

Distributed Computing architecture for image-based Wavefront Sensing and 2 D FFTs

Jeffrey S. Smith¹, Bruce H. Dean², and Shadan Haghani³

Optics Branch / Code 551, NASA Goddard Space Flight Center Greenbelt, MD 20771

ABSTRACT

Image-based wavefront sensing (WFS) provides significant advantages over interferometric-based wavefront sensors such as optical design simplicity and stability. However, the image-based approach is computational intensive, and therefore, specialized high-performance computing architectures are required in applications utilizing the image-based approach. The development and testing of these high-performance computing architectures are essential to such missions as James Webb Space Telescope (JWST), Terrestrial Planet Finder-Coronagraph (TPF-C and CorSpec), and Spherical Primary Optical Telescope (SPOT). The development of these specialized computing architectures require numerous two-dimensional Fourier Transforms, which necessitate an all-to-all communication when applied on a distributed computational architecture. Several solutions for distributed computing are presented with an emphasis on a 64 Node cluster of DSPs, multiple DSP FPGAs, and an application of low-diameter graph theory. Timing results and performance analysis will be presented. The solutions offered could be applied to other all-to-all communication and scientifically computationally complex problems.

Keywords: Digital Signal Processors (DSP), Fast Fourier Transform (FFT), James Webb Space Telescope (JWST), Field Programmable Gate Arrays (FPGA).

1. INTRODUCTION

Future light-weighted and segmented primary mirror systems require active optical control to maintain mirror positioning and figure to within nanometer tolerances. An image-based wavefront sensor offers a simple solution that differs from conventional wavefront sensing approaches (e.g., Shack-Hartmann, shearing interferometry) in that complicated optical hardware is replaced by a computational approach where the science camera itself serves as the wavefront sensor. Although the image-based approach is simpler to implement in hardware, and thus preferred for space-optics deployment due to reduced risk of optical system failure, the image-based approach trades hardware for a software solution and is therefore computationally intensive. As a result, near real-time operation of the wavefront sensing system requires substantial floating-point performance in addition to high data transfer and communication rates.

Phase-diverse phase-retrieval is an image-based wavefront sensing method that utilizes point source images (or other known object) to recover optical phase information. A number of image-based phase-retrieval techniques have been developed that can be classified into one of two general categories: (a) iterative transform (ITA) [1, 2] or (b) parametric [3, 4]. Fig. 1 depicts the mechanism of classical iterative transform phase-retrieval (termed "error reduction" in the earlier literature [5]), which operates by iteratively enforcing constraints between conjugate Fourier domains.

¹Jeffrey.S.Smith@nasa.gov

²Bruce.Dean@nasa.gov

³Shadan.Hagani@nasa.gov

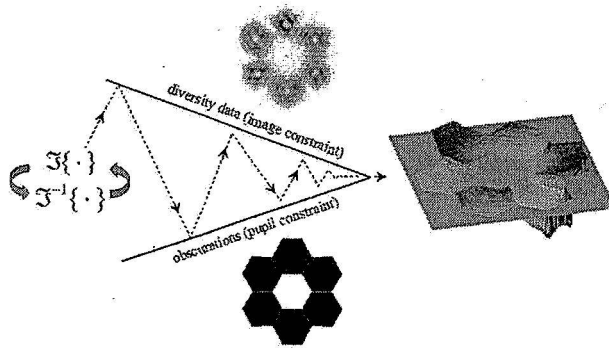


Fig. 1. Iterative transform phase-retrieval ($\mathcal{F}\{\cdot\}$ denotes Fourier transform).

Modifications to the original iterative transform approach have been based on the introduction of a defocus diversity function [6, 7] or on the input-output method [5]. Various implementations of the focus-diverse iterative-transform method have appeared in [8] deviating slightly by utilizing a single wavelength and varying the placement and number of defocused image planes. Modifications to the parametric approach include minimizing alternative objective functions as well as implementing a variety of nonlinear optimization methods such as Levenburg-Marquardt [9], simplex, and quasi-Newton techniques [10].

