



IBM Atlas Extensions 6.0.3.2 Performance and Scalability

March 2016

© Copyright IBM Corporation 2016

Enterprise Content Management

www.ibm.com

Do not reproduce any part of this document in any form by any means without prior written authorization of IBM.

This document is provided “as is” without warranty of any kind, either expressed or implied. This statement includes but is not limited to the implied warranty of merchantability or fitness for a particular purpose. This document is intended for informational purposes only. It could include technical inaccuracies or typographical errors. The information in this paper, and any conclusions that are drawn from it, are subject to change without notice. Many factors contributed to the results described in this paper, and IBM does not guarantee comparable results. Performance numbers will vary greatly depending upon system configuration. All data in this document pertains only to the specific test configuration and specific releases of the software described.



1. Introduction

Many of the IBM Atlas Policy Suite features depend on a replication of the company's business hierarchy, including lists of employees, department organization, and matter definitions. In most companies, this data is represented in an existing database. Atlas Extensions provides a way to automatically import people, organization, and matter information from an external source. It is without the use of an intermediate storage medium, with changes to the original source automatically propagated to Atlas.

This paper presents the scalability characteristics and performance tuning guidelines for the Atlas Extensions 6.0.3.2 HRLoader and OULoader on WebSphere Application Server and DB2.

- HRLoader tests demonstrate linear scalability as the number of imported people increases. The complexity of the organization tree will affect the loading performance. Optimal performance can be found by varying the number of HRLoader threads.
- OULoader tests show that performance is affected by the complexity of the organization tree structure. OULoader can use incremental mode to upload delta changes to an organization structure as it changes over time, and demonstrates linear scalability as the amount of updated data increases.
- HRLoader can finish importing 1 million people in less than 4 hours. With best tuning on the lab's environment, it took less than 14 minutes to import 100 thousand people. OULoader can finish importing 100 thousand people with depth tree structure around 1 hour.

The performance results that are reported in this paper represent data and workloads that are run on an isolated network on specific operating environments and system configurations. Actual performance in real customer environments with production workloads might vary depending on many factors such as system configurations, workload characteristics, and data volume. The presented results here are not guaranteed to be repeatable in other systems.



2. Test Environment Description

2.1. System and Architectural Overview

IBM Atlas Policy Extensions provides two methods for importing information into the Atlas system: push and pull. The push method uses SQL to load the data to the Atlas database, while the pull method maps existing tables and columns into the configuration that Atlas expects. As you add data to your native system, it will automatically appear in the Atlas database. The tests that are described in this paper used the push method.

The IBM Atlas Suite database contains three default source tables, which were shown in Figure 1.

