Performance Analysis Study of XBRL Document Processing

A systematic approach for evaluating XBRL processing engine performance

Executive Summary

This white paper is a study intended for IT managers and developers who have an interest in adding XBRL capabilities to their applications environment. The goals of this paper are to offer the reader a straightforward methodology for accurate performance testing of XBRL processing engines, to illustrate the use of the methodology in testing UBmatrix™ Processing Engine, and demonstrate through detailed testing results how the UBmatrix XBRL processing engine offers enterprise-class performance for the XBRL-enabled environments.

Copyright © UBmatrix, Inc. 2008

Table of Contents

Introduction

When selecting an XBRL (eXtensible Business Reporting Language) processing engine, it is vital that companies analyze the various factors that impact processing time, understand how to optimize the engine based on the given systems environment, and test the engine in that environment to assess actual performance. In the absence of industry performance standards, processor performance and tuning can be unpredictable, leaving open an undesirable potential for error and delivering testing results that can be inaccurate and misleading.

To facilitate a more systematic approach to performance testing, UBmatrix has invested significant time and resources to create this methodology and document it in this paper. The methodology includes:

- real-world simulations and a variety of performance testing scenarios
- stress test scenarios to evaluate the impact of excessively large and complex documents
- comparison of performance impact of single- and multi-threading CPUs
- granular performance measurements to aid in engine tuning

  This methodology offers a way to produce an 'apples-to-apples' performance validation that considers the many factors that can impact processing speeds, as well as introduces opportunities for tuning engines to optimize performance in a given environment.

Methodology Overview

UBmatrix has created a testing methodology that offers a practical and predictable approach to analyze engine processing speeds, measuring various performance numbers to isolate major factors that impact performance within a given deployment. To verify accuracy and ensure repeatable results, all tests were performed 50 times, with the averaged data reported as the final result.

A key aspect of the test methodology is repeatability. The ability to run the same test over and over, isolating a specific variable, is necessary to analyze the impact that variable has on performance. Importantly, repeatability also allows others to run the tests in their own environment using this study's results as benchmarks.

One aspect of repeatability is the set of XBRL taxonomies and instance documents used in this test. All of the sample XBRL instance documents were obtained from third parties such as XBRL-US and CEBS (Committee of European Banking Supervisors) and are freely available to others running these tests. They represent a range of different real-world examples with a few "worst case" stress test examples sprinkled in. Please refer to Appendix A for links to the taxonomies and documents used in this study.

Copyright © UBmatrix, Inc. 2008

Another way the methodology isolates variables is in the distinction between different processing engine functions. XBRL processing engines perform a variety of XBRL tasks. By isolating them in the tests, the study provided visibility into which functions impact performance under different conditions.

This study isolated the following processor functions:

- **Load** – The time needed by the processor to load all or part of the XBRL instance document and taxonomy extensions (as applies) into memory before other tasks could begin.

- **Validate** –This is the key function of the processing engine used in scenarios where scalability is important. It represents the time needed for the engine to perform XBRL validation and output any error or warning messages.

- **Calculate** – A feature of some engines. It represents the time it takes for the engine to trace the calculation linkbase and compare the calculated results to fact values presented in the instance document.

- **Render** – A commonly-used function of the engine is the rendering of instance document data into another output format. The UBmatrix engine offers XSLT as it's rendering method. Given the end less possibilities of what XSLT can do, from displaying an HTML view of the document in a browser to preparing data to be inserted into a database, the tests performed a simple HTML rendering script.

- **Close** – Once all requested tasks are performed, the engine must free its cache memory and ready itself for the next instance document. This process is referred to as "close" and measures its resource impact on performance.

Testing Platform

The tests were run on a modest hardware platform using a 2-CPU machine with 2GB of RAM. Organizations dedicate significantly more horsepower for a simulated or real production environment. However, testing in this lesser environment serves to magnify the strengths and weaknesses of the engine being tested and offers a conservative benchmark for comparison. In addition, the components of this test platform are readily available versus a more typical 4- or 8-CPU machine. This makes the tests easily repeatable.

Testing Platform Specifications

Platform:
Red Hat Linux

CPU:
2 CPU (Intel 2.4 GHz duo)

Java Version:
Sun Java 1.5

RAM:
2 GB

XBRL Processor:
UBmatrix Processing Engine 3.1.3

Copyright © UBmatrix, Inc. 2008

Performance Testing Scenarios

The testing scenarios simulate real-world examples of instance documents for use cases common in the U.S. and Europe, and because they were readily available for others to test against. All XBRL taxonomies and instance documents used in this study are referenced in Appendix A and may be freely downloaded.

