NASA/CR—2002-211340 AIAA–2002–3706

National Combustion Code Parallel Performance Enhancements

Angela Quealy
QSS Group, Inc., Cleveland, Ohio

August 2002

NASA/CR—2002-211340 AIAA–2002–3706

National Combustion Code Parallel Performance Enhancements

Angela Quealy
QSS Group, Inc., Cleveland, Ohio

Prepared for the
38th Joint Propulsion Conference and Exhibit
cosponsored by the AIAA, ASME, SAE, and ASEE
Indianapolis, Indiana, July 7–10, 2002

Prepared under Contract NAS3–00145

National Aeronautics and
Space Administration

Glenn Research Center

August 2002

Acknowledgments

This work was supported by the NASA Glenn Research Center under contract NAS3–00145 with Theresa Benyo as the monitor. The use of the SGI Origin 2000 via the NASA IT/HPCC program is gratefully acknowledged.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

NATIONAL COMBUSTION CODE PARALLEL PERFORMANCE ENHANCEMENTS

Angela Quealy
QSS Group, Inc.
Cleveland, Ohio 44135

Abstract

The National Combustion Code (NCC) is being developed by an industry-government team for the design and analysis of combustion systems. The unstructured grid, reacting flow code uses a distributed memory, message passing model for its parallel implementation. The focus of the present effort has been to improve the performance of the NCC code to meet combustor designer requirements for model accuracy and analysis turnaround time. Improving the performance of this code contributes significantly to the overall reduction in time and cost of the combustor design cycle. This report describes recent parallel processing modifications to NCC that have improved the parallel scalability of the code, enabling a two hour turnaround for a 1.3 million element fully reacting combustion simulation on an SGI Origin 2000.

Introduction

The National Combustion Code (NCC) is an integrated system of computer codes being developed by an industry-government team for the design and analysis of combustion systems. The objective of this effort is to develop a multidisciplinary combustion simulation capability that will provide detailed analyses during the combustor design process for gas turbine engines. NCC will enable the analysis of a full combustor from compressor exit to turbine inlet. Such a system is critical for optimizing the combustor design process.

The primary flow solver for NCC is a Navier-Stokes flow solver based on an explicit four-stage Runge-Kutta scheme. The original code, CORSAIR,[1] was developed by Pratt & Whitney and was designed from the beginning to use unstructured grids and parallel processing. This code has since been upgraded by NASA Glenn Research Center with new models (chemistry, spray, turbulence) and enhanced parallel processing.[2,3]

The Numerical Propulsion System Simulation (NPSS) project at NASA Glenn has supported the NCC performance enhancement effort. An NPSS milestone to use NCC to run a large scale, fully reacting combustor simulation within a three hour turnaround time was met in April 2001. The effort to meet this milestone will be described in the Performance Improvements section.

Parallel Implementation

Process Organization

A Single Program Multiple Data (SPMD) strategy was used with NCC. All processes are computational processes consisting of a copy of NCC operating on its own local domain. Processes that share common cell faces exchange information.

Domain Decomposition

The original NCC domain decomposition method is based on the number of computational elements in the simulation geometry and on the number of available processors. During a preprocessing stage, the cells of the unstructured grid are re-ordered to run consecutively along the longest axis of the grid. The number of cells is then evenly divided among the number of available processors. The last processor takes on any extra cells if the division between processes is not even. These extra cells are typically not a significant factor in the overall load balance.

This "on-the-fly" domain decomposition allows the user to select the number of processors used by the simulation at startup, based on processor availability. The load is well balanced across all processors rather than being statically determined during grid generation. However, no effort is made to minimize the size of messages exchanged between processes by minimizing the number of cells along the process interface boundaries. This issue was addressed by implementing an alternative domain decomposition strategy that will be described in the Performance Improvements section.

Message Passing Requirements

The NCC code solves 19 partial differential equations in the benchmark configuration used in this study. The message passing to handle these computational requirements in the original code was significant. Depending on the simulation, each process exchanged as many as 563 messages with a neighboring process each iteration. This had been reduced to 130 messages through previous performance improvements.[2] The message passing has been reduced even further through additional code enhancements that will be described in the Performance Improvements section.