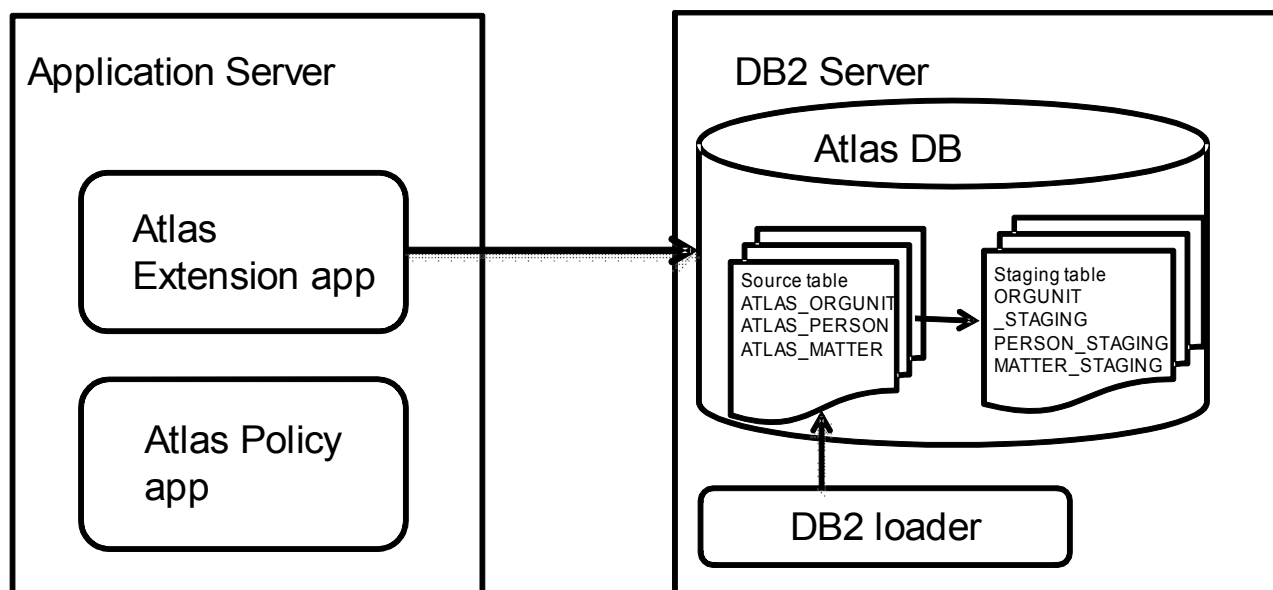


Figure 1. System and Architectural map

Component	Hardware	Software
Policy Atlas and Extension server	Intel Xeon 2.4 GHz (four processors), 8G RAM	Windows server 2012 standard 64 bit, IBM WebSphere 8.5.5.5, Atlas 6.0.3.2
Database server	Intel Xeon 2.4 GHz (four processors), 8G RAM	Windows server 2012 standard 64 bit, DB2 10.5.3



2.2. Atlas Application Server Configuration

WebSphere Application Server performance tuning was applied based on the Atlas Suite installation guidelines in the product documentation.

Setting	Value
Total Transaction Lifetime Time-out	3600
Maximum Transaction Lifetime Time-out	3600
JVM Initial Heap Size	1024
JVM Maximum Heap Size	1024
Generic JVM Arguments	-XX:+UseParallelGC -XX:PermSize=128m -XX:MaxPermSize=256m
Data Source Connection Pool Maximum Connections	100
Data Source Connection Pool Minimum Connections	100
Data Source Statement Cache Size	250



3. Test Data

For testing different extremes of possible organization structures, three types of test data structures of organizations and people were created:

- “Flat tree”. The flat tree structure has a depth of two, having only the root node and child departments nodes. Each department has multiple people.
- “Width tree”. An evenly balanced, multi-level tree of departments and subdepartments. Each department (node) at each level has multiple people.
- “Depth tree”. An unbalanced, multi-level tree of departments. Only one department (node) at each level has all of the subdepartments at the next level of the tree. Each department at each level has multiple people.

The Atlas database was preinstalled with a flat tree that has 1 thousand organizations and 10K people.