To help explain the nature of each test scenario, this study has broken out each instance document into its relative size, number of facts, contexts, and number of dimensional mappings. This offers the most transparency into the reported results and serves to ensure 'apples-to-apples' comparisons can be drawn. Figure A provides the details of each tested document.

Figure A – Instance Documents Tested

Test Case	Instance Doc Size (kb)	# Facts Reported	# Contexts Reported	# Dimensional Mappings Included	Comments
US-GAAP 07 Met Life	87	552	7	1	2007 US GAAP taxonomy. Sample instance from XBRL-US. Tested as a typical SEC submission.
US-GAAP basi	95	549	7	0	2005 US GAAP taxonomy. Tested as a typical SEC submission. Compare to new US GAAP
US-GAAP mda	62	373	8	0	2005 US GAAP taxonomy. Tested as a typical SEC submission. Compare to new US GAAP
US-GAAP ci	59	245	28	0	2005 US GAAP taxonomy. Tested as a typical SEC submission. Compare to new US GAAP
IFRS (typical)	43	256	7	0	Tested as a typical IFRS submission
COREP 1.2.4 ca	58	177	47	90	Tested as a typical COREP submission
COREP 1.2.4 sec irb	56	235	9	639	Tested as a typical COREP submission
Stress Test Samples					
IFRS (max)	838	5055	3	0	Manufactured and tested to measure the impact of size (# facts) on engine performance. (This is not a "real world" sample.)
FINREP 1.3.0	691	2962	111	1550	Manufactured and tested to measure the impact of dimensions on engine performance. (This is not a "real world" sample.)

Note in each of the Stress Test documents the "over-sampled" factors are highlighted above.

While most of the document characteristics identified in Figure A are straightforward, “Dimensional Mappings” warrants further explanation. More and more XBRL taxonomies are using dimensions to classify different “slices” of the same concept. For example, US GAAP offers a concept for tracking Services Revenue. However, there is an expectation that Services Revenue for “continuing operations” and that for “discontinued operations” are very distinct. Rather than tracking these as separate concepts, US GAAP accounts for them as separate dimensions of the same Services Revenue concept.

In order to validate an instance document that has dimensional tags, a processing engine must determine whether the taxonomy allows the concept to participate in the specified dimension. This extra step adds time to the validation of a given fact (measured in microseconds of course.)

The reason that the number of dimensional mappings is illustrated in Figure A is to provide additional transparency to performance numbers for those test cases with dimensions and those without. It also helps explain why the “FINREP 1.3.0” use case, which contains 2000 fewer concepts than “IFRS (max)” takes approximately the same amount of time to validate.

For real-world performance planning, dimensionality must be taken into account when designing and configuring an XBRL processing engine and the server upon which it runs.

Results Analysis for UBmatrix Processing Engine 3.1

Summary results for the tests of UBmatrix Processing Engine 3.1 demonstrate that the product offers enterprise-class scalability. Considering the scale of XBRL processing and validation that would be needed in each of the test cases in the study, this processing engine appears ready to handle the load.

The performance test results have been split into five separate categories for comparison:

1. **Documents per hour** – This analysis offers the most real-world view of a processing engine's performance. Say for example that a user of the processing engine receives 20,000 report submissions per quarter, with half submitting on the first day of the new quarter (10,000), and 10% of them submitting between 4-5pm Eastern. That means the processing engine would need to handle a peak of 1,000 documents per hour. This may not be a realistic example but illustrates an important measure of an engine's performance.