Performance Measurement

Benchmark Test Cases

The test case used for this performance improvement effort was the Lean Direct Injection/Multiple Venturi Swirler (LDI-MVS) combustor, which is a full three-dimensional, planar, periodic sector rig.[4] This combustor has been proposed for use in the evaluation of advanced low-emission combustor concepts. The computational domain consists of approximately 444,000 tetrahedral elements. The problem size of interest for the NPSS three hour turnaround milestone is 1.3 million elements. A geometry of this size was not available when this effort began, so the execution time of the smaller 444k element test case is being scaled up by a factor of three to estimate the performance of the larger problem. An LDI-MVS combustor geometry with approximately 971,000 tetrahedral elements later became available. The results for this test case are being scaled by a factor of 1.34 to also estimate the execution time of the larger 1.3 million element problem.

A 12-species, 10-step reduced kinetics mechanism is being used to account for the amount of computational resources required for the chemistry simulation. The Intrinsic Low Dimensional Manifold (ILDM) kinetics module[5,6] is being used to model the chemistry. All turbulence and species equations are turned on during benchmarking. The enthalpy equations were initially off but were turned back on as described in the Performance Improvements section. Convergence for a fully reacting solution is estimated to require 10,000 iterations.

Recently, an additional test case has become available with approximately 1.3 million tetrahedral elements. This three-dimensional test case is a premixed hydrogen/air combustor.[7] The combustion is simulated by the ILDM approach. This test case has been used to verify that the estimates for the 1.3 million element problem based on the 444k and 971k element benchmark test cases are valid.

Benchmark Hardware Platforms

An SGI Origin 2000, which is used as the primary benchmark hardware for this performance evaluation, has a cache-coherent, non-uniform memory access (ccNUMA) architecture. Although the SGI Origin 2000 appears to the user as a shared memory machine, memory is physically distributed among the processing nodes. Benchmark results on this platform are averaged over multiple runs in a typically loaded environment. The scheduler on this platform ensures that the processors are not oversubscribed, so the results are fairly repeatable. SGI's version of MPI was used for all benchmarks on this platform.

Initial benchmark measurements were run on a 256 processor SGI Origin 2000 with 250 MHz R10000 processors and a 4 MB secondary cache. A new 512 processor machine was later made available with 400 MHz R12000 processors and an 8 MB secondary cache.

Metrics

The metric of greatest interest in this performance improvement effort has been the time required to reach a solution for a 1.3 million element problem. This "estimated time to solution" is calculated by timing the main iteration loop of NCC after one full iteration has been completed. This allows systems that load code incrementally from disk to complete an entire cycle before benchmark timing begins. The main iteration loop is timed for a fixed number of iterations so that an average time per iteration can be calculated. The number of iterations timed varies depending on the size of the problem and the number of processors used. Typically 200 to 500 iterations are timed. The estimated time required to complete 10,000 iterations for the 1.3 million element problem is then estimated using the appropriate scaling factor for the test case. The goal of this effort was to achieve a three hour estimated time to solution for a 1.3 million element problem.

The initialization and termination sections of the code are excluded from benchmark timing because these sections of code consume very little time relative to the time required to reach a solution. This time segment would have been enough to skew the results of a 200 iteration benchmark result.

Parallel speedup and efficiency are calculated to indicate how well NCC uses the available parallel resources. The parallel speedup metric is calculated by taking the ratio of the time per iteration for the serial case versus the time per iteration for the parallel case. The parallel efficiency is the ratio of the parallel speedup to the number of processors used in the calculation. A desire to keep the parallel efficiency above 80% determined the maximum number of processors used at any point during this performance improvement effort.

