ADABAS STRESS TESTS ON THE MAINFRAME
LEVERAGING IBM® INNOVATIONS FOR INCREASED SCALABILITY AND THROUGHPUT

By Arno Zude and Nate Sathre, R&D, Software AG

Software AG is a member of zSeries® Partner in Development and actively adopts IBM® mainframe hardware and software innovations. This past November, Software AG tested the latest version of Adabas and its add-on products on new IBM mainframe hardware and operating system software at the IBM System z® Benchmark Center in Poughkeepsie, NY. Recommendations on how to leverage IBM’s zHPF™, Flash Express™ and Large Pages with Adabas for increased scalability and throughput are discussed below. To dive into the detailed test results, please read the white paper at www.softwareag.com/adabasperformance.

TEST OBJECTIVES
The test database and workload were designed to align with the well-known TPC-C OLTP benchmark. Multiple scenarios with Adabas and various add-on products set up on IBM’s latest mainframe hardware and system software were established to test for optimal performance and scalability. The objectives of the tests were to determine the following:

• Scalability of Adabas Cluster Services and Adabas Parallel Services
• Maximum throughput achieved with Adabas Parallel Services
• Relative performance of Adabas with and without IBM zHPF
• Relative performance of Adabas with and without large pages
• Impact of paging with IBM Flash Express on Adabas performance
• Relative performance of Entire Net-Work with different line drivers

SCALABILITY
Adabas Cluster Services scaled well up to 12 nuclei in a cluster, increasing the external throughput (number of Adabas commands per elapsed wall clock second) as nuclei were added. Beyond that cluster size, the external throughput flattened out, probably because the Adabas Cluster Services nuclei were limited by the many read I/Os from ASSO and DATA in this test.

Adabas Parallel Services scaled well up to 24 nuclei in the cluster, increasing the external throughput as nuclei were added as shown in Figure 1. Beyond that cluster size, the external throughput flattened out, probably because the Adabas Parallel Services nuclei were limited by the many database I/Os in this test.

For both Adabas Cluster Services and Adabas Parallel Services, as additional nuclei were added to the cluster, the average CPU consumption per command increased only slightly per nucleus added. In fact, when a small cluster was increased, the average CPU consumption at first decreased (with a low at four nuclei with Adabas Cluster Services and 20 nuclei with Adabas Parallel Services), then it increased by about 1 percent or less per nucleus added.

Recommendations: Put an Adabas Cluster Services nucleus on every system within the Parallel Sysplex that hosts users who issue a significant number of Adabas calls. The better performance gained from users making local calls to the Adabas Cluster Services nuclei outweighs the small performance cost of increasing the cluster.
If the update load is very high, tune the Adabas buffer flush and the I/O subsystem to achieve that updated blocks are written out to the database as fast as the updates are performed in memory. Avoid that the cache structure runs full with updated blocks. Structure-full events for Adabas Cluster Services reduce the external throughput. Use as many Adabas Parallel Services nuclei in the cluster as needed to keep each individual nucleus from becoming a bottleneck, whether by consuming up to 100 percent of one CPU or by being limited in its WORK/PLOG writes or by something else. Adabas Parallel Services clusters scale well. If individual nuclei are resource-constrained, adding another one may relieve the constraint, up to at least 20 nuclei.

**MAXIMUM THROUGHPUT**

A number of tests were performed to explore which configuration of Adabas Parallel Services achieved the highest External Throughput Rate (ETR) as shown in Table 1. The highest external throughput seen in the standard update test, with ~5.2 percent update/insert commands and ~5.9 percent commit/backout commands, was 1,140,272 Adabas commands per second elapsed (wall clock) time.

The highest external throughput seen in the read-only test, with no update commands, was 1,463,546 Adabas commands per second elapsed (wall clock) time.

**Recommendations:** If a large amount of memory is available for buffer pool or cache space, allocate most of it to the shared Adabas Parallel Services cache and make the private buffer pools of the Adabas Parallel Services nuclei just large enough for their smooth operation. Configure the cache to store unchanged database blocks (i.e., set CLUCACHEUNCHANGED=YES). Sharing cached database blocks between the cluster nuclei is more efficient than keeping them multiple times in private caches/buffers.

**ZHPF GREATLY IMPROVES PERFORMANCE**

IBM’s System z High Performance FICON (ZHPF) provides a streamlined I/O architecture and channel program protocol to improve I/O latencies and throughput. Several tests were performed to explore the effects of turning ZHPF on or off. We tested if potential improvements in I/O performance translated to performance gains for Adabas applications.

Adabas benefited significantly from using ZHPF. It delivered faster I/Os (lower latencies) and higher I/O loads (throughput). A single, non-cluster Adabas nucleus achieved ~3-4 percent higher external throughput and lower average CPU time per Adabas command with ZHPF on an I/O-intensive workload. An Adabas Parallel Services cluster with 31 nuclei achieved ~14 percent higher external throughput and slightly higher CPU time per Adabas command on a very I/O-intensive workload.

The external throughput was lower with ZHPF than without when the WORK and PLOG I/Os were in the same order of magnitude as the ASSO and DATA I/Os. This may be due to a subtle interaction between the better I/O performance with ZHPF and the Adabas logic for writing to the WORK and PLOG datasets.

**Recommendation:** We highly recommend using ZHPF, if it is supported by the operating system and I/O subsystem. Our tests have shown that Adabas achieves a significantly better I/O performance with ZHPF that is beneficial especially for I/O-intensive workloads.