Including the modifications to the iterative transform approach can be seen in Fig. 2. For each inner loop and for each image, a single iteration is made constraining between conjugate Fourier domains. Thus, a single iterations results in a two dimensional Fourier and Inverse Fourier Transforms; hence, the number of 2-D FFTs is the product of the number of diversity-defocus images, the number of outer iterations, the number of inner iterations, and then 2 (for the Fourier and Inverse Fourier Transform).

The iterative-transform algorithm requires numerous iterations from the image plane to the pupil plane via the Fourier transform pair, see Fig. 1 and Fig. 2. Current desktop and server line general-purpose central processing units (CPU) have been optimized for performing multiple tasks and use principles of data locality to increase performance, yet for performing large 2-D FFT's, performance dramatically decreases [11, 12]. For example, a current general-purpose CPU can take several seconds for a double precision 2-D FFT of size 2048x2048. This is mainly due to the fact that the memory architecture is not optimized for such large data sets. To perform the numerous large 2-D FFT's efficiently, several application specific architectures have been developed and are discussed further below. Many 1-D and 2-D FFT algorithms exist, yet for each element of the output, the algorithms require access to every element of the input [13]. Thus, as one naive approach to parallelization, an image cannot be divided into sub-components that are processed completely independent. Typically, the 2-D FFT is computed as a series of 1-D FFT's [14]. Thus, a 1-D FFT is performed on each row; then, the 1-D FFT is performed on each column of the result. Thus, the total number of 1-D FFTs is the product of the number of single inner loop iterations, the diversity-defocus image size, and 2 (for the rows and for the columns). For example, 4 diversity-defocus images of size 512x512, with 50 outer loops and 10 inner loops results in more than 4,000,000 1-D FFTs.

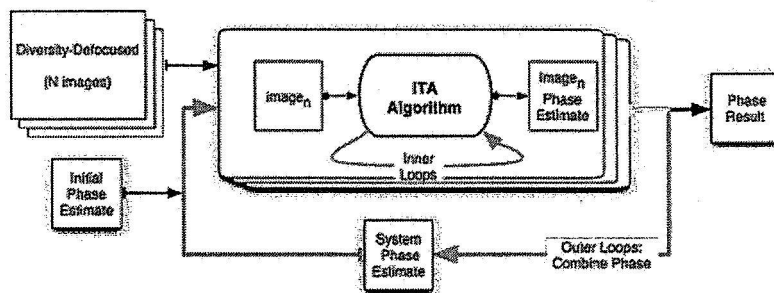


Fig. 2. Block Diagram of Iterative transform phase-retrieval of Fig. 1.

The computational complexity of iterations increases with the adoption of the Hybrid Diversity Algorithm (HDA) which has been validated at the nanometer level [15]. The HDA will perform the ITA algorithm in Fig. 2 multiple times, making modification to an adaptive diversity function to incorporate feedback.

2. DISTRIBUTED ARCHITECTURES

Image-based wavefront sensing has a significantly higher computational requirement than analog optical wavefront sensing. Additionally, the computational requirements are proportional to the accuracy of the wavefront measurement. E.g., course alignment and course phasing use less complicated models to determine the necessary corrections; similarly, fine phasing requires more complex algorithms. Without sacrificing accuracy, it is optimal to minimize the time between image acquisition and correction [16]. Additionally, to increase the resolution of the wavefront requires an increase in the data set size, and thus computational complexity.

The process of performing wavefront sensing on N Diversity-Defocused images is highly parallel for each image, as such, past approaches to increase performance have used 1 to N general-purpose CPUs as a cluster [17]. Consequently, this provides a maximum factor of N improvement, while having the negative effect of increasing power requirements, footprint, and cooling requirements by N . The primary solution is to provide N application specific, highly optimized computational cores. An earlier iterative-transform algorithm implementation for the Analog Devices Hammerhead DSP has been presented discussed, [18].

To address these bottlenecks, super computing architectures are utilized to achieve the necessary computational demands of the 2-D FFT. As described, the 2-D FFT can be performed as a multiple step process of performing the 1-D FFT on each row and resulting column. The bottleneck for performing this on a distributed system becomes the resulting transpose of the entire image. As seen in Fig. 3, the distributed transpose necessitates an all-to-all communication, and thus is non-trivial as one increases the number of processing elements. This compares to the traditionally naive approach of modifying the index methodology for a 2-D image, i.e. $x[i][j] = x[j][i]$, $\forall i, j \in [1 : N]$.