Performance Improvements

Efforts to achieve a three hour turnaround with NCC focused on the steady-state problem only. In April 1995 using the original NCC code, the 444k element benchmark test case consumed 61.4 seconds per iteration when running with 64 processors on an IBM SP-2. It was estimated that a solution for a 1.3 million element simulation could be reached within 512 hours using this baseline code. Significant effort improved that performance so that by September 1999, the 444k element benchmark test case required only 1.08 seconds per iteration using 56 processors on an SGI Origin 2000.[2] The estimated time required to reach a solution to a 1.3 million element simulation dropped to nine hours. This nine hour mark became the new baseline for the current performance improvement effort. The baseline speedup curve is illustrated in figure 1.

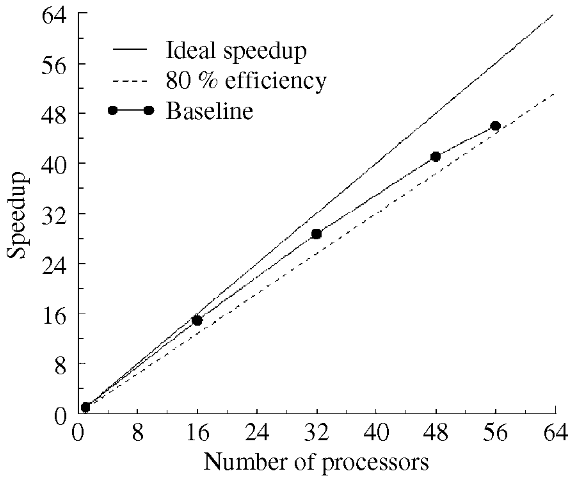

Figure 1.—Baseline speedup curve for the LDI-MVS 444k element test case on the SGI Origin 2000.

System Configuration Improvements

An upgrade of SGI's FORTRAN 90 compiler and MPI implementation improved performance by 4.5%. A solution to a 1.3 million element problem was estimated to be achievable in 8.6 hours using 56 processors on an SGI Origin 2000.

Message Reduction

The number of messages being exchanged per iteration was reduced considerably by packing multiple arrays into fewer, larger messages. Although the number of messages being exchanged between communicating processes dropped from 130 to 51, the number of bytes exchanged remained the same. This had minimal affect on performance; however, it was anticipated that future modifications might benefit from fewer messages being exchanged.

Test Case Modification

The parallel performance of the benchmark test case has thus far been reported with the enthalpy equation turned off. Solving the enthalpy equation in the benchmark test case produces a significant drop in performance but was necessary for test case accuracy. When 56 processors were used with the enthalpy equation turned on, the time required per iteration increased to 1.24 seconds per iteration, increasing the estimated time to reach a solution for a 1.3 million element problem to 10.3 hours. The number of messages being exchanged increased slightly from 51 to 64.

METIS Domain Decomposition Strategy

The original NCC domain decomposition method evenly balanced the computational load across all available processors. No effort was made, however, to minimize the size of messages exchanged between processes by minimizing the number of cells along the process interface boundaries. To address this issue, an alternative domain decomposition strategy was implemented for NCC using METIS,[8] a grid partitioning tool developed at the University of Minnesota. METIS is used to obtain a partitioning of an NCC mesh. This partitioning is then used by NCC to distribute the computational domain across the available processors.

The 444k element benchmark test case was run using 96 processors with both domain decompositions. Figure 2 illustrates the number of computational cells in the original decomposition and the number of ghost cells where information is required from neighboring processes. Figure 3 illustrates the number of computational and ghost cell elements in the METIS decomposition. The number of computational cells is about the same with both decompositions; however, the number of ghost cells is greatly reduced with the METIS decomposition. This difference is reflected in the size of the messages exchanged between processes. With the original decomposition, a total of 415 MB are exchanged each iteration. With the METIS decomposition, a total of 62 MB are exchanged each iteration. The number of communication partners increased from four to eighteen with the METIS decomposition, so more messages were exchanged. However, the total message size per iteration was almost seven times smaller and more balanced in size between