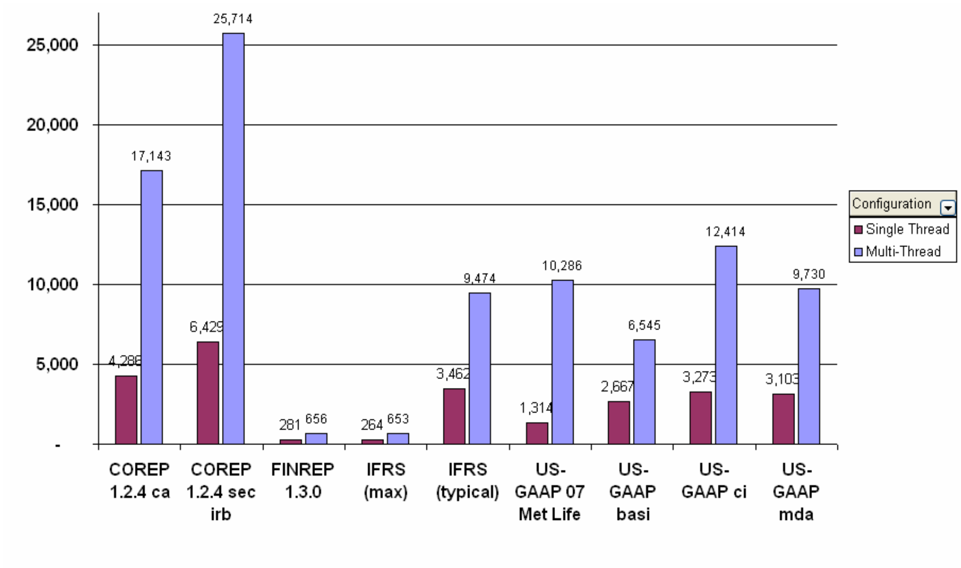
2. **Validation time per document** – This measure offers the best view of the impact taxonomy and document density/complexity affect engine throughput. Regardless how fast one engine is versus another, performance can be managed through the content being processed and by the engine functions used.

3. **Validation time per fact** – This is where the actual speed of the processing engine is most evident. Down at the fact level (assuming there are no memory/caching issues), it matters little how big the instance document or taxonomy is. It is reasonable to expect an engine to operate within a tight standard deviation between use cases.

4. **Single- vs. Multi-threaded CPUs** – Due to the type of tasks performed by XBRL processing engines, the study found that the UBmatrix Processing Engine benefits greatly from the use of multithreading. The study includes performance comparisons to illustrate how much of a performance gain can be found from this one tuning option.

5. **Processor functions used** – Another important factor to consider when evaluating engine performance is the amount of time required to complete each of the different steps performed by the engine. With visibility into each step it is possible to verify statistics. It's also useful in the tuning process, making it easier to determine how the engine can be optimized to perform only the functions needed.

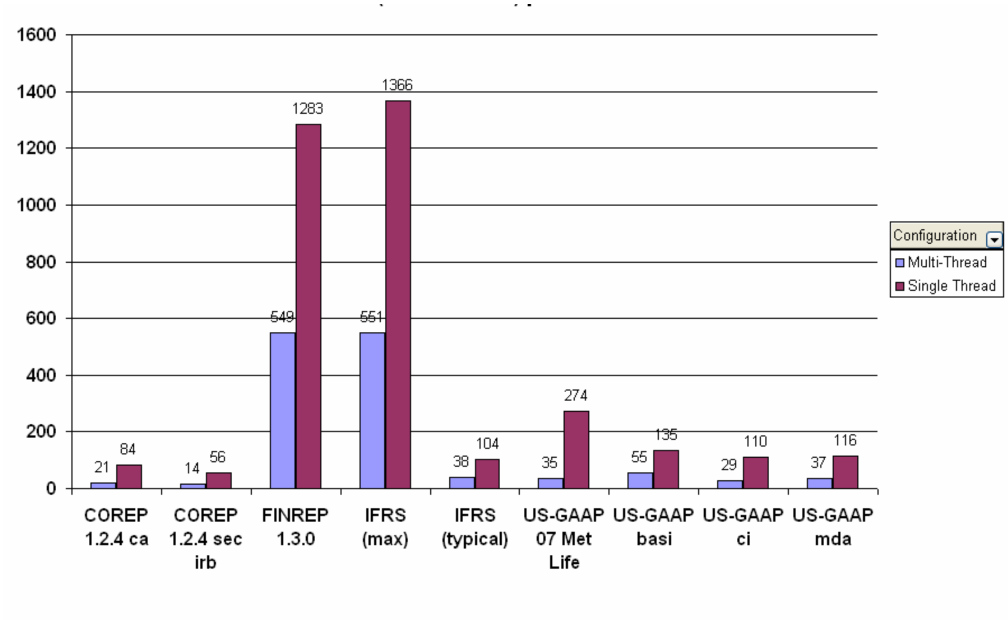
Figure B – Validated Documents per Hour



Predictably, the less complex documents results in significantly higher number of validated documents per hour. The highest performance is seen in the multi-threaded configuration of COREP 1.2.4 sec irb, which is a relatively small 56k while reporting 235 facts and 9 contexts, and includes 639 dimensions. This document's result of 25,714 validations per hour is extremely high when compared with the IFRS (max) document that, even using multi-threading to improve performance, validated 653 documents per hour. Clearly the size and complexity of the document (838k and reporting 5,055 facts and 3 contexts, and including 0 dimensions) had a highly significant impact on processing speed.