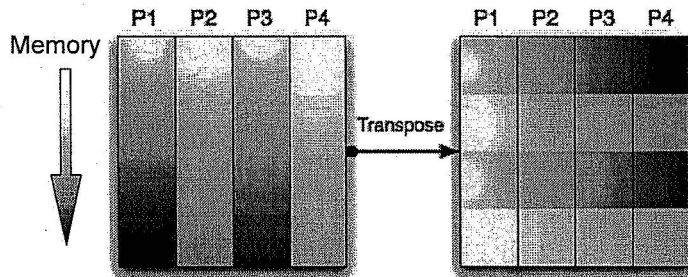


Fig. 3. Characterization of a distributed transpose for 4 Processing units

The computational cores used on each image can be further divided to perform a distributed 2-D FFT. To perform the distributed 2-D FFT, the computational requirements scales linearly with the number of sub-components up to the size of the diversity-defocus image. Thus, the computational requirements of the ITA on a 512x512 diversity-defocus image size scales linearly until one reaches 512 sub-components of a computational core, or 512 Processing units. For implementations of distributed 1-D FFTs, and thus, the scalability for an increase in the number of sub-components larger than the diversity-defocus image size, various algorithms have been explored externally [19]. All discussion in this paper will assume the 1-D FFT is the smallest component of division.

The transpose within the 2-D FFT is the main component of data transfer between sub-components; the transpose necessitates a transfer of sub-blocks of data from each processing unit to every other processing unit, Fig. 3. The all-to-all communication does not scale without careful care given to the underlying architecture of a distributed system. To achieve linear scalability in performance for an all-to-all communication, one must have an architecture that provides a direct connection of all processing units to all other processing units, such as the crossbar switch architecture [20] or the complete-graph of type K_n (i.e. n branches for n nodes) [21]. In practice, this is not ideal because the number of interconnects per node grows as $N-1$, and thus, total number of interconnects for the complete-graph of type K_n grows as $O(n^2)$, as seen in (Eq. 1). Similarly, the crossbar switch architecture grows as $O(n^2)$ with the total number of switches.

$$\sum_{i=1}^{N-1} i = \frac{N(N-1)}{2} \quad (\text{Eq. 1})$$

To address this problem, several solutions such as the n-dimensional hypercube or the grid architecture are used in practice. These architectures are not the optimal choice under the constraint of number of nodes and the number of branches per node. The optimal homogenous architecture for the all-to-all communication would minimize the graph diameter, under the constraint of degree of each node. Thus, the optimal graph would restrict the number of branches per node, while minimizing the maximum of all of the “shortest paths”. An example of a solution to this problem is the Peterson Graph, as seen in Fig. 4 (a). Furthermore, the Moore’s bound for a p-node undirected graph provides a methodology for placing a bound on the diameter, as seen in (Eq. 2 and Fig. 4 (b,c,d), [22]). For Fig. 4 (b,c,d), the leaf nodes need additional branches added to be a complete d-degree graph. For significantly large graphs, only optimum solutions exist, for example the Levi Graph, with 30 nodes and diameter 4, is an optimum solution for the 46 node 3-degree diameter 4 graph in Fig. 4 (d), [23].

