

Erratum

in
Comparing the Performance of Object Databases and ORM Tools
Van Zyl, Kourie, Boake

A coding error has been identified that affects some of the results in our publication in the ACM Digital Library:

Van Zyl, P., Kourie, D.G. and Boake, A., Comparing the Performance of Object Databases and ORM Tools. *In Proceedings of SAICSIT 2006*, 1-11, Eds: J. Bishop and D Kourie, Somerset West, South Africa, October 2006. (<http://doi.acm.org/10.1145/1216262.1216263>)

Since the ACM Digital Library has given no indication of where to place an erratum, we take the liberty of placing it here.

The error affects sections 5.2, 5.3 and 5.4 of the article, as well as some of the conclusions. The article reported execution times for certain traversals, queries, inserts and deletes of the 007 benchmark, run on db4o and Hibernate, respectively. The tables below provide the original and revised comparative results. They are expressed as a percentage of time differences relative to the db4o times, i.e. as: $100 * (\text{Hibernate time} - \text{db4o time}) / \text{db4o time}$.

Traversal	Original	Corrected
T1	852 %	181%
T2a	81%	-81%
T2b	24%	-69%
T2c	39%	7%
T3a	22%	-63%
T3b	43%	-67%
T3c	55%	-3%
T6	2497%	143%
T8	-36%	200%
T9	-36%	*

* db4o time too small to record.

Queries	Original	Corrected
Q1	-36%	254%
Q2	969%	5786%
Q3	3759%	15176. %
Q4	118 %	4125%
Q5	6957%	149432%
Q7	1517%	3243%
Q8	3487%	5762%

Insert/Delete	Original	Corrected
Ins: Cold Run	79%	111%
Ins: Avg Hot Run	95%	39%
Del: Cold Run	13%	17%
Del: Avg Hot Run	40%	56%

Overall, Hibernate is faster in 5 out of 10 times, whereas in the original, it was faster in only 2 of the 10 traversals. Thus, our conclusion that db4o generally performs traversals faster than Hibernate has to be withdrawn. Indeed, Hibernate performs better in the larger collections, namely T2b, T3b and T3c.

The corrected data shows db4o outperforming Hibernate in all queries, and by a wider margin than before. Indeed, Hibernate no longer outperforms db4o in Q1. Finally, db4o is still faster than Hibernate for all insert and delete scenarios that were tested.

The authors are grateful to William Cook for drawing our attention to the unlikely nature of the original transaction data, and regret any inconvenience caused by the coding error.

Author Addresses:

P. van Zyl, ESPRESSO Research Group (<http://espresso.cs.up.ac.za>) , Department of Computer Science, University of Pretoria, Pretoria, 0002, South Africa; pvzyl@csir.co.za.