WaveAccess
We Make IT Easy

Algorithms and code performance optimization

Algorithm performance tuning projects appear when code complexity and data flow size cause issues in achieving results in a timely fashion. The most popular technologies that we tune codebase on: Java, R, Python, .Net, C++, and SQL.

Test Coverage

One major aspect of the WaveAccess process is to cover the code with a variety of tests – general, exceptions, extremes, etc. Our development team does not start to make any changes until it has made sure that our optimizations do not affect the algorithm. It is common for one algorithm to have more than 100 test data sets. The validation criteria are often a subject of discussion because some algorithms are discrete with strict input and output data, but there are cases with fractions number computing that have epsilons and mistake thresholds.

The tests are implemented as a unit test module and integrated into a continuous integration environment. The test environment is usually a pre-built server like a TeamCity that compiles every commit into a code repository and runs all the tests.
This approach helps us find issues at the early stages and to ensures that we can deliver a stable version at any time.

Performance Optimization Options

WaveAccess uses a rich set of options to increase the algorithms computing speed:

1) Change common sub algorithms to be more efficient

For example, changing some part with a sorting code to be a more efficient one – many common algorithms have disadvantages that can cause computing deterioration. Our job is to find the most stable and efficient algorithm.

2) Add caching

This popular approach finds places in the code that have a good hit cache rate and puts the cacheable data into memory, shared cache (Redis for example), or disk.

3) Utilize a CPU's fastest instructions

Processors evaluate enough to perform several instructions per cycle. Sometimes the computing hot spot can be changed to use different data structures and operations that are more native for a CPU and can be executed much faster.

4) Utilize memory better

The whole idea of this case is to use cache from the CPU L1-3 better. This means that the way code iterates on simple arrays (sequence or randomly) is important for the CPU memory cache hit rates.

5) Multi-threading and clustering

This is a way to horizontally scale code execution. Multi-threading also has its own specificity – coherent memory access, thread contention, etc.

6) Increase efficiency on code execution

Based on reverse engineering analysis, it is possible to find places that can be re-implemented better – code style, SQL queries, or layer architecture. After refactoring, the app will have improved SQL query execution plans, fewer IO operations/queries to the database, fewer code cycles, etc. As a result, a better execution speed is achieved.

7) Faster porting technology

When it appears that the current technology executes code slower than others, it is always possible to implement a small module on Java/C++ and seamlessly integrate it with R etc. keep-ing the global app interfaces the same.

More details in our article "R performance optimization using Java"
[link to http://www.wave-access.com/public_en/blog/2015/april/21/r-optimizations.aspx]

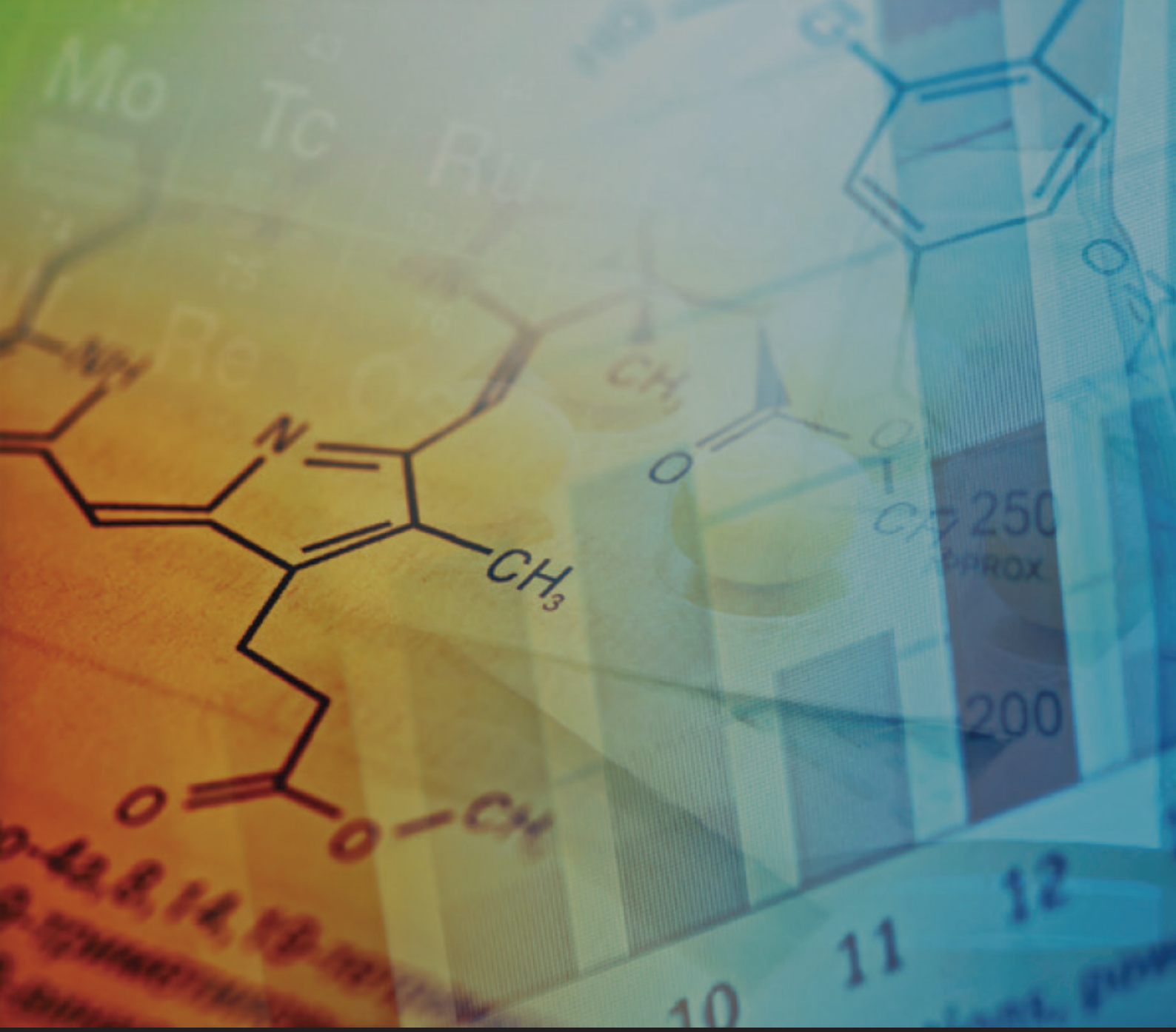
8) Use GPU acceleration

This radical approach can dramatically increase computing speed 6-100 times in some cases. The idea being that >200 weak cores on GPU are much faster the 4-16 core on CPU especially on highly parallelized algorithms with heavy and random memory access. The WaveAccess team uses CUDA SDK and OpenCL for GPU accelerated code implementation.

More details can be found in our article "Breakthrough in CUDA data compression"
[link to http://www.wave-access.com/public_en/blog/2011/april/22/breakthrough-in-cuda-data-compression.aspx]

The Result

Algorithm performance tuning by the WaveAccess team delivers computation results in a reasonable time and thereby drives and improves the efficiency of processes.



WaveAccess
We Make IT Easy

If you have a science intensive project for us, please get in touch

scientific@wave-access.com
skype: wave_access
+1 866 311-24-67