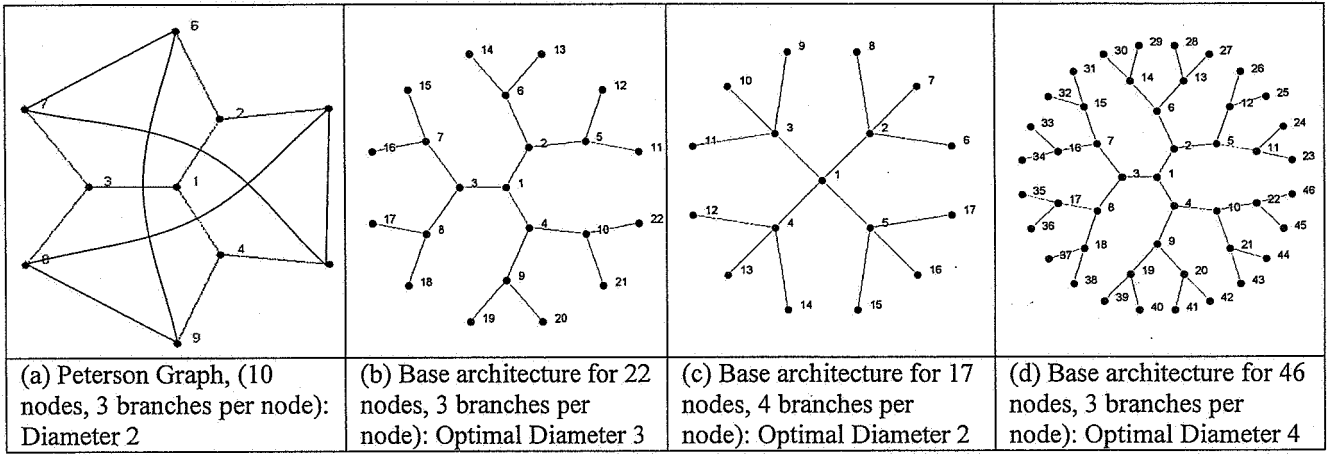


Fig. 4. Architecture branches per node with varying diameters.

$$p \leq 1 + d \frac{(d-1)^D - 1}{d-2} \quad \text{or} \quad D \geq \log_{d-1} \left[\frac{(p-1)(d-2)}{d} + 1 \right] \quad (\text{Eq. 2})$$

In practice, it is impractical to develop such architectures because of scalability, feasibility, or cost. In many cases, simply minimizing the graph diameter does not provide an optimal solution because of load balancing, irregular network flow, or other problems. Yet, for the 2-D FFT transpose, the all-to-all communication is perfectly balanced and each node communicates at a pre-determined and predictable time. Thus, the graph diameter is the primary variable that is optimized for a given architecture for the solving the transpose problem.

For this research several low-diameter architectures were evaluated and implemented, Fig. 8 and Fig. 9. The majority of the research was conducted on three independent systems: a 64 node cluster of Analog Devices TS101 DSPs, a 24 node cluster of Analog Devices TS201 DSPs, and a 4 node cluster of Virtex 4 SX55 Xilinx FPGAs. The TS101 system is rated at 96 GFlops and the TS201 system is rated at 72 GFlops. The TS101 and TS201 both provide methodologies for increasing the performance of HDA for include ground-based wavefront sensing, telescope image processing, laboratory optical processing, system design and tolerancing, Monte-Carlo simulations, and finite element modeling. The Virtex 4 SX55 system provides a path to radiation tolerant environments, [24].

The TS101 and TS201 systems were purchased from Bittware, Inc, and the Virtex 4 SX system was purchased from Alpha Data. The TS101 and TS201 systems are both arranged in clusters of 4 DSPs per board; with a shared SDRAM and 2 dedicated link ports to other nodes in the cluster, as seen in Fig. 6. The remaining 2 link ports are either routine to an external I/O port for the TS-101 or routed to a Virtex 2 Pro for the TS201. The Virtex 4 SX chips have

similar communication ports, a 146 I/O LVDS, which can be divided 4 ways that perform at the core clock speed of the Virtex 4.

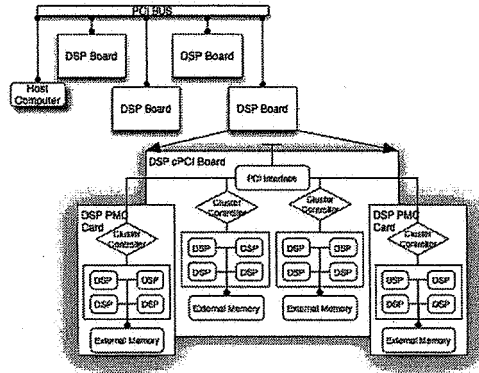


Fig. 5: Sixty Four node cluster of TS101 DSPs.

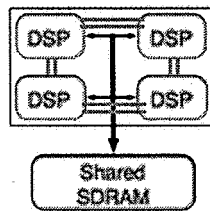


Fig. 6: Cluster of four DSPs (TS101 and TS201) with shared bus to SDRAM memory and link ports.