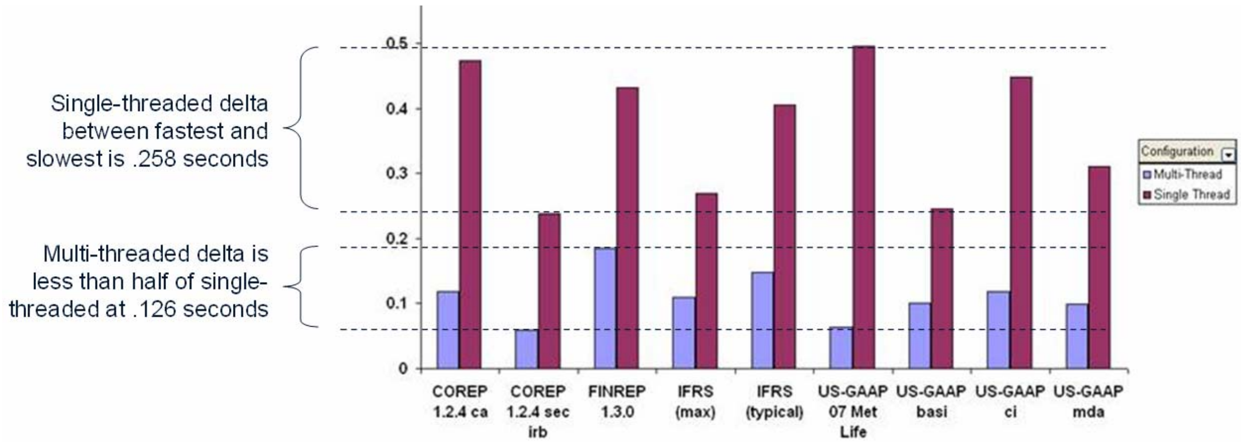
More typical processing results were seen with the US-GAAP and typical IFRS samples. For instance, the MetLife document from XBRL-US (new US-GAAP taxonomy) demonstrated scaling to over 10,000 documents per hour.

Figure C- Validate Time (milliseconds) per Document



To further break down performance, data was analyzed looking at validation time by milliseconds per each single document, as well as validation time on a per-fact basis. The per document times illustrated in Figure C demonstrate the speed at which typical real-world scenarios can be loaded into the processing engine and validated. The result was less than 60 milliseconds per document.

Figure D – Validate Time (milliseconds) per Fact



The per fact breakdown in Figure D makes it easier to evaluate and understand performance even when looking at processing speeds for documents that vary significantly in terms of size and complexity. Viewing validation times on a per-fact basis, it is clear that UBmatrix Processing Engine performance is consistent between taxonomies of similar complexity. Even considering the difference in size and complexity of the tested documents, the actual delta between the fastest and slowest documents was relatively minimal (.258 seconds for single threaded and .126 seconds for multi-threaded).

This level of analysis also demonstrates that while the complex IFRS (max) document took significantly longer to process as a whole, the per-fact data within the document was actually faster to process than the smaller, less complex documents. This is a critical differentiation when seeking a true 'apples-to-apples' comparison of processing engines.

Optimized XBRL processing using multi-threading

Due to the wide variety of preferred hardware and software environments, the testing focused on performance factors specifically managed by the XBRL processing engine. However, one environmental configuration offered such dramatically improved performance that it had to be address in the analysis.

