

Introduction

by

ProfDog was originally developed in order to better understand the interactions between components of a complex three-tier application. We wanted to know:

- how chatty the application was over the network

The application in question is a "fat" PC client (.NET) invoking web services to IIS, within which a custom ORM-layer interacts with a back-end SQL Server database. We could independently run SOAP trace tools for the PC/IIS interaction and the SQL Server Profiler utility for the IIS/Database interaction, but it was hard to correlate the two.

- discover where the time was being spent

We expected that the data layer should account for perhaps 80%-90% of the total response time, but SQL Server profiling suggested that wasn't the case. We had various theories as to whether the additional time was being taken in inefficient domain logic, or in application-level encoding, or in IIS-level SOAP encoding marshalling/unmarshalling, or whether our heavy use of reflection was the root cause.

- provide the capability for production-time monitoring of the service

We wanted to monitoring capability that was lightweight enough that it could potentially be left permanently switched on even on the running production server.

With ProfDog we now have a tool that addresses these points.