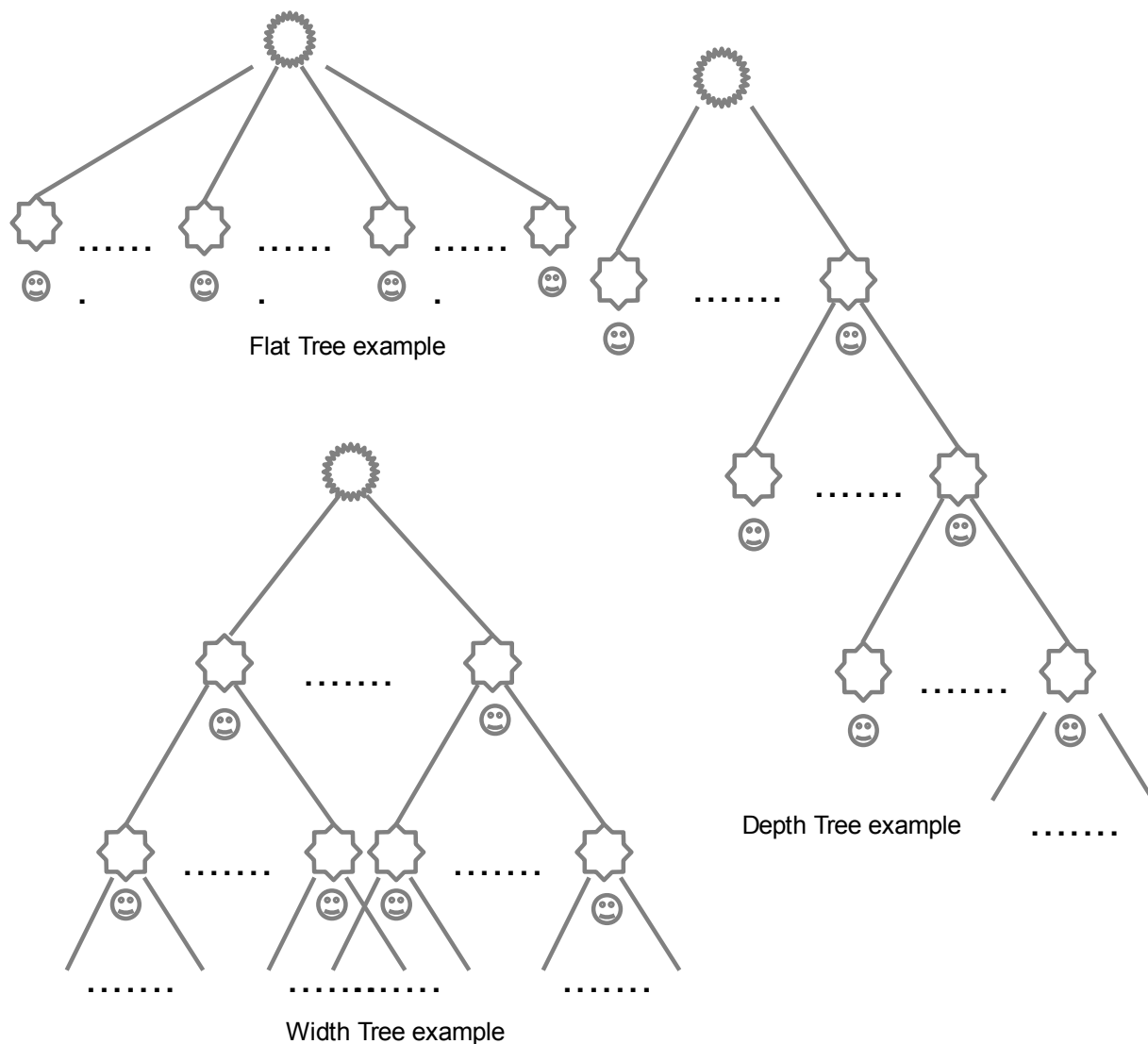


Figure 2. Test data structures



4. HRLoader

4.1. HRLoader Scalability

The scalability of HRLoader was demonstrated by importing people in both a flat tree and a width tree. The following parameters are configured for the Atlas Extensions server:

- HR_INCLUDES_ORGUNIT=no // This will not include organization unit ingestion when run HRloader
- ASSOCIATED_ORGNAMES_UPDATE_TASK_INCLUDED=no // This will prevent the system from trying to resolve organization associations

Figure 3 shows that both tests demonstrated good scalability, loading 1 million people in less than 4 hours. The performance was slightly better for the simpler structure of the flat tree.