The four Virtex 4SX chips have been prototyped as a complete-graph K_4 architecture, additionally the 4 Virtex 4SX chips are controlled by a master FPGA, a Virtex 4FX. Each SX chip is connected to the master FPGA using the same I/O configuration, Fig. 7. The SX chip is an FPGA that has been optimized for digital signal processing, while the FX chips have been designed to mimic a microcontroller.

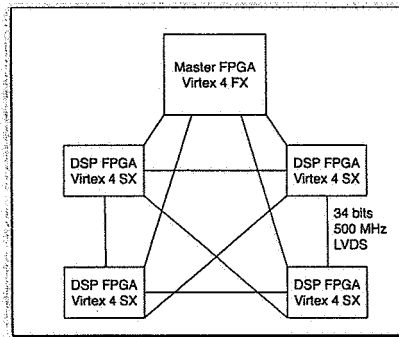


Fig. 7: Cluster of four Virtex 4SX chips and one Virtex 4FX chip.

The TS101 processor has 4 bi-directional link ports that allow the direct connection between two chips. Other processors offer similar communication ports. Similarly, the TS201 DSPs have 4 dual channel uni-directional link ports, and thus minimize the turn around time in changing directions for data flow. This is advantages for the 2-D FFT because every node is sending and receiving data. As a first example of a the effects of a minimum diameter graph for the 2-D FFT refer to Fig. 8. In this example, the node's unique identifier and shortest path from 0 are listed. For graph

(a), a cube, the *shortest-path* between *node 0* and all other nodes is 3 or less, but for graph (b), a 1-möbius cube, the *shortest-path* between any two nodes is 2 or less [25, 26]. For graph (b), the 2-D FFT is 17% faster than graph (a).

To further explore the capabilities of the 64 node TS101 DSP system, the number of DSPs per image was developed and tested for the cases 16, Fig. 9 (a,b) and 32 nodes, Fig. 9 (c). Results of this trade are listed in section 3. Currently, a division of the 64 DSP cluster is being used at the James-Webb-Space-Telescope test bed. Furthermore, for most applications of the ITA, multiple diversity-defocus images are used but for some applications only a single image is applicable. For this case, an architecture for 64 DSPs per image was constructed, Fig. 10. This builds upon the proven methodology from the 16 and 32 DSPs per image.

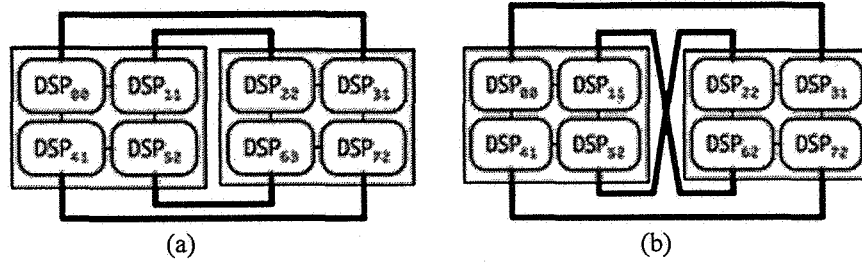


Fig. 8: Two 8 DSP's grid architectures, cube (a) and 1-möbius cube (b). Each node ID is listed, and distance of the *shortest-path* from *node 0*.

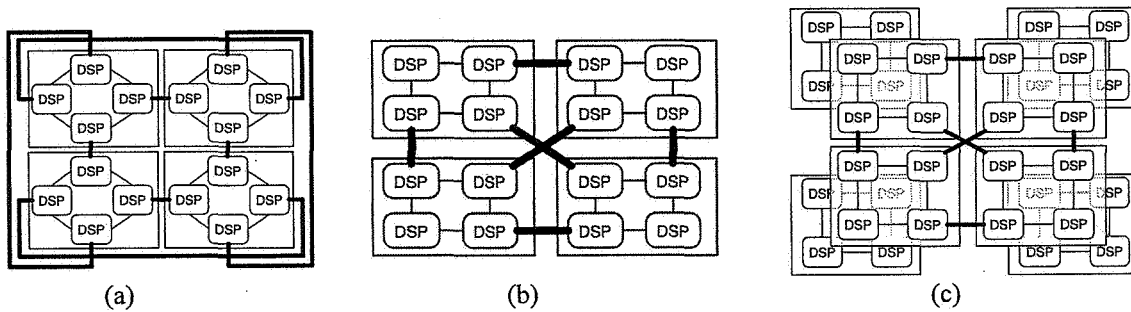


Fig. 9: Architectures developed and tested for 16 (a,b) and 32 (c) DSP.

