Software Algorithm Testing – Guidance

Summary

One of the three attributes for the Net-Ready KPP (NR-KPP) is that Information Technology (IT) must be able to support mission operations. For IT systems supporting operational mission threads, this means the mission threads must be executable within time periods that support the mission.

Each software system may be unique, but many computer software algorithm considerations are similar across the various systems. Software algorithms used for processing large amounts of data need to be efficient, incorporating industry best practices. This is especially important for fast searching, sorting, and merging of data files. Government testing, particularly during DT, may not look at actual data structure and algorithm coding within software modules. Instead, the software is considered a black box, with testing focused on input parameters, state variables, and results returned from the black box as well as the timeliness of receiving the outputs. The primary goal in looking at software algorithms during developmental testing is to ensure that industry best practices have been employed to ensure operational mission threads involving large data sets operate efficiently. Significant insights can be learned from focused testing in a DT controlled environment, even though the tester may not have direct access to the data structures or software code.

Algorithm performance testing should be considered during DT whenever large amounts of data are being manipulated, and the data processing time might be excessive to the point of potential mission impact.

Types of algorithms that may need performance testing

There are several types of algorithms that may need performance testing to try to ascertain whether the developer used industry best practices. Each of these categories of work needing to be performed can be categorized based on roughly how much longer the processing should take as the data set increases in size.

- Searching one or more large data sets to find data elements matching certain criteria, to include creation and execution of complex ad hoc data queries
- Sorting a large data set into a particular sorted order
- Merging two or more data sets, at least one of which is large, with resultant list possibly in some sorted order
- Optimization algorithms which seek to determine optimal routing of a delivery vehicle to visit multiple locations (for example, a optimizing a bomber route as it flies over or near multiple targets)

Industry best practices

The subject of combinatorial algorithms deals with the problems associated with performing fast computations on discrete data structures. Many types of algorithms can also be

found through simple internet searches, and Wikipedia will show the name of the algorithm and best case, average case, worst case, memory usage, and whether the algorithm is stable. http://en.wikipedia.org/wiki/Sorting_algorithm shows information for various sorting algorithms. Unless significant information is known about the data sets, industry best practices should generally use algorithms based on good average performance.

Big O notation characterizes functions such as the processing time according to their growth rates, usually providing an upper bound on the growth rate of the function. See http://en.wikipedia.org/wiki/Big_O_notation .

References

A New Approach for Delivering Information Technology Capabilities in the Department of Defense, Report to Congress, November 2010

CJCSI 6212.01F, 21 March 2012, Net Ready Key Performance Parameter (NR-KPP)

Examples

Software Algorithm Testing – Examples