: Performance Monitoring Unit Tree

You can then drill into individual transactions and see what they did, and what part of the application was executing, as shown in Figure 6. So if you need to change the code, you can find it easily.

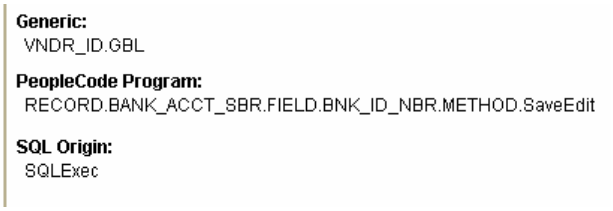


Figure 6: Performance Monitoring Transaction Context

In conclusion, the PeopleTools Performance Monitor is extremely impressive. It is the more sophisticated that anything else I have seen or heard of. It could revolutionise analysis of functional and performance problems. (It could also but me out of business!) When an end-user reports a performance problem, it is reasonable to ask them to enable the performance trace and then demonstrate their problem. The customer's in-house support team, who will necessarily be familiar with PeopleTools, can understand and analyse the information, determine whether there really is a problem, and if necessary make changes to the code.

I'll leave you with two questions: will the Performance Monitor will be built into the Fusion product from day one, and if so will you have to pay extra for it? PeopleTools Performance Monitor is not a separately licensed option. If you have PeopleTools you have the Performance Monitor.

David Kurtz is a performance specialist working Enterprise PeopleSoft applications and Oracle (www.go-faster.co.uk). He is the author of 'PeopleSoft for the Oracle DBA', published by Apress (www.psftdba.com). He is a director of UK Oracle User Group.