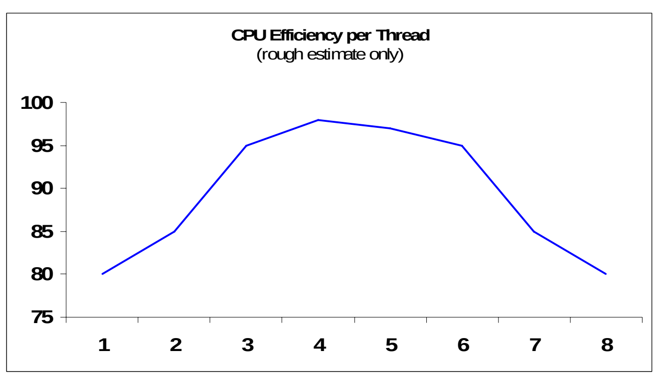
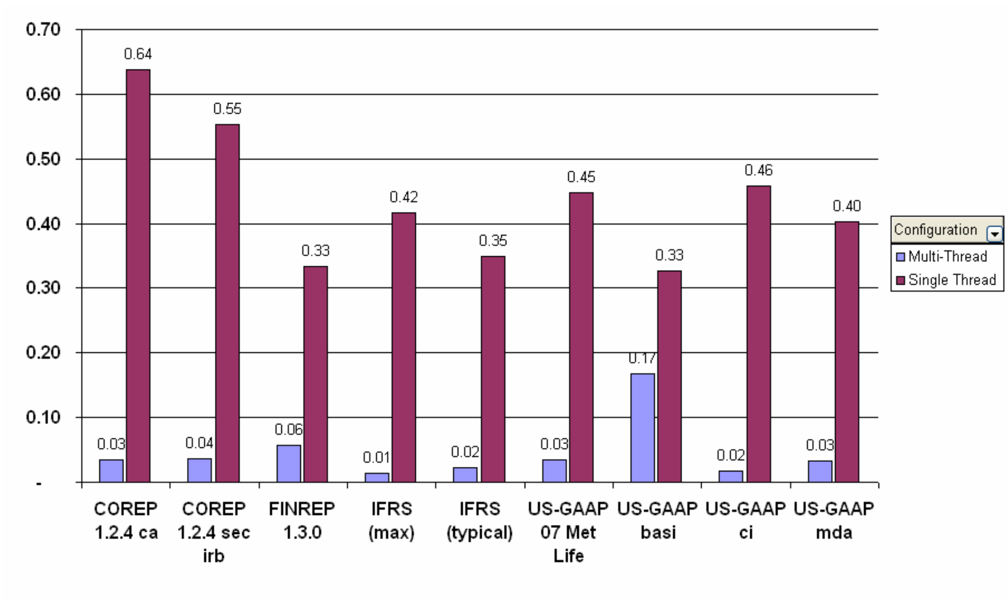


Figure E

The results of the testing call out the significance of multi-threading on document processing time. During the tests, various numbers of threads were tried to understand the performance impact. As illustrated in Figure E, this study found the optimal setting for the environment was between 3 and 5 threads per CPU. Our analysis was based upon very rough estimates (thus the range of threads rather than a specific recommendation). In a real production environment this and other server tuning metrics should be examined in more detail to ensure the most efficient use of resources.

In order to illustrate how much of an impact multithreading has, in Figure F, Document Load Time was isolated as one of the factors most affected by multithreading. The chart details the results of tests using four threads per processor. The results are quite dramatic and accrue to the overall performance metrics throughout the tests.

Figure F – Document Load Time (milliseconds) per KB



Conclusion

Testing and tuning the performance of an XBRL processing engine can be a challenge. This performance analysis study has provided a logical approach to measuring engine performance and offered results from UBmatrix Processing Engine 3.1.3 as an example. By providing the testing results as well as the methodology for arriving at those statistics, the goal is to provide an 'apples-to-apples' methodology for comparing the performance of other similar engines.

The results of the testing demonstrate the high level of scalability and speed offered by the UBmatrix Processing Engine, even on a modest platform of just two processors and 2GB RAM. UBmatrix Processing Engine has a strong pedigree of successful customer implementations and been designed with scalability in mind, even for the largest of enterprises.

For more information on the UBmatrix Processing Engine or other UBmatrix solutions, visit www.ubmatrix.com. For inquiries, please send an email to info@ubmatrix.com.

KEY DEFINITIONS:

- **XBRL instance documents** are business reports in an electronic format created according to the rules of XBRL. These documents contain facts that are defined by the elements in the taxonomy they refer to, together with their values, units, and an explanation of the context in which they are placed.
- **XSLT (Extensible Stylesheet Language Transformations)** is an XML-based language used for the transformation of XML documents into other XML or "human-readable" documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document may be serialized (output) by the processor in standard XML syntax or in another format, such as HTML or plain text.