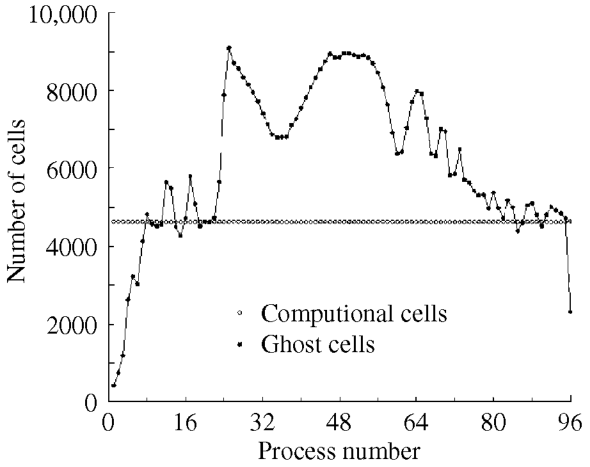

Figure 2.—Computational and ghost cells for the LDI-MVS 444k element test case using the original domain decomposition.

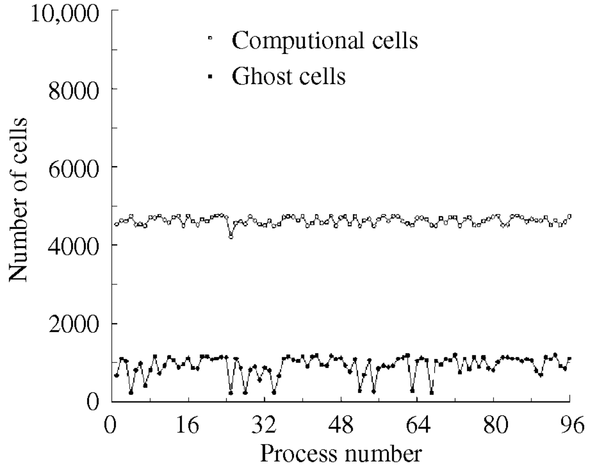

Figure 3.—Computational and ghost cells for the LDI-MVS 444k element test case using the METIS domain decomposition.

processes, which eliminates excessive waiting time. The computation-to-communication ratio improved from 2:1 to 2.5:1. The 444k element benchmark test case required 1.04 seconds per iteration when using 56 processors with the METIS decomposition. This is a 1.2x improvement over the original domain decomposition.

Using METIS improved the scalability of NCC on the SGI Origin 2000. With the original domain decomposition and 96 processors, the 444k element test case required 0.93 seconds per iteration. The parallel efficiency was 60%. With METIS, the same 96 processor case required 0.69 seconds per iteration while maintaining a parallel efficiency

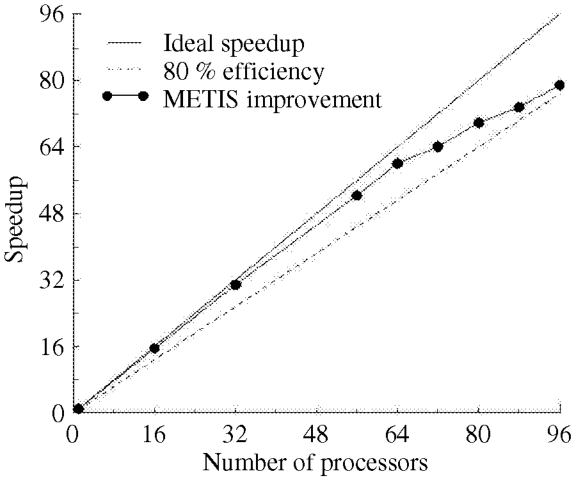

Figure 4.—Speedup for the LDI-MVS 444k element test case using the METIS domain decomposition strategy.

of 82%. A parallel speedup of 78.9 was achieved. The speedup curve is illustrated in figure 4. It was estimated that the time required to reach a solution to the 1.3 million element problem was 5.8 hours.

Hardware Improvements

NCC was ported to an SGI Origin 2000 with 400 MHz processors and an 8 MB secondary cache. The 444k element benchmark test case was run with 96 processors to determine the performance improvement attributed to the new hardware. The test case required 0.44 seconds per iteration and maintained a parallel efficiency of 80%. It was estimated that a solution to a 1.3 million element problem could be reached in 3.7 hours. This is a 1.6x improvement over running the same code on an SGI Origin 2000 with 250 MHz processors and a 4 MB secondary cache. The speedup curve is illustrated in figure 5.

