End to End Performance Monitoring at BARCLAYS BANK

Barclays Bank is one of the major financial institutions within the UK. These notes describe the steps that have, and are being taken to manage the IT services delivered to UK Banking Services.

UKBS is the retail banking division of the Barclays Group. It is the largest division in the Group and is divided into seven discrete sectors: Personal, Business, Risk, Operations, International, Finance and Personnel. All of these sectors share access to, and use of, UKBS’s major computer systems.

There are three other main players in the IT supply chain: Barclays Technology Services (BTS), IT Service Management (ITSM) and UK Systems Development (UKSD).

BTS is responsible for the building and maintaining of the core technology infrastructure of the Barclays Group as a whole. This includes the operation of the main computer centres, the telecommunications infrastructure, (comprising Barclays owned and BT leased networks), and other major infrastructure service components.

ITSM is the operational owner of shared IT services. ITSM ensures that all IT systems, whether under development or already in use, meet operational business targets in terms of performance, availability, cost and stability. Shared services include the Branch Platform, telephony, self-service operations (which includes ATM’s and PC banking), as well as key enterprise wide systems such as Branch Accounting, Counter and Customer.

The function of UKSD is to design, build and maintain IT systems on behalf of UKBS, making it our customer and partner. UKSD’s role does not, however, extend to all business units within UKBS. For example, Barclays Life and Barclaycard each have their own IT services function.

UKSD comprises four main groups: Resources, Projects Group, Operational Systems Development (OSD) and Technical Services Group (TSG).

Resources deal with personnel and staffing issues. Projects Group are responsible for the delivery of new projects from inception through to implementation. In production the projects become the responsibility of Operational Systems Development who both support and enhance existing systems.

Technical Services Group are a central function who supply specialist skills to both Projects Group and OSD. There are a number of different disciplines supported from within TSG, including Technical Design, Database Administration and Performance Assurance to name but three. It is within the Performance Assurance Team (PAT) that Dave and I both work.

The PAT support projects through all the stages in their development, and on into production.
During project inception, design and build, PAT assists projects in understanding the performance attributes of the systems they are building. They provide modelling services to predict the likely performance of services being designed. They also assist the projects in defining the workload profiles and likely resource usage, information which is then fed through to BTS to provide timely capacity planning information.

Seven or eight years ago the Barclays computing environment was relatively simple. The majority of logic was executed on a central server, there was very little distribution of logic. The service provided to the End User was generally simple, the majority of dialogues consisting of one input message and one output message.

Measurement of these systems was carried out using standard monitors which measured the elapsed time of a request arriving at the central server to its’ output acknowledgement at the receiving platform.

Over the next few years a number of events took place which significantly changed the picture:

- 1989 Our then IT systems director Joseph De Feo declared that Barclays would develop an Open Systems Architecture.

- 1990/91 A number of business prototypes were built.

- 1992 The first Branch Platform was delivered. This was built on a standard UNIX box and offered a number of standard services including:
  - User Registration
  - Security Services
  - Print Services
  - Software Distribution
  - Backup and Recovery Services

- 1993 The first major client server application was delivered

- 1994 Barclays Customer System implemented

The impact of the events charted above has been dramatic. Resulting in an explosion in the number and types of devices attached to our central servers.
To add to the challenge there has been a dramatic increase in the workloads placed across these new architectures:

- Customer System processes 800,000 transactions per day
- Branch Accounting posts 8,000,000 entries every night
- Barclaybank ATM’s service 235,000,000 cash withdrawals per year
- Barclays clearing systems handle 3 - 6 million cheques per day

To support this workload Barclays needs:

- 5 data centres
- 10 IBM ES9000
- 7 CMOS boxes
- > 3,000 UNIX Servers
- > 3,000 ATM Machines

A significant amount of logic has now been distributed onto many different types of device. The services supplied to users are often complex, combining multiple input and output messages, with logic being executed on different platforms.

It is no longer possible to monitor these services from the outside using standard monitors because there isn’t one. An additional challenge is that the open architectures, (in general), do not recognise the concept of a transaction, a new approach is therefore needed.

Why are we going to all this trouble? Because as service providers, we must be able to measure the quality of service that we deliver to our Customers in a clear and objective way. There is a danger that if they come to rely on subjective measures, they may become disillusioned with the service provided and they will go elsewhere!

Our Customers are becoming more sophisticated and demanding, they are no longer prepared to sit back and accept whatever we are prepared to give them. They can’t afford to, because they realise that it’s IT which provides them with a competitive edge. It is an integral part of the service that they supply to their Customers.

As a result they are demanding the introduction of Service Level Agreements (SLA’s). This will allow them to judge the quality of service in an objective way. We in turn must provide a mechanism by which those services can be measured and reported on.