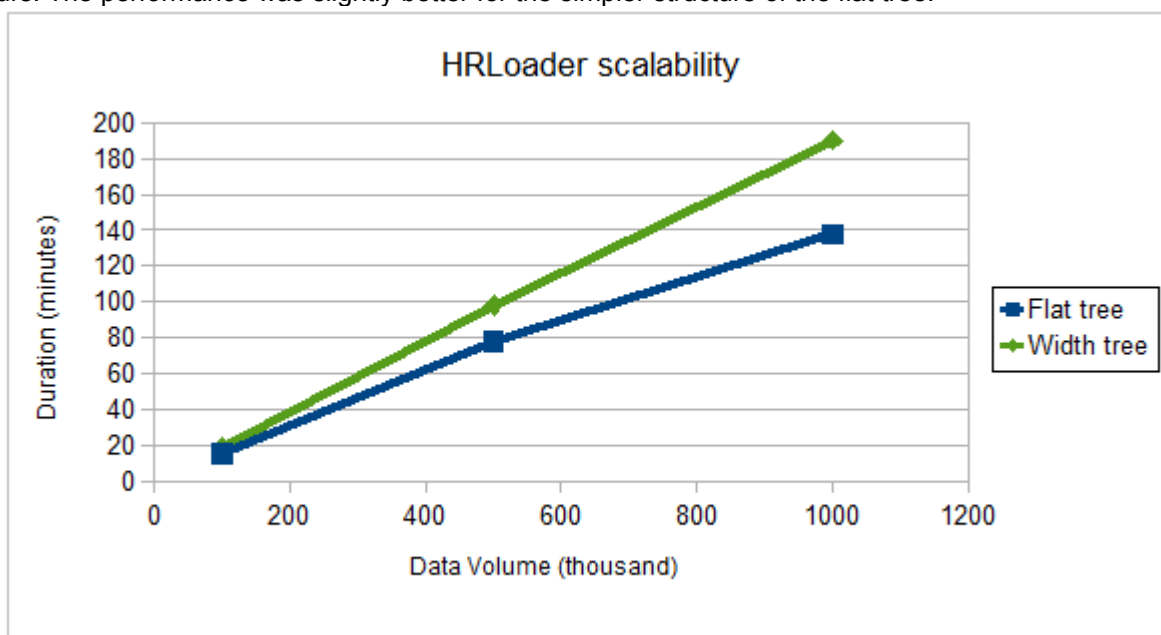


Figure 3. HRLoader scalability for flat and width trees

4.2. Performance impact by varying the number of loading threads

For testing the IBM Atlas 6.0.3.2 Extensions HRLoader, the following parameters were configured:

- HR_INCLUDES_ORGUNIT=no // This will not include organization unit ingestion when running HRloader
- ASSOCIATED_ORGNAMES_UPDATE_TASK_INCLUDED=no // This will prevent the system from trying to resolve organization associations
- The number of loader threads varied 5 and 50

Figure 4 shows the duration for loading 100 thousand people for varying numbers of threads, and Figure 5 shows the same for loading 1 million people. The number of threads had little impact on the total duration beyond 10, which is not surprising on servers with four CPUs.