A superlinear speedup was observed when 32 processors were used. This superlinear speedup can occur on the SGI Origin 2000 when the single-processor case requires more memory than is available on any given node, and additional memory must be accessed from other physical nodes. The time per iteration for the single-processor case is slower than it would be if all memory was locally accessible. When larger numbers of processors are used, the computational domain per process can fit within the local memory of each individual processor, so no additional performance cost is incurred. With the larger secondary cache on the upgraded SGI Origin 2000, the parallel version may also fit better in the cache than the

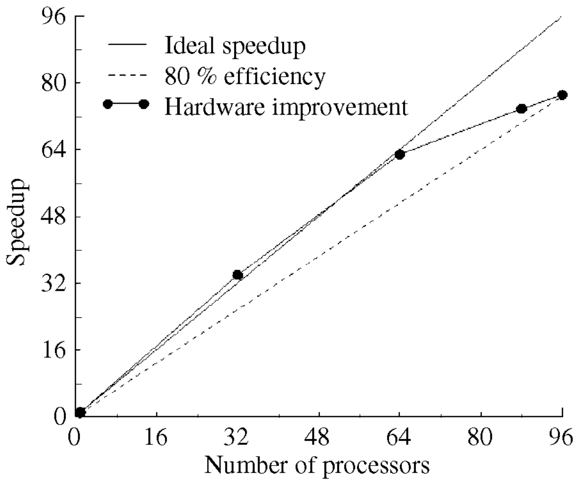

Figure 5.—Speedup for LDI-MVS 444k element test case on the upgraded SGI Origin 2000.

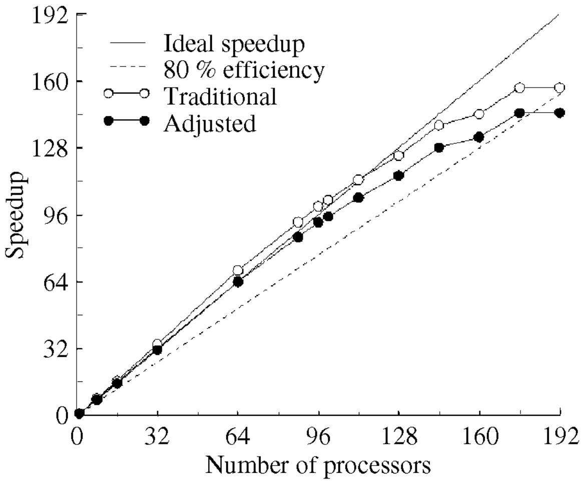

Figure 6.—Traditional and adjusted speedup for the LDI-MVS 444k element test case on the upgraded SGI Origin 2000.

single-processor case. The difference in memory access between the parallel and single-processor cases is reflected as a superlinear speedup.

Message Passing Improvements

Previous attempts were made to improve performance by packing many smaller messages into fewer, larger messages. This packing reduced the overall number of messages being exchanged per iteration. However, the overall performance did not improve over the unpacked case. It is speculated that the larger size of the packed messages was in the range where communication performance is non-optimal.

The packing of many smaller messages into fewer, larger messages was again attempted after the METIS domain decomposition software was utilized. METIS reduced the size of the packed messages over that obtained with the simple, single-axis domain decomposition strategy used earlier. The number of messages exchanged between processes each iteration decreased from 64 to 11. As a result, the overall execution time was reduced.

The 444k element benchmark test case was run with 96 processors on the SGI Origin 2000. With the message passing improvements, the time per iteration dropped to 0.36 seconds. The parallel efficiency was 104.3%, again indicating superlinear speedup due to the large memory requirements of the single-processor case. To account for this, an adjusted speedup and efficiency were calculated using the assumed near-linear performance of the 64 processor test case to estimate the single-processor performance. The adjusted speedup for the 96 processor case was then calculated to be 92.4 and the adjusted parallel efficiency was 96.3%. Once again, the scalability of the code improved, which allowed increasing the number of processors being used for this test case to 176 while maintaining an adjusted parallel efficiency above 80%.

