2009 WebSphere System z Podcasts – 'Did you say Mainframe?'

TITLE: Enhancing Java environments with pureQuery

HOST: Hi, and welcome to the "Did you say Mainframe?" podcast series. This is where we regularly interview IBM technical experts who can help you to understand important IBM mainframe hardware and software issues. I'm your host Sherrie Abshire.

Today we're going to talk about Integrated Data Management and,in particular, pureQuery technology.

Our guest today is Holly Hayes from the Data Studio Enablement Team. Holly, it's great to have you here.

SME: Thanks, Sherrie, I'm glad to be here.

HOST: Before we begin, I'd like to mention to our listeners that there is a teleconference entitled 'Let pureQuery improve the quality of service and reduce costs for WebSphere and DB2 applications'. I'll be giving out more information at the end of this podcast.

HOST [Question 1]: So Holly, Let's start with what is Integrated Data Management and why should we care about it?

SME [Answer 1]: Sure, Integrated Data Management puts a focus on managing data across its full lifecycle, beginning when you first describe the requirements through development, deployment, and all the way through archival or deletion. I like to say "from requirements through retirement". Lots of different people get involved across the lifecycle: business analysts, data architects, developers, testers, administration, compliance officers… But the tools in the market have not done a good job of helping all these folks collaborate. So IBM is developing a set of integrated capabilities — integrated meaning common policies, common models, and common metadata — shared across the lifecycle. Thus, the tools will improve collaboration and efficiency among the people involved in the process. This should translate into helping companies grow their business without growing cost, produce enterprise-ready applications faster, meet increasingly challenging service level targets, and help the business side and the IT side work together to enhance data governance.

HOST [Question 2]: So what is pureQuery and how does it fit into the picture?

SME [Answer 2]: I like to think of pureQuery as part of the secret sauce of Integrated Data Management. pureQuery is a high performance data access platform that provides value across the lifecycle and enables developers and DBAs to work together better.

Organizations that are adopting pureQuery are looking for a number of things First, to deliver enterprise-ready applications faster – pureQuery gives developers productivity boosts for developing Java database applications and helps then adopt best practices for higher performance. But it also gives

developers and DBAs more flexibility and control over execution characteristics of the data access. One key feature is that it enables static SQL execution against DB2 databases which has lots of benefits.

Moving from dynamic execution to static execution can improve performance and significantly reduce hardware and software costs because it reduces CPU/transaction for DB2 data access. This is actually the number 1 reason that organizations are looking at pureQuery today. But its not the only reason. In addition, pureQuery can also enable more function to migrate to zIIP and zAAP specialty processors to reduce costs.

pureQuery improves quality of service by locking in access plans so you get guaranteed service levels. And it helps speed up problem isolation to resolve issues faster.

And, finally, it can enhance security. It can eliminate SQL injection risk by allowing you to lock in the allowable SQL. And moving to a static SQL execution model has a different security model that grants authorization to execute just a set of SQL statements rather than granting access to the table itself.
So that's why I call pureQuery a lifecycle technology because it is adding value to application development, deployment, operation, and optimization.

HOST [Question 3]: Does pureQuery have any benefit for existing environments?

SME [Answer 3]: Yes it does. You don't have to be doing new development. You can use pureQuery to improve existing applications without modifying them. We can capture the SQL and related metadata from an executing application. We would expect to do this in a test environment. Once we have this, then we have the basis for switching from dynamic SQL to static SQL to reduce costs, enhance security, improving manageability, and so on. Plus you can visualize the metadata to help with impact analysis, like understanding how schema changes might affect the application. Or we can use that same metadata to facilitate problem resolution by showing where a specific SQL statement is issued within the application.

HOST [Question 4]: What about organizations that are using other Java frameworks? Does pureQuery provide any value to them?

SME [Answer 4]: Yes. Absolutely. pureQuery technology can be used either as a complement to existing framework deployments or as an alternative to them. As an alternative, pureQuery provides the ease of use associated with object-relational mapping technologies, but provides greater control over the generated SQL. So pureQuery might be used instead of frameworks where persistence management is not needed or where performance requirements dictate greater control over the generated SQL.

But, back to your original question, pureQuery can be an effective complement to frameworks, like Spring, iBatis, Hibernate or JPA. As a matter of fact, we can use pureQuery with any framework that generates JDBC. So as I was just talking

about a moment ago, we can capture the SQL and related metadata from an executing application and that would be a framework-based application. We can use that information to improve performance, make response times stable, and enhance security and manageability. What is really nice for framework environments that generated the SQL is the pureQuery correlates the application code with the generated SQL and you can visualize that to speed problem isolation and support impact analysis.

HOST [Question 5]: Are there specific synergies with WebSphere?

SME [Answer 5]: Definitely, pureQuery is built to run anywhere but it is optimized for WebSphere. So WebSphere Application Server V7 delivers an enhanced JPA (Java Persistence API) implementation that supports pureQuery. It allows us to gather the SQL metadata without executing the application. And it automatically takes advantage of static SQL as well as other performance enhancements. Plus the WebSphere administrative console has been extended to make it more convenient to take advantage of pureQuery capability. As a matter of fact, there's a developerWorks article that was recently published on Integrating JPA and pureQuery.

I'd also like to mention that pureQuery was originally built as the data access layer for Project Zero, now WebSphere sMash, so it gets extensive use in that context as well.

HOST: Thank you Holly, that was really interesting.

SME: Thanks very much for giving me the opportunity to talk about pureQuery.

HOST: Well, that wraps up this podcast discussion. To find out more about the teleconference on pureQuery I mentioned earlier, please go to the description for this podcast at:
http://www.ibm.com/software/os/systemz/podcasts/websphereonz

There you will also find the URL for registering for the IBM Impact 2009 event which is being held in Las Vegas in May.

Join us next time as we talk about another important mainframe topic. For now, this is Sherrie Abshire saying "Thanks for listening".