Caveat: When running a large cluster (more than eight nuclei) with high buffer efficiency and an update-intensive application workload, watch the relation between WORK/PLOG writes and ASSO/DATA I/Os before and after the switch to ZHPF.

**LARGE PAGES MAY SLIGHTLY REDUCE CPU CONSUMPTION**

IBM z/OS 2 GB Large Pages, supported by Adabas Parallel Services and Adabas Caching Facility, may provide lower CPU consumption for large memory objects in long-running memory-intensive applications, such as a database server.

We measured the performance effects of using different size page frames in memory for the global, shared cache of Adabas Parallel Services. Theoretically, using larger page frames for memory areas that exhibit suitable usage patterns should reduce the number of virtual-to-real address translations by the processor because each address translation covers more memory.

---

### TABLE 1: Adabas Parallel Services Throughput Test Results

<table>
<thead>
<tr>
<th>Test ID</th>
<th>Cluster Nuclei</th>
<th>CluCache-Size</th>
<th>User Commands</th>
<th>Avg. TXG Runtime (minutes)</th>
<th>Total Adabas CPU Tme (minutes)</th>
<th>ETR</th>
<th>ITR</th>
</tr>
</thead>
<tbody>
<tr>
<td>D27PM</td>
<td>28</td>
<td>245G</td>
<td>509,983,708</td>
<td>7.50</td>
<td>156.44</td>
<td>1,140,272</td>
<td>54,333</td>
</tr>
<tr>
<td>D26P1*</td>
<td>31</td>
<td>260G</td>
<td>906,576,746</td>
<td>12.05</td>
<td>252.38</td>
<td>1,254,222</td>
<td>59,868</td>
</tr>
<tr>
<td>D27P3*</td>
<td>31</td>
<td>245G</td>
<td>906,490,607</td>
<td>10.46</td>
<td>248.78</td>
<td>1,463,546</td>
<td>60,728</td>
</tr>
</tbody>
</table>

*Note: In contrast to update test D27PM, D26P1 and D27P3 performed the read-only test application.*
Backing the large, shared Adabas Parallel Services cache with large pages (1 MB pages or 2 GB pages) reduced the average CPU consumption per command in the range of 3.7 to 6.8 percent, but did not demonstrate a clear improvement in the external throughput. The benefit from large pages may be smaller if there are fewer than the 31 nuclei in the cluster that we used in our tests.

**Recommendation:** The expectation of CPU savings should be weighed case by case based on application memory use, to ensure that the application is a good fit for the use of large pages.

**FLASH EXPRESS FASTER THAN DISK BUT NO PAGING IS BEST**

Flash Express exploits flash memory for paging as a new optional feature of the IBM zEnterprise® EC12 (zEC12) processor. The flash memory cards on the zEC12 processor come in pairs with each pair of cards providing 1.4 terabytes of storage.

We explored the performance effects of paging in a number of configurations using Adabas Parallel Services with Flash Express. Theoretically, paging may lead to a considerable drag on the external throughput, especially when Adabas executes many commands in parallel.

When we increased the large shared cache such that Adabas Parallel Services incurred paging to Flash Express, the external throughput increased, too. But paging to disk and severe paging to Flash Express caused the external throughput to drop precipitously. The impact of paging would probably be smaller with fewer parallel users. But we achieved the best external throughput with an Adabas configuration that did not incur paging.

**Recommendation:** For best performance, configure Adabas to avoid paging. Occasional light paging to Flash Express devices may be tolerable. However, reconfiguring Adabas to avoid paging promises the best results.

**ENTIRE NET-WORK PERFORMS BEST WITH FCTC LINE DRIVER**

Entire Net-Work transparently connects Software AG’s server products—including Adabas—with client programs (e.g., Adabas users) that are running on other systems (e.g., mainframe, UNIX® and Windows®). We explored the relative performance of different Entire Net-Work line drivers when the Adabas users were remote to Adabas.

Entire Net-Work performed best with the FCTC line driver of version 6.3 SP1, achieving higher external throughput and lower Net-Work CPU time per Adabas command than the CTCA driver and the FCTC driver of version 6.2 SP2. With the payload data generated by the test application, send and receive block sizes of 32 KB performed slightly better than 60 KB blocks.

The configurations with the TCP/IP and XCF line drivers did not perform as well, but they were not specifically tuned for best performance. It is possible that they would have fared better with other configuration settings.

**Recommendation:** For best performance, configure Entire Net-Work to use the FCTC line driver of version 6.3 SP1 (or later) for the primary communication link between nodes on co-located mainframe systems.

**CONCLUSION**

Having the opportunity to test the latest Software AG products with the latest IBM hardware and software provides tremendous benefit for both Software AG and IBM. The Software AG team conducting the tests received invaluable support from the IBM team responsible for the test system and we thank IBM for that partnership. To learn more about the test setup, results and conclusions, please visit www.softwareag.com/adabasperformance.

---

**ABOUT THE AUTHORS**

Arno Zude is an Adabas Mainframe Consultant at Software AG in Darmstadt. Since 1989, Arno has specialized in the Adabas database management system on mainframes. He contributed to the development of the Adabas core product as well as add-ons like Adabas Delta Save Facility, Adabas Cluster Services and Adabas Parallel Services.

Nate Sathre is Lead Quality Engineer for Adabas mainframe products at Software AG. With 15 years of experience with Adabas, Nate has played a role in the quality of nearly all of the Adabas products. Nate is also the developer of the ATA suite of tools and processes which are widely used in the testing of Adabas mainframe products.