Performance Summary

Current benchmarks with the 444k element test case on the SGI Origin 2000 using 176 processors require 0.23 seconds per iteration. It is now estimated that two hours are required to reach a solution for a 1.3 million element problem. The adjusted speedup is 144.7 and the adjusted parallel efficiency is 82%. Figure 6 illustrates the traditional and adjusted speedup curves for the current code.

The performance results for the 971k element benchmark test case support the conclusion that the 1.3 million element problem could be solved in less than three hours to meet the NPSS milestone. Using 240 processors on the Origin 2000, the 971k element test case requires 0.35 seconds per iteration. It is estimated that 1.3 hours are required to reach a solution for a 1.3 million element problem. The adjusted speedup is 206.6 and the adjusted parallel efficiency is 86%. The traditional and adjusted speedup curves are illustrated in figure 7.

The performance results for the 1.3 million element test case again confirm that a solution can be reached in less than three hours. Using 320 processors, the 1.3 million element test case requires 0.42 seconds per iteration. The estimated time to reach a solution is 1.2 hours. The adjusted speedup is 277.3 and the adjusted parallel

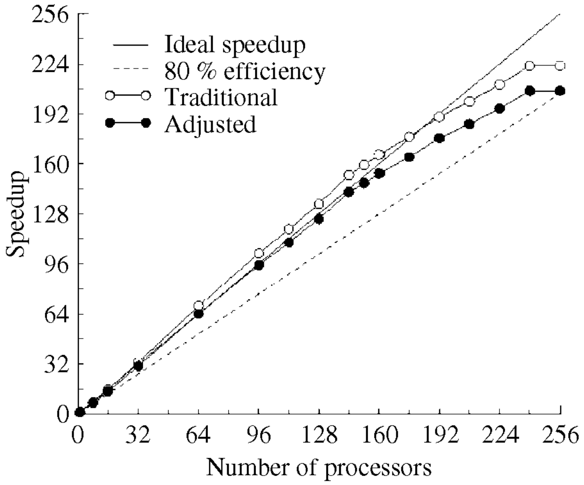

Figure 7.—Traditional and adjusted speedup for the LDI-MVS 971k element test case on the upgraded SGI Origin 2000.

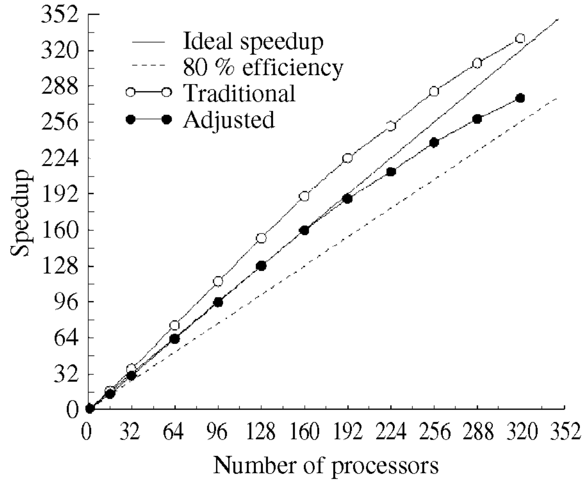

Figure 8.—Traditional and adjusted speedup for the 1.3M element test case on the upgraded SGI Origin 2000.

efficiency is 86.7%. The traditional and adjusted speedup curves are illustrated in figure 8.

The initialization time increased significantly for the 1.3 million element test case as the number of processors increased. The problem was traced to the initialization I/O which was streamlined, reducing the initialization time for the 256 processor case from almost 700 seconds to 48 seconds.

The overall results since the NCC performance improvement effort began in 1995 are illustrated in figure 9.