There are however potential benefits which may be realised from the collection of metrics on service usage and performance:

The monitoring of services could allow the raising of alerts when critical thresholds are breached. This would permit the proactive resolution of problems, instead of
waiting for Customers to report problems through help desks. This, over a period of time, would lead to improvements in the quality of the service being delivered.

By generating and collecting data about service usage, we can add value to the service that we supply. The data is a potentially valuable source of management information, it could be loaded into a data warehouse, and analysed to allow the targeting of business resources more effectively.

Data collected is also a valuable source of information from which to discover trends in service usage. This may be fed into the capacity planning process, providing more timely and accurate information than might otherwise be available. It can also be used for comparison with the workload definitions held in the SLA’s.

Barclays aim is to develop generic services which themselves may be re-used many times. Two of the most heavily used services are ‘Locate Customer’ and ‘Download Customer Data’, because you can’t perform any service for a Customer without first identifying who they are, and secondly what accounts they have.

As more and more generic services are built and used by different business sectors, and across different delivery channels, it becomes increasingly difficult to apportion accurate charges for those services. It follows therefore that if service usage can be broken down by sector and/or delivery channel, not only can charges be made against services used, but the relative costs of providing the same service through different delivery channels may be determined.

There is an additional benefit which can be realised from service measurement during the testing phase of a project. Anyone who has spent many happy hours trying to find out where a transaction went to, by scouring IMS logs will appreciate the benefit that a ‘roadmap’ of a transactions progress would give.

If we can’t measure it, we can’t manage it effectively! How do we decide where to target scarce resources if we don’t know where the problems are?

We believe therefore that the key to providing effective service measurement is to add instrumentation to applications, it must become as integral a part of applications as error checking.

Why is there a need for instrumentation within applications? In 1996 there were no monitors (that we knew of), which could measure across the heterogeneous environments in which we operated. Our view was that it was extremely unlikely that anything would emerge in the short to medium term. The instrumentation of applications was therefore the only viable solution.

Having drawn that conclusion, the next question to be answered is - ‘What should be measured?’ ‘The end to end response time’ is the obvious answer. If however a problem is detected with the performance, the next logical question to ask is why? where is the time being spent? This may be difficult to determine, let me explain why.

The following events represent the flow through one of our simpler services - Customer Locate, it gives an indication of the number of components involved:
1.) A personal banker (PB) enters a sort code and account number, then hits ENTER
2.) Data is transmitted across a LAN to the Branch Platform (BP).
3.) On the BP some validation of the input is performed, a transaction is formatted and passed to the Transaction Handler for transmission to a central server.
4.) The transaction traverses the WAN to the central server.
5.) The central server processes the transaction and creates the response.
6.) The returning transaction passes back over the WAN to the BP.
7.) The response is processed, data decoded and a screen is formatted.
8.) Finally the screen is transmitted across the LAN to the PB.

There is the potential for a performance problem to occur in any of the components listed above. The ability to determine the time spent in each component will greatly speed problem investigations when they occur.

Having detected in which component the most time is being spent, it may be necessary to break a component into its’ sub components. On a mainframe this may mean breaking out the time spent in transaction, database and I/O services. On the BP it may require breaking the time spent in the various platform services eg Transaction Handler and Application Support Layer. Let me explain what they are:

The Transaction Handler is a service similar in function to MQSeries. It isolates the complexity of communications from the calling service, it also provides guaranteed message delivery. All application services which need to communicate with other servers, do so via the Transaction Handler.

The Application Support Layer is a set of services, which provide common functions like data conversion or calculation. It also provides the means by which application services are started and stopped. Its’ aim is to simplify the programming environment, it does however complicate the collection of performance metrics.

From a performance measurement perspective there is a further complication. As previously stated, Barclays aim is to build services which may be reused many times, it is therefore difficult to measure a complicated service.

One of our more complicated services is the Small Business Loan Scoring Service. Before I explain its’ composition, I’d like to introduce a few terms:

Business Process - This is the set of activities, both computer based and manual tasks which need to be completed to achieve a business objective eg ‘Perform an Account Transfer’ or in this case ‘Score a loan application’.

Business Transaction - This is defined as the time from the user submitting a request to receiving a response, ie finger to screen.

Computer Transaction - This is defined as the time from an application submitting a transaction request, to the application receiving a response.

In this example the ‘Score a Loan Application’ is the business process. This business process is made up of two business transactions with a manual task in between.

The first business transaction is ‘Customer Identification’, which will initiate the computer transaction ‘Locate Customer’ for each combination of sort code and
account number entered. The names and basic details of all Customers associated with those account will be returned and displayed.

The user then has to decide which of the Customers returned are to be used in the scoring process, this is the manual component of the business process.