Appendix A – XBRL documents used in this study

Test Case	Taxonomy Schema Source / File Name	Instance Document Source / File Name
US-GAAP 07 Met Life	▪ http://xbrl.us/USGAAPreview/Samples/Pages/default.aspx ▪ met-20061231.xsd	▪ http://xbrl.us/USGAAPreview/Samples/Pages/default.aspx ▪ met-20061231XML.xml
US-GAAP basi	▪ http://www.xbrl.org/us/fr/gaap/basi/2005-02-28/us-gaap-basi%20Summary%20Page.htm ▪ basi/Suntrust/sti-20050314.xsd	▪ http://www.xbrl.org/us/fr/IDs/Suntrust.zip ▪ sti-20050314.xml
US-GAAP mda	▪ http://www.xbrl.org/us/fr/rpt/mda/2005-02-28/usfr-mda%20Summary%20Page.htm ▪ mda/MetLife /met-20050228.xsd	▪ http://www.xbrl.org/us/fr/IDs/MetLife.zip ▪ met-20040930.xml
US-GAAP ci	▪ http://www.xbrl.org/us/fr/gaap/ci/2005-02-28/us-gaap-ci%20Summary%20Page.htm ▪ ci/Bowne and Co., Inc/2004-09-30/bne-20050204.xsd	▪ http://www.xbrl.org/us/fr/IDs/Bowne%20and%20Co.,%20Inc.zip ▪ bne-20040930.xml
IFRS (typical)	▪ http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/samples.htm ▪ SampleCompany-Taxonomy.xsd	▪ http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/SampleCompany-2005-05-15.xml ▪ SampleCompany-2005-05-15.xml
COREP 1.2.4 ca	▪ http://www.corep.info/corepTaxonomy/corep_taxonomy_documentation.htm ▪ t-ca-2006-07-01.xsd	▪ http://www.corep.info/corepTaxonomy/COREPInstances1.2.4.zip ▪ CA_instance.xml
COREP 1.2.4 sec irb	▪ http://www.corep.info/corepTaxonomy/corep_taxonomy_documentation.htm ▪ t-si-2006-07-01.xsd	▪ http://www.corep.info/corepTaxonomy/COREPInstances1.2.4.zip ▪ CR_SEC_IRB_instance.xml
IFRS (max)	▪ http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/samples.htm ▪ Proof-ifrs-gp-2005-05-15.xsd	▪ http://xbrl.iasb.org/int/fr/ifrs/gp/2005-05-15/Proof-ifrs-gp-2005-05-15-instance.xml ▪ Proof-ifrs-gp-2005-05-15-instance.xml
FINREP 1.3.0	▪ http://www.corep.info/finrepTaxonomy/finrep_taxonomy_documentation.htm ▪ t-FINREP-2008-01-01.xsd	▪ http://www.finrep.info/taxonomy/SampleInstancev1.3.zip ▪ SampleInstancev1.3.xbrl

Appendix B – Detailed Performance Measurements

As mentioned above, the study included detailed measurements of the XBRL processing engine's specific tasks. By breaking out processing times for load, validation, (together representing the "validate-only" use case), XSLT transform, and validate/transform/render (all functions performed), it is possible to make adjustments to the system to achieve greater performance. Figure G details the time required in the tests for UBmatrix Processing Engine to perform each of the given functions for each unique document type.

Test Case	Configuration	Validate-Only Docs per Hour	Validate & Transform/Render Docs per Hour	Load Time per KB (milliseconds)	Validate Time per Fact (milliseconds)
US-GAAP 07 Met Life	Single Thread	1,314	837	0.448	0.496
	Multi-Thread	10,286	3,364	0.034	0.063
US-GAAP basi	Single Thread	2,667	1,846	0.326	0.246
	Multi-Thread	6,545	3,303	0.168	0.100
US-GAAP mda	Single Thread	3,103	1,837	0.403	0.311
	Multi-Thread	9,730	3,429	0.032	0.099
US-GAAP ci	Single Thread	3,273	2,466	0.458	0.449
	Multi-Thread	12,414	5,455	0.017	0.118
IFRS (typical)	Single Thread	3,462	1,915	0.349	0.406
	Multi-Thread	9,474	3,273	0.023	0.148
IFRS (max)	Single Thread	264	221	0.416	0.270
	Multi-Thread	653	377	0.013	0.109
COREP 1.2.4 ca	Single Thread	4,286	2,880	0.638	0.475
	Multi-Thread	17,143	5,902	0.034	0.119
COREP 1.2.4 sec irb	Single Thread	6,429	3,273	0.554	0.238
	Multi-Thread	25,714	5,714	0.036	0.060
FINREP 1.3.0	Single Thread	281	266	0.333	0.433
	Multi-Thread	656	526	0.058	0.185

Copyright © UBmatrix, Inc. 2008

UBmatrix, Inc.
333 Twin Dolphin Drive
Redwood City, CA 94065
USA
Tel: +1.650.264.4510
Fax: +1.650.264.4515
E-mail: info@ubmatrix.com

April 2008