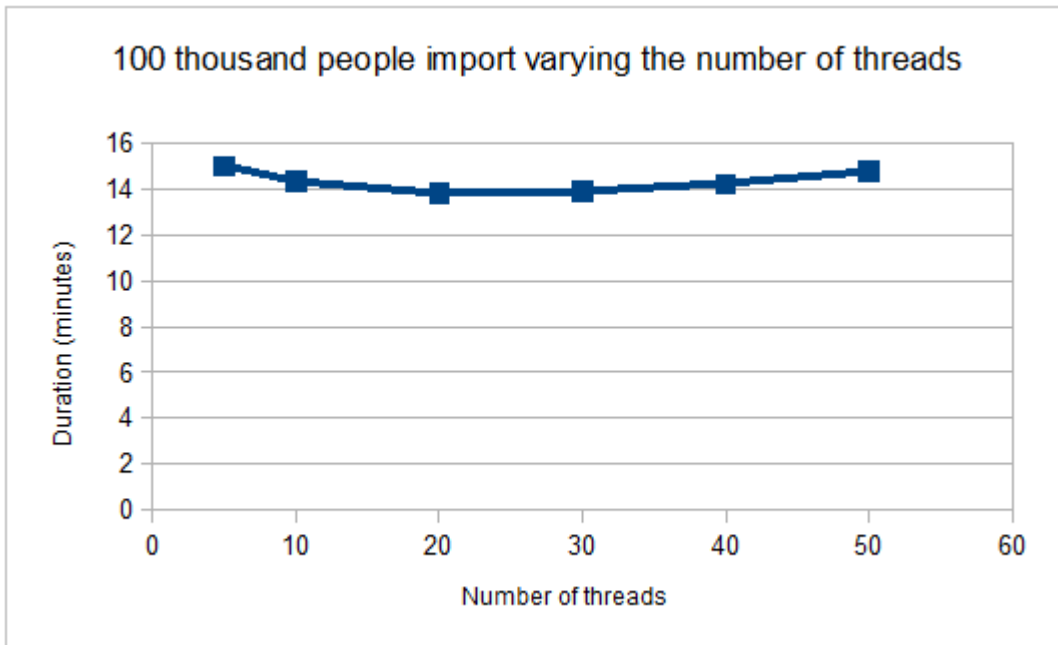


Figure 4. HRLoader tuning of thread for 100 thousand people

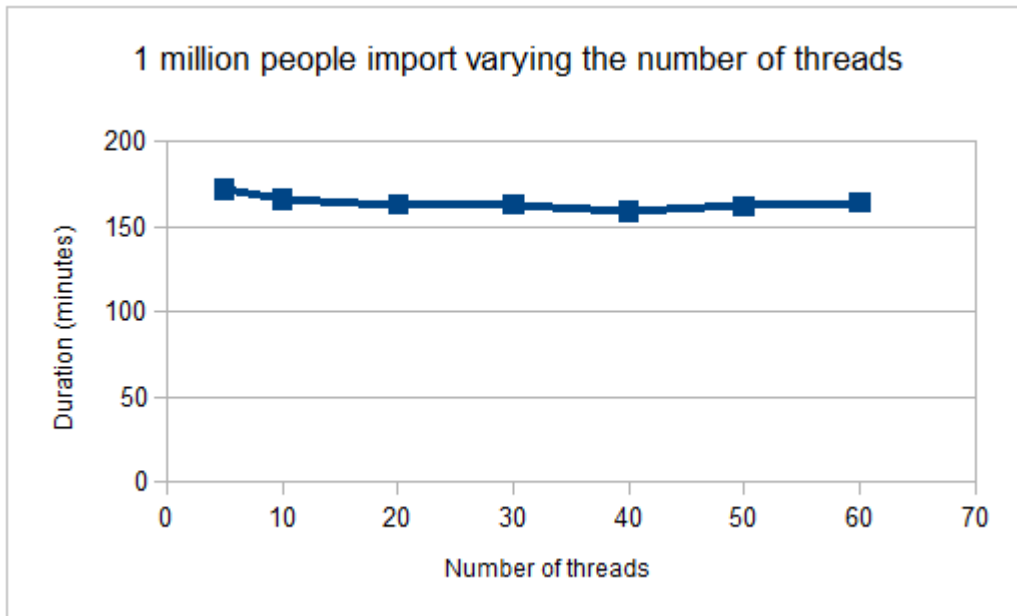


Figure 5. HRLoader tuning of thread for 1 million people



5. OULoader

5.1. The Scalability of OULoader

The scalability of IBM Atlas Extensions OULoader was demonstrated by importing company organizations into the Atlas system. The parameter “integration mode” was set to “full” in OrgUnitDBPullerAgent.xml, in the \AtlasHome\Properties\defaultimpl\db directory.

Figures 6 and 7 show how loading performance depends on the tree type and depth. For loading 100 thousand organizations, the flat tree took the least time and the depth tree took the longest time to load. Similarly, the load time increases as the tree depth increases.