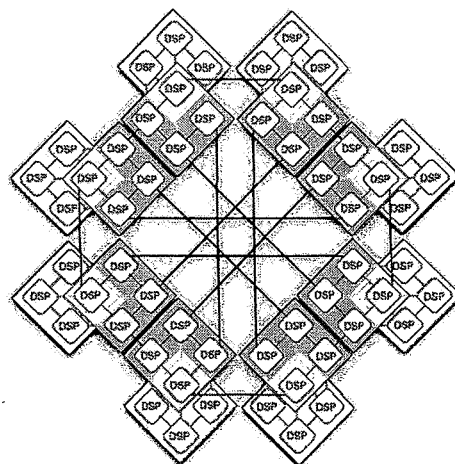


Fig. 10: Architecture design for 64 DSP.

3. PERFORMANCE AND TESTING

The transition from 8 to 16 DSP's per image reduced the sub-component block size for the all-to-all communication by 4, yet, still require the same total data transfer. For example, a 512x512 image on 8 DSP's results in each DSP performing 1-D FFT's on a 64x512 block. Each block is then divided into 8 sub-blocks of 64x64, where each sub-block is transmitted to the corresponding DSP. The 64x64 sub-block has 4096 elements, but for the 16 DSP case, the sub-block is 32x32 and has 1024 elements. The first architecture investigated based on the 16 DSP's is shown in Fig. 9 (a) which maintains an overall low-diameter. With the necessary packet overhead for routing, the reduced sub-block size, and the turn-around time for data flow on the physical layer of the link ports, the result was less efficient than desired, and thus, an alternative 16 DSP architecture was explored, Fig. 9 (b). This architecture utilized edge routing between the clusters of DSP's, thus creating larger packets. This improvement resulted in an overall system improvement for the 2-D FFT by 24% for the 512x512 case. For larger images, the improvement was less significant because the sub-block size increases.

For the 16 to 32 DSP's per image, only the sub-clustered routing architecture was explored which is shown in Fig. 9 (c). For processing, i.e. 1-D FFT's and other mathematical routines, the improvement between 16 and 32 DSP's is nearly linear. Thus, the computational routines provided a factor of x2 improvement. For most parallel computing applications, communication is the bottleneck preventing linear scalability. Contrary to bus architectures, there was an improvement for increasing the number of nodes. This is due to the fact that increasing the number of DSP's also increases the total number of link port channels in the system, and hence, we are increasing the overall effective bandwidth. For the all-to-all communication, the 32 DSP's architecture of Fig. 9 (c), was x1.2 times faster than the 16 DSP's architecture, Fig. 9 (b), for an overall improvement for the 2-D FFT of x1.7. Table 1 summarizes the actual timing for several relevant configurations (it is of interest to note that the 4-image 512x512 case of Table 1 is the configuration baselined for the James-Webb-Space-Telescope wavefront sensing). The 16 and 32 DSP's per image use the architectures shown in Fig. 9 (b) and (c) respectively. Furthermore, there are 4 of these clusters operating on each image. For the algorithm listed in Fig. 2, there were 20 outer-loops and 5 inner-loops for a total of 100 iterations; increasing or decreasing the total number of iterations will scale the timing accordingly by that amount.

The implemented architectures of Fig. 9(b,c) for the 16 and 32 node architectures provide linear scalability with respect to image size. Intuitively, an NxN diversity defocus image would take 4 times longer than one half it size in each direction, i.e. $N^{1/2} \times N^{1/2} = \frac{1}{4}(NxN)$. Yet, for general purpose CPUs, this is not the case, as it has been shown, [12], that the computational performance, measured in Gflops, decreases with image size. The optimal image size for the FFT on a general purpose CPU is between 16x16 and 64x64, depending on the chipset. For the TS101 and TS201 architectures, this is not the case. The computational 1-D FFT scales linearly with array size, and the distributed transpose scales near linear. Thus, the difference in performance between various image sizes for the TS101 architecture can be seen in Table 1, and averages 2.5 between images of size less than 512x512 (smaller numbers signify an independence between image size and performance time. For images larger than this, there is a performance hit between the 512 and the 1024 case because the external SDRAM must be used more heavily.

Table 1: Image size scalability for 16 TS101 DSP, Fig. 9 (b); all times listed in seconds.

Timing of the ITA algorithm, :
1 Diversity-Defocus Images, 1 iterations (seconds)

Diversity Defocus Image Size (NxN)	16 DSP timing	Timing Increase between images
32	0.000179	
64	0.000435	x 2.43
128	0.00117	x 2.68
256	0.00290	x 2.48
512	0.00819	x 2.83
1024	0.0783	x 9.56
2048	0.3265	x 4.17

4. CONCLUSIONS

In summary, the HDA and ITA algorithms entail numerous 2-D FFTs for a single iteration of closed loop control. To achieve the requirements of current space missions that necessitate image-based wavefront sensing, a computational infrastructure must exist to achieve the requirements of closed loop control. Furthermore, a distributed computational architecture processing on each diversity defocus image is the optimal architecture in terms of performance, and various solutions depending on the desired footprint, power consumption, operating environment sets the criteria for the specification of the sub-components. For ground-based WFS, a cluster of DSPs is optimal for performance, while a radiation environment, just as L2 for JWST and TPF required radiation toleration sub-components, such as FPGAs.

Given the different infrastructures between an FPGA and a DSP it is difficult to compare the performance of the TS101 and TS201 DSP system to the Virtex 4 FPGAs. Currently, the baseline for the FPGA is a fixed point processing model, while the DSPs are performing 32 bit floating point.

Additionally, for the TS101 and TS201 systems, many subtle performance enhancements were discovered, such as the optimal minimum image size for a fixed number of DSPs. For 16 DSPs, the run-time for a 32x32 diversity defocus image is the same as a 16x16 image, and similar results were seen for the 8 and 32 DSP architectures. It would be the work of a future research to determine the performance of a 1024 node cluster for a 2048x2048 image size.

5. ACKNOWLEDGEMENTS

The authors thank Lee Feinberg, Peter Maymon, Rene Bouracut, Kim Mehalick and Bruce Jacobs for discussion and support of this work. We also thank Damon Bradley, Semion Kiznher, Sally Heap, Chuck Bowers, Bruce Woodgate, and Mark Flanagan for their support and discussions of this research.

6. REFERENCES

1. R.W. Gerchberg and W.O. Saxton, Phase Determination from Image and Diffraction Plane Pictures in an Electron Microscope, OPTIK, 34, 275 (1971).
R.W. Gerchberg and W.O. Saxton, A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures, OPTIK, 35, 237-246 (1972).

-
- W.O. Saxton, Computer Techniques for Image Processing in Electron Microscopy, in *Advances in Electronics and Electron Physics*, Supplement 10, L Marton and C. Marton, eds. (Academic Press, New York, NY, 1978).
2. N. C. Gallagher, B. Liu, Method for Computing Kineforms that Reduces Image-Reconstruction Error, *Appl. Optics* 12(10) 2328-2335 (1973)
 - N. C. Gallagher, B. Liu, Method for Computing Kineforms that Reduces Image-Reconstruction Error, *Appl. Optics* 12(10) 2328-2335 (1973)
 - R. A. Gonsalves, R. Dumais, P. Considine, On Optimal Holographic Filters, *Proc. SPIE*, Vol. 45, 1974.
 3. R. A. Gonsalves, P. Considine, Phase-Retrieval from Modulus Data, *J. Opt. Soc. Am.*, 66, 961-964 (1976).
 4. W. H. Southwell, Wavefront Analyzer using a Maximum Likelihood Algorithm, *J. Opt. Soc. Am. A3*, 396-399 (1977).
 5. J. R. Fienup, Phase Retrieval Algorithms: a Comparison, *Appl Opt.*, 21(15), 2758-2769, (1982).
 6. D. L. Misell, A method for the solution of the phase problem in electron microscopy, *J Phys.*, D6, L6-L9 (1973).
 - 7 R. A. Gonsalves, Phase Retrieval and Diversity in Adaptive Optics, *Optical Engineering*, 21(5), 829-832, (1982).
 8. C. Roddier, F. Roddier, Wavefront Reconstruction from Defocused Images and the Testing of Ground-based Optical Telescopes, *J. Opt. Soc. Am.*, A10, 2277-2287 (1993).
 - C. Roddier, F. Roddier, Combined Approach to the Hubble Space Telescope Wave-Front Distortion Analysis, *Appl. Opt.*, 32, 2992-3008 (1993).
 - J.R. Fienup, J.C. Marron, R. G. Paxman, T.J. Schulz, J. H. Seldin, B. Thelen, Image Inversion Analysis of the Hubble Space Telescope, in Final Report: JPL Contract 958892 on HST OTA Analysis, August 1991.
 - R. G. Lyon, DCATT Comparative Wavefront Sensing and Optical Control Study Final Report, Feb. 22, 1999.
 - D. Redding, S. Basinger, A. Lowman, F. Shi, P. Bely, R. Burg, G. Mosier, Wavefront Sensing and Control for a next Generation Space Telescope, in *Space Telescopes and Instruments V*, P. Bely, J. Breckinridge, eds., *Proc. SPIE* (3356), 758-772 (1998).
 9. Press, W.H., Flannery, B.P., Teukolsky, S.A., & Wetterling. W.T. *Numerical Recipes in Pascal*. Cambridge University Press (1989).
 10. J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer-Verlag, NY (1999).
 11. Hennessy, John L. & Patterson, David A. (2003) *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, San Francisco, CA.
 12. M. Frigo and S. G. Johnson. <http://www.fftw.org>.
 13. M. Frigo and S. G. Johnson. FFTW: An Adaptive Software Architecture for the FFT.
 14. Gonzalez, R. C. & Woods R. E. (2002) *Digital Image Processing*. Prentice Hall, Upper Saddle River, N.J.
 15. B. H. Dean, Fixed-Lens Wavefront Sensing, JWST Technical Memoranda, NASA/Goddard Space Flight Center, Greenbelt, MD, June 5, 2003.
 16. Tyson, Robert K. & Frazier, Benjamin W. (2004) *Field Guide to Adaptive Optics*. SPIE Press, Bellingham, WA.
 17. W. Hayden, L. Boyce, K. Rehm, D. Redding, C. Ohara, Analysis of Wavefront Sensing and Control Onboard Resource Requirements – 1 & 2, Computer Science Corporation, Technical Memorandum, Apr-2002.
 18. J. S. Smith, Ed Lo, Hardware Implementation of a Phase-Retrieval Algorithm, NASA GSFC Summer Student Presentations, July 2003.
 19. Agarwal, R.C., Gustavson, F.G., Zubair, M., A high performance parallel algorithm for 1-D FFT; *IEEE Supercomputing '94. Proceedings*, pages: 34-40, Nov 1994.
 - Daisuke Takahashi and Yasumasa Kanada, High-Performance Radix-2, 3 and 5 Parallel 1-D Complex FFT Algorithms for Distributed-Memory Parallel Computers; Springer Netherlands, pages 207-208, February 2000.
 20. Andrew S. Tanenbaum and Maarten van Steen, (2002) *Distributed Systems: Principles and Paradigms*. Prentice-Hall, Upper Saddle River, N. J.
 21. Eric W. Weisstein. "Moore Graph." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/MooreGraph.html>.
 22. Behrooz Parhami, (1999) *Introduction to Parallel Processing: Algorithms and Architectures*. Plenum Press, New York, NY.
 23. Eric W. Weisstein. "Levi Graph." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/LeviGraph.html>.
 24. J. Smith. "Radiation Tolerant Hardware Analysis", NASA Goddard Space Flight Center / Technology Transfer, Sept 2005.

-
25. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd edn. MIT Press, Cambridge, MA, 2001.
 26. Tanenbaum, Andrew S. & Van Steen, Maarten, *Distributed Systems: Principles and Paradigms*. Prentice Hall, Upper Saddle River, N.J., 2002.