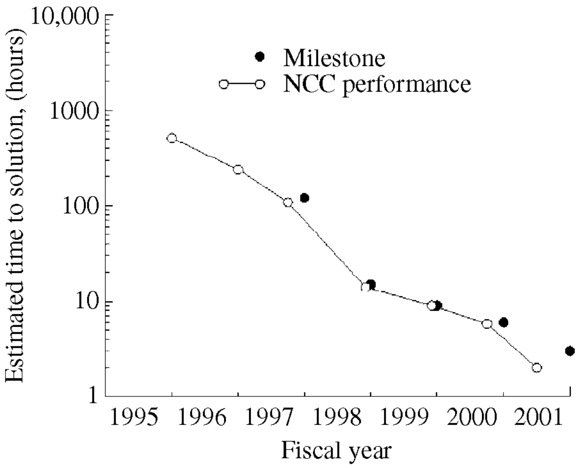

Figure 9.—Overview of performance improvements since 1995.

The original NCC code exchanged 563 messages per iteration between neighboring processes. This number has been reduced by a factor of 50 to 11 messages per iteration.

Concluding Remarks

The performance of the NCC code has been enhanced significantly over the past several years. Recent performance improvements have included the addition of the METIS domain decomposition strategy and the streamlining of message passing in the code. Additional improvements can be attributed to hardware and software upgrades. It was estimated that the baseline code would require more than 500 hours to reach a solution for a 1.3 million element problem in 1995. The estimate for the current code to achieve a solution to the same problem is less than two hours.

References

1. Ryder, R.: CORSAIR User's Manual. SID: Y965, version 1.0, United Technologies Corporation, 1993.
2. Quealy, A., et al.: National Combustion Code: Parallel Implementation and Performance. NASA/TM—2000-209801, 2000. http://gltrs.grc.nasa.gov/GLTRS
3. Liu, Nan-Suey; and Quealy, Angela: NCC: Multidisciplinary Design/Analysis Tool for Combustion Systems. HPCCP/CAS Workshop Proceedings 1998, NASA/CP—1999-208757, 1999, pp. 183–188. http://gltrs.grc.nasa.gov/GLTRS
4. Stubbs, Robert M.; and Liu, Nan-Suey: Preview of the National Combustion Code. AIAA Paper 97–3114, 1997.
5. Norris, A.T.: Automated Simplification of Full Chemical Mechanisms. AIAA Paper 97–3115, 1997.

6. Norris, A.T.: Automated Simplification of Full Chemical Mechanisms—Implementation in National Combustion Code. AIAA Paper 98–3987, 1998.
7. Shih, T.-H., et al.: A Study of Hydrogen/Air Combustor Using NCC. AIAA Paper 2001–0808, 2001.
8. Karypis, G. and Kumar, V., "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices," Version 4.0 User's Manual, University of Minnesota, September 20, 1998.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE August 2002	3. REPORT TYPE AND DATES COVERED Final Contractor Report

4. TITLE AND SUBTITLE

National Combustion Code Parallel Performance Enhancements

5. FUNDING NUMBERS

WU–704–20–13–00
NAS3–00145

6. AUTHOR(S)

Angela Quealy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

QSS Group, Inc.
21000 Brookpark Road
Cleveland, Ohio 44135

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13151

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2002-211340
AIAA–2002–3706

11. SUPPLEMENTARY NOTES

Prepared for the 38th Joint Propulsion Conference and Exhibit cosponsored by the AIAA, ASME, SAE, and ASEE, Indianapolis, Indiana, July 7–10, 2002. Project Manager, Theresa Benyo, Turbomachinery and Propulsion Systems Division, NASA Glenn Research Center, organization code 5880, 216–433–8723.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories: 61 and 05 Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov

This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The National Combustion Code (NCC) is being developed by an industry-government team for the design and analysis of combustion systems. The unstructured grid, reacting flow code uses a distributed memory, message passing model for its parallel implementation. The focus of the present effort has been to improve the performance of the NCC code to meet combustor designer requirements for model accuracy and analysis turnaround time. Improving the performance of this code contributes significantly to the overall reduction in time and cost of the combustor design cycle. This report describes recent parallel processing modifications to NCC that have improved the parallel scalability of the code, enabling a two hour turnaround for a 1.3 million element fully reacting combustion simulation on an SGI Origin 2000.

14. SUBJECT TERMS Parallel processing; Computational fluid dynamics; Computer codes; Cost reduction	15. NUMBER OF PAGES 13 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT