

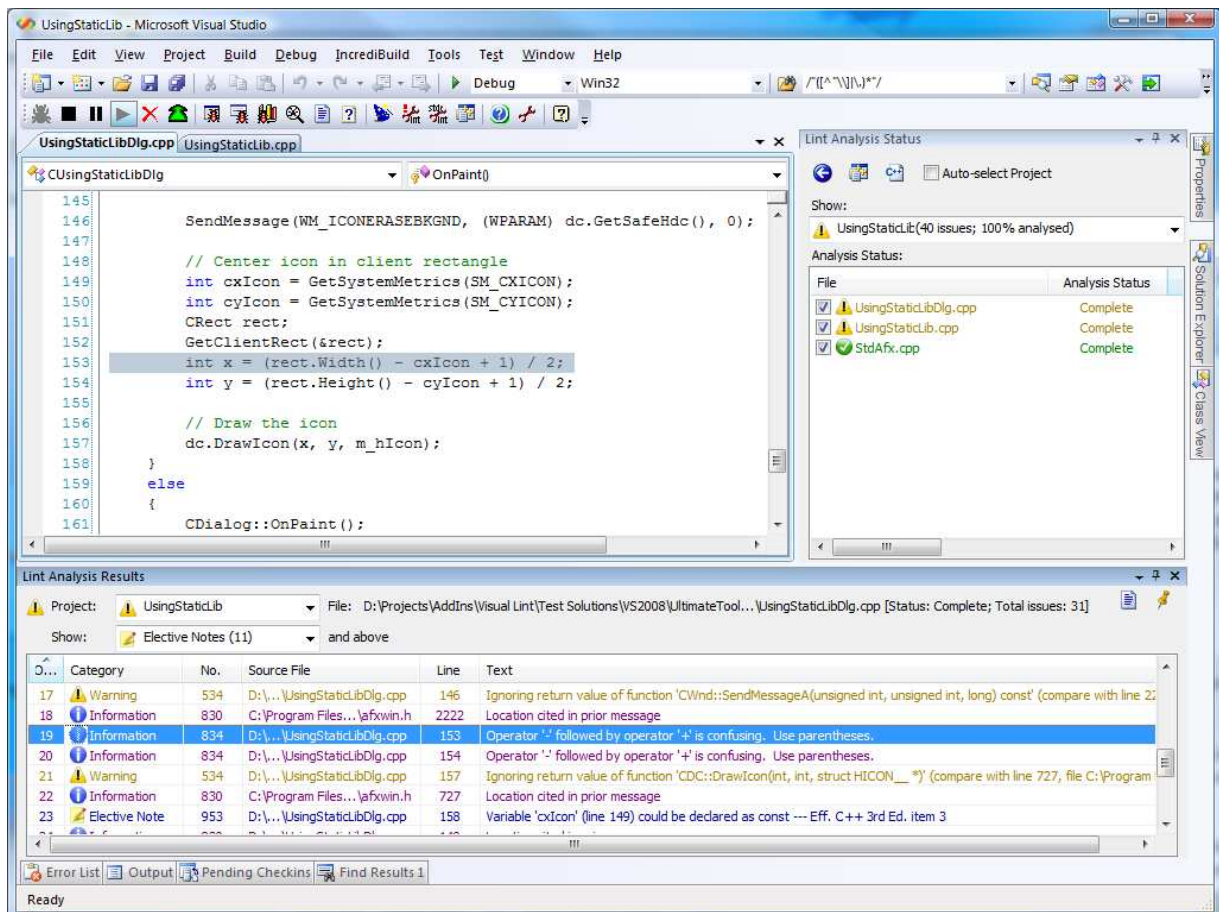
Abstract

This article describes how the IncrediBuild XGE interfaces were used by Riverblade to accelerate PC-lint C++ code analysis tasks in order to complete the static analysis of a Visual Studio solution in a fraction of the time possible with conventional methods.

PC-lint analyzes C++ source code, so analysis runs using it can be quite time consuming if the code being analyzed uses the Windows SDK header files or has significant internal dependencies. Riverblade's solution is to use the Xoreax IncrediBuild XGE Interfaces Extension Package to allow PC-lint analysis tasks to be delegated to spare development machines on the network. In doing so, speed improvements of up to 17x over conventional single threaded analysis approaches have been observed.

The Challenge

Riverblade offer Visual Studio integration and code analysis tools, of which the most well known is Visual Lint (http://www.riverblade.co.uk/products/visual_lint) – a product which integrates the Gimpel PC-lint C++ code analysis tool directly into the Visual Studio development environment:



Anyone who has ever used static code analysis tools (and PC-lint in particular) will know that they can be **very** thorough, but that it can take much longer to complete a full analysis of a codebase than it would to compile the same code. Given the depth of analysis performed this is a reasonable trade-off, but in some cases the time required for an analysis run can be a problem – which in turn can act as a disincentive to use code analysis tools at all.

One of the original design aims of Visual Lint was to be able to take advantage of multicore processors to complete PC-lint code analysis runs significantly faster than a conventional command line driven process. It does this very well – being typically able to halve the analysis time when run on a dual core system, for example. However, with the evolution of the number of cores in desktop processors typically being a slow process (and one potentially also subject

to bottlenecks in disk and memory I/O) any further improvements in analysis time using this approach alone are fundamentally limited.

To achieve further reductions in analysis time without waiting for the availability of faster processors with more cores than current designs the obvious thing to do is to utilize grid computing technology to distribute analysis tasks across multiple machines on the network. Indeed, this is something Riverblade has been planning for some time, but until recently we had not identified a suitable toolkit for our purposes.

The Alternatives

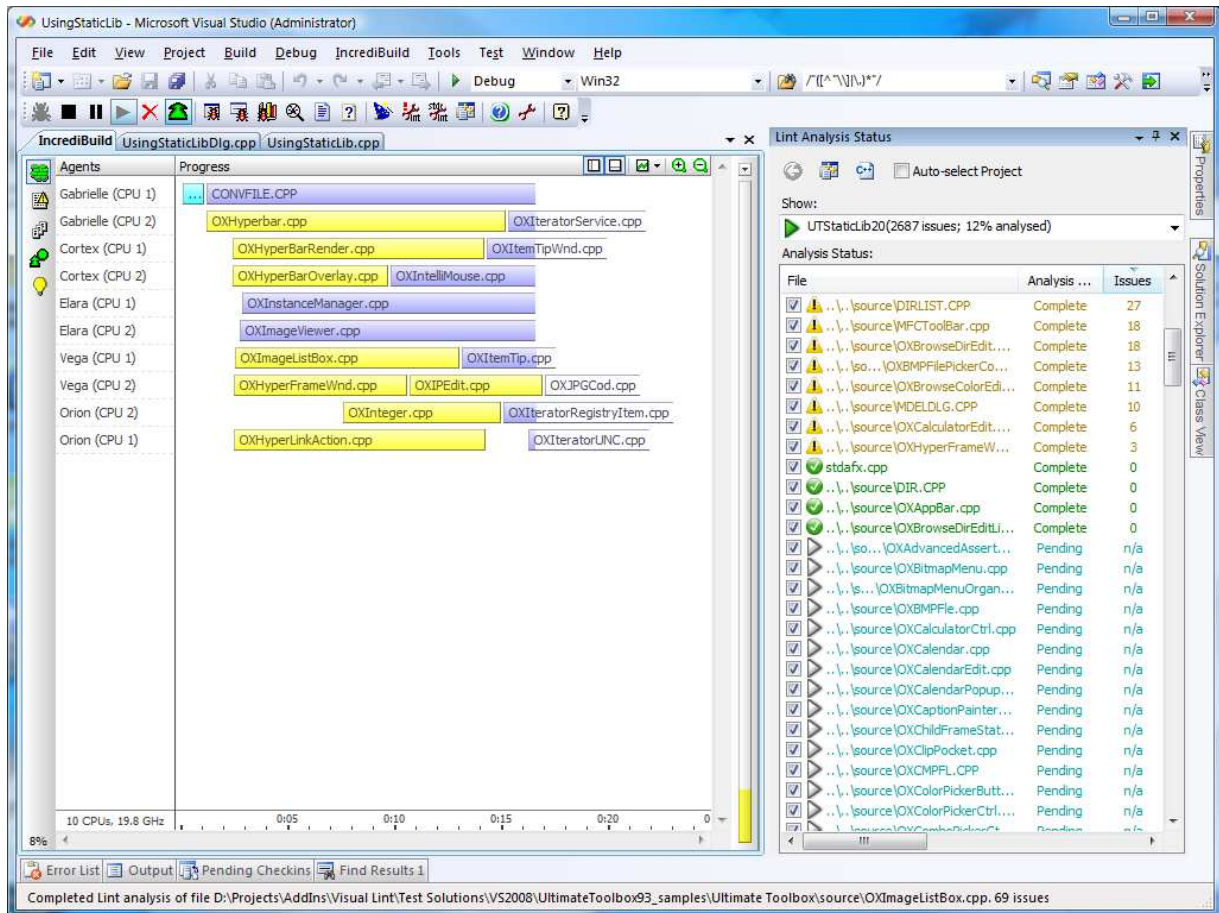
There are several conventional ways to speed up PC-lint analysis runs:

- Buy a faster machine with more cores
- Reduce include dependencies in the codebase(s) to be analyzed
- Use PC-lint 9.0 precompiled and bypass header functionality to cache common include file definitions.

Although the new PC-lint 9.0 precompiled/bypass header support shows promise (our tests showed it could yield a threefold reduction in analysis time on our own codebase), our experience suggests that it can require significant effort to configure correctly in parallel analysis scenarios and may (at least with PC-lint 9.00b) generate spurious analysis errors which are not present in a conventional PC-lint analysis.

Hence all of these solutions are fundamentally limited in the gains which can be achieved, and can require a significant investment in time and/or capital expenditure to reap what could turn out to be a fairly limited benefit.

The Solution – Visual Lint and IncrediBuild



Several months ago we became aware that IncrediBuild (a product I had encountered several years ago but which we had no direct need for in our build process as our build times are quite short) now supported external interfaces for general purpose grid computing via the *XGE Interfaces Extension Package*. Given our long term objective to use grid computing techniques to further accelerate PC-lint analysis tasks, it was obvious that this represented a potential solution we had to investigate.

Of the available XGE interfaces (XML, Automatic Interception and Submission), the XML interface seemed best suited to our requirements as it allowed us to specify in detail the properties of each task, and receive raw analysis results from completed analysis tasks directly.

We found the XML interface to be relatively straightforward, and despite some initial difficulties in uniquely identifying the output from specific tasks (solved by using the *SourceFile* and *Output* attributes of the interface in a slightly non-standard way), we soon had a working development build of Visual Lint which was capable of using IncrediBuild to run PC-lint analysis tasks.

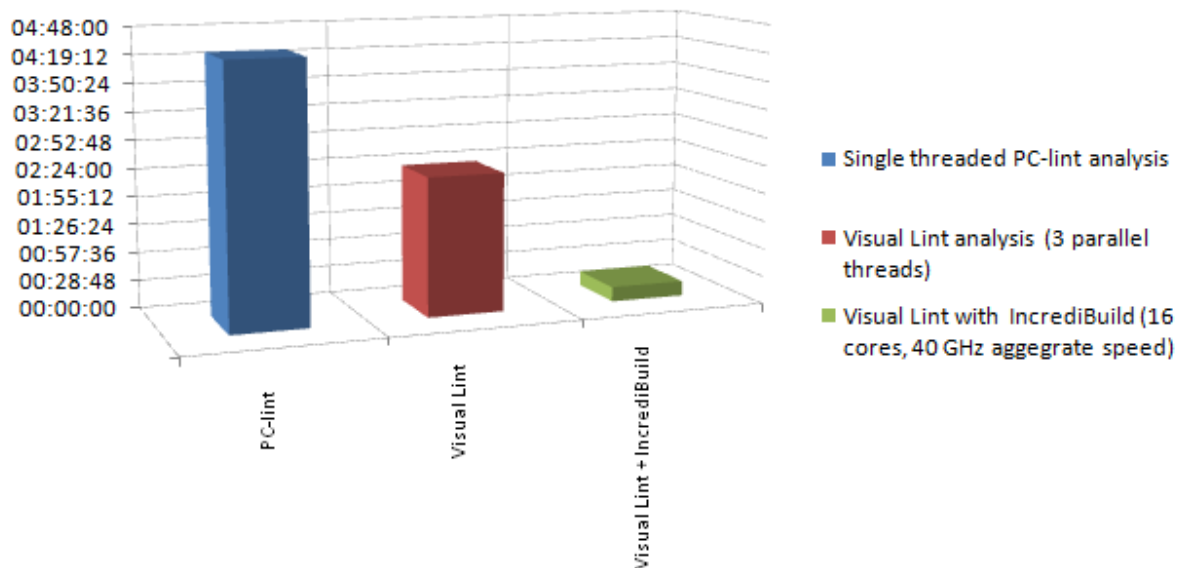
The results of the tests performed using this combination are (to say the least) striking. Using just six agents (four dual and two quad core desktop systems) we were able to complete a complete analysis of the Visual Lint codebase in just over **15 minutes** – a seventeen-fold improvement on the single threaded equivalent.

This is illustrated in the table below:

Description	Analysis time (178 kLOC codebase)	Speed relative to single threaded analysis
1 Core (2.2 GHz). Single threaded conventional PC-lint analysis	4 hours 25 minutes	x1
2 Cores (aggregate speed 4.4 GHz). Native Visual Lint analysis using 3 parallel analysis threads	2 hours 22 minutes	x1.8
16 Cores (aggregate speed 40 GHz) using XGE	15 minutes 10 seconds	x17.5

As you can see from the table above, the reduction in the time taken to complete the analysis is quite considerable.

Analysis Time Comparison



Benefits

Integrating IncrediBuild support into Visual Lint has allowed us to complete full static analysis runs of our own codebase in far less time than previously possible. As a result, we are now able to identify and correct any regressions (typically introduced during refactoring or the development of new features) much more quickly.

Furthermore, the IncrediBuild XGE interfaces have allowed us to integrate this functionality into our product at a small fraction of the cost which we would have incurred had we attempted to implement a grid computing code analysis solution ourselves. It also provides a very clear visualization of the progress of analysis tasks using the IncrediBuild Build Monitor, which is of course integrated into the development environment alongside the Visual Lint displays themselves.

Summary

The integrated Visual Lint/IncrediBuild solution allows developers to perform what are normally time consuming PC-lint code analysis operations in a fraction of the time, dramatically increasing the effectiveness of the code analysis process.

About Riverblade

Riverblade are a UK based Independent Software Vendor and Microsoft Partner specializing in code analysis tools and Visual Studio integration.



Our core product *Visual Lint* integrates the industry standard PC-lint C++ code analysis tool within the Microsoft Visual Studio and Visual C++ development environments. Since its first release in November 2005, Visual Lint has brought automated PC-lint code analysis to Microsoft Visual Studio and Visual C++ developers in a way never before seen in any mainstream code analysis product. Rather than spending time waiting for slow manual analysis operations to complete, Visual Lint allows developers to get on with what they do best - writing code. Analysis results are collected in the background without interrupting the developer, and presented in straightforward, easy to interpret displays.

Further details can be found at http://www.riverblade.co.uk/products/visual_lint. For additional information on Visual Lint including technical or sales information please contact us at enquiries@riverblade.co.uk or visit <http://www.riverblade.co.uk>.

About the Author

Anna-Jayne Metcalfe is the founder of Riverblade Ltd, and the original architect of the Visual Lint static code analysis product. She is an active member of ACCU (<http://www.accu.org>) and the CodeProject (<http://www.codeproject.com>) developer community. She has several articles published on codeproject.com, and her article *“Taming the Lint Monster”* was recently published in the ACCU member journal CVu.



Prior to founding Riverblade she worked extensively in the marine and defense sectors, specializing in virtual instrumentation, automatic test systems and subsea acoustic navigation products.