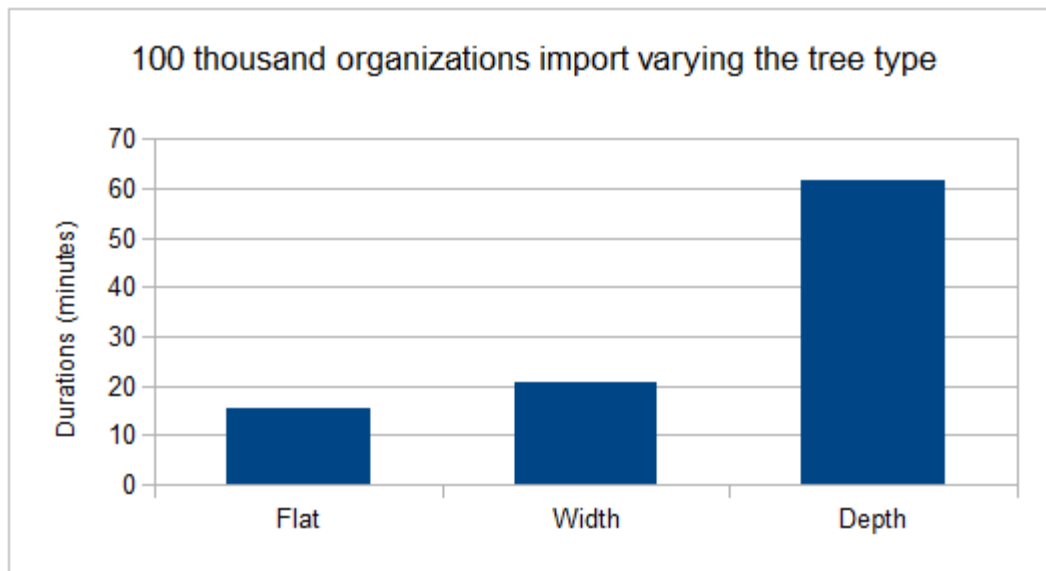


Figure 6. OULoader scalability for three types of tree structure

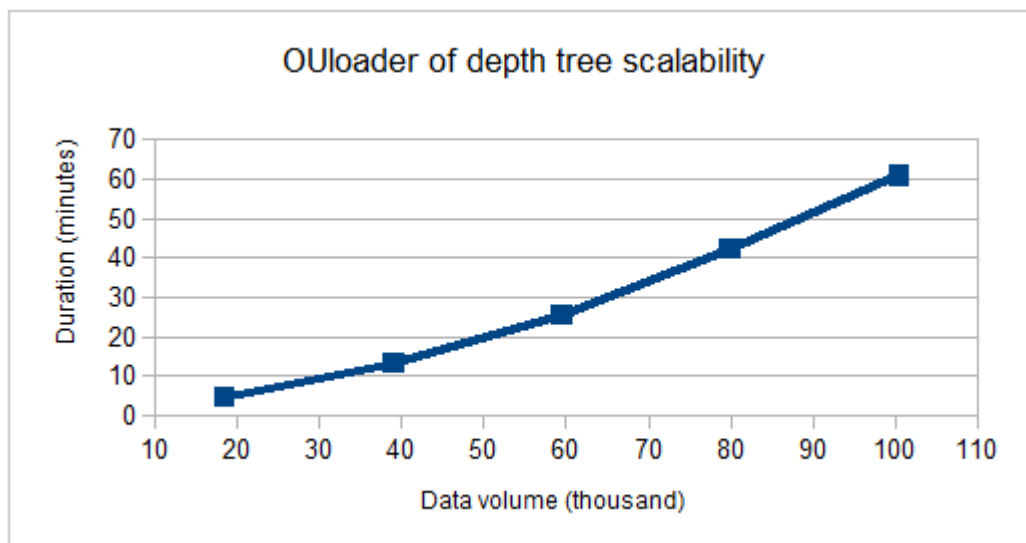


Figure 7. OULoader scalability for depth tree



5.2. The full/incremental mode of OULoader

IBM Atlas Extensions can be configured to automatically run at set intervals in an incremental mode. It can be beneficial when a company's HR or organizations are large and frequently updated. This test is the scenario to test the incremental mode performance and scalability of OULoader.