Having selected the Customers, a second business transaction is initiated, this is the ‘Score Business Transaction’. This will issue a computer transaction ‘Customer Primary Download’ for each Customer selected by the user, which will download the complete set of accounts associated with the Customer. Without returning control to the user, the application will create the computer transaction ‘FHA Score’ which will perform the scoring process before returning a decision which will be displayed to the user. This signifies the end of the second business transaction and the completion of the business process.

Understanding what has happened, and how long it has taken is almost impossible without having a thread which links all the components together. This is achieved using application instrumentation.

Having decided how and where to generate the data, the next challenge is to collect it. At Barclays we currently employ two methods of collection. One we’ll term ‘piggybacking’, the other consolidation.

The ‘Piggybacking’ technique is used by the ATM and Counter services. It consists of data from the current transaction being processed at the central server, and stored away in a database complete with unique key. The message returned to the originating point is used by the ATM software to calculate the overall response time. This is then returned with the next transaction complete with the previously generated unique key, on the ‘back’ of the next transaction to leave that device. The data held on the central server may then be analysed to marry transaction details with end to end response time.

The second technique involves the writing to a file, (held on the client), of significant events and their timestamps. This data is then subsequently collected centrally to be analysed and reported on.

There is a third technique for collection which we will call ‘Dynamic Collection’. It currently is no more than ‘vapourware’, but could provide many benefits. In essence it would consist of a number of services which would ‘catch’ event data passed from applications. This data would be dynamically analysed to determine response times, and whether they were within acceptable service levels. Any SLA failures could be used to trigger alerts, perhaps to one of the proprietary service monitors. The performance data collected could be built into a transaction and transmitted at regular intervals to a central collection point. This would provide almost ‘real time’ response time monitoring.

Having collected the data the next task is to analyse and report it. We chose SAS IT Service Vision, combined with the complementary SAS family of products because it gave us:
• Software from a company recognised as a leader in performance and capacity reporting

• An integrated set of software with the requisite features to allow us to rapidly build a reporting infrastructure

• The ability to develop a solution to pre determined hardware/software budgets, with the ability to upgrade in a controlled way

• The potential to migrate to alternative software platforms if required

The result is BarclayView.

An end to end performance reporting tool with an intuitive user interface which demonstrates the quality of service being delivered.
End To End Performance
Monitoring At
BARCLAYS BANK

Derek Cowell
Dave Knight
Barclays IT Supply Chain

UKBS

ITSM    BTS    UKSD
Significant Events

1989 Open Systems Declaration

1990/91 First Business Prototypes

1992 Branch Platform Delivered

1993 First Major Client Server Application

1994 Customer system Implemented
Barclays Workloads

- Customer System processes 800,000 transactions per day
- Branch Accounting posts 8,000,000 entries every night
- Barclaybank ATM’s service 235,000,000 cash withdrawals per year
- Clearing Systems handle 3 - 6 million cheques per day
Barclays Computing Resources

- 5 Data Centres
- 10 IBM ES9000
- 7 CMOS Boxes
- > 3,000 UNIX Servers
- > 3,000 ATM Machines
Why is there a need for Self Instrumentation?

- Varied Heterogeneous Environment
- No Current Monitoring Tool Can Help
- Unlikely That a Tool will Emerge in the short/medium term

IF WE CAN’T MEASURE THEN WE CAN’T MANAGE
Component Parts of the Transaction Lifecycle
Branch Platform Architecture

Application Support Layer

- Service A
- Customer Locate
- Customer Download
- Score

Print

Txn Handler
Benefits Of Response Time Management

- Monitor Adherence to Service Level Agreements
- Automatic exception Reporting
- Management Information
- Prediction of Future Capacity Requirements
- Apportioning of Operational Costs
A Solution - Self Instrumentation

• The generation of data by a service about its’ own operation
  – Capture
  – Analysis
  – Reporting
SI Collection Mechanisms

- “Piggybacking”
- File Consolidation
- Dynamic Capture
BarclayView

An end-to-end performance reporting tool with an intuitive user interface which demonstrates the quality of service being delivered
BarclayView

- BarclayView Infrastructure
- SLA reporting
- BarclayView Demonstration
Infrastructure

- SAS IT Service Vision on a UNIX platform
- SAS performance databases
- SAS ITSV software handles data management/summarisation
- SAS/AF application User interface
BarclayView: SYSTEM DESIGN

Data on multiple platforms

Performance Assurance
UNIX box

BarclayView REPORTING
SAS AF Application
SLA Reporting Conventions

- **SLA performance indicator traffic lights:**
  - Met SLA = Green
  - 90-100% SLA = Amber
  - Over SLA = Red

- **Amber allows prediction of problems before they affect the Customer**
BarclayView Demonstration