For these tests, these parameters were configured:

- "ENABLE_CYCLIC_ORG_CHECK=No" // Set in the file integration.properties in the AtlasHome\Properties directory, and skips the validity check of whether the organization structure is cyclic.
- "<integration mode='incremental'>" // Set in the file OrgUnitDBPullerAgent.xml in the AtlasHome\Properties\defaultimp\ldb directory
- Disabled all time tasks to ensure that each test ran in isolation

The Atlas database was preinstalled with 100 thousand organizations in a depth tree of 101 levels and 32 nodes. Figure 8 shows the linear performance of updating varying subsets of the 100 thousand organizations, demonstrating the value of incremental mode.

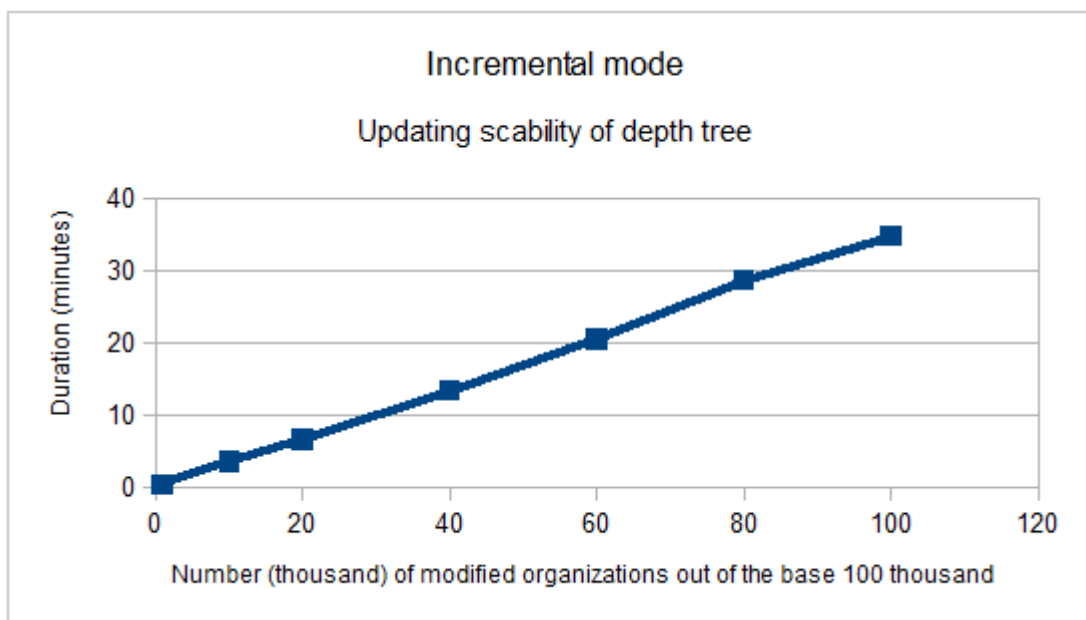


Figure 8. OULoader incremental mode performance



Authors

Xue Mei Deng, IBM Atlas SVT team
Angela Dong, IBM Atlas SVT team
Lu Huang, IBM Atlas SVT team
Yun Xin Yin, IBM Atlas SVT team

Thanks to these contributors who helped improve this paper

Dave Royer, IBM ECM Performance Engineering Architect
Xin Wang, Manager, IBM Atlas SVT
Parthasarthi Komperla, IBM Atlas Development
Mayank Shah, IBM Atlas Development
Syed Moinudeen/India/IBM, IBM Atlas Development
Nazrul Islam, IBM Atlas Director of Engineering
Christina Gonzalez, ILG Box content design team
Tori L Nemethy, ILG Box content design team