DL-TR-2007-002

CCLRC

Gigabit Ethernet – an HPC interconnect? II: Further findings from synthetic benchmark studies and enhanced methods for summarising results

Richard Wain, Miles Deegan, Gabriel Sallah, Martyn Guest, Christine Kitchen *and* **Igor Kozin**

March 2007

Council for the Central Laboratory of the Research Councils

Library and Information Services
CCLRC Daresbury Laboratory
Daresbury Warrington
Cheshire WA4 4AD
UK
Tel: +44 (0)1925 603397
Fax: +44 (0)1925 603779
Email: library@dl.ac.uk

ISSN 1362-0207

Gigabit Ethernet – an HPC interconnect?
II: Further findings from synthetic benchmark studies and enhanced methods for summarising results

Richard Wain[*], Miles Deegan*, Gabriel Sallah[§], Martyn Guest*, Christine Kitchen*, Igor Kozin*

Abstract

The Message Passing Interface standard (MPI) is currently the most common programming model deployed by the HPC community to parallelise a wide range of applications in computational science and engineering for distributed memory architectures.

We build on a previous in-depth study of the performance of a range of Gigabit Ethernet MPI implementations and switches using the synthetic MPI benchmark IMB. This paper is essentially an appendix to that study, and presents results from benchmarks of a further three typical 1U Gigabit switches deployed in 'plug and play' mode, along with an enhanced method for analysing the significant amounts of data produced, which aims to accurately capture performance and summarise findings in a more digestible format.

Contents

1. Introduction 2
2. System Details: Hardware and Software 3
3. Benchmarking Methodology 3
4. Analysis of Results: Methodology 4
5. Analysis of Results: Discussion 5
   5.1 The Geometric Mean: The Correct Way to Summarise Benchmark Results? 5
   5.2 New/Further Findings 7
6. Conclusions 7
7. Acknowledgments 8
8. References 8

[*] Distributed Computing Group, CCLRC Daresbury Laboratory, Warrington, WA4 4AD, UK.
[§] Barclays Capital, 5 The North Colonnade, Canary Wharf, London E14 4BB.

1. Introduction

The introduction of 'Beowulf clusters' – clusters of commodity servers or PCs first with Fast Ethernet and then Gigabit Ethernet (GbE) interconnects – has led to a marked increase in access to affordable parallel computing facilities for researchers over the last decade. Prior to this, scientists with HPC needs were reliant on limited access to expensive, often remote, centralised resources based on proprietary RISC or vector technologies.

Our previous study [1] - hereafter referred to as 'paper I' - includes an overview of this area of HPC and references other benchmark studies of Gigabit clusters. For the sake of brevity and tautology avoidance we will not re-present this material here.

There is a need to regularly revisit these types of studies however, due to changes in hardware, e.g. new processors/servers, GbE switches not previously available etc; and previously untested combinations of software - new MPI libraries and compilers, new Linux kernels with altered TCP stacks etc.

Here we concentrate on assessing three additional switches that became available to us since the publication of paper I. Moreover, we examine ways of building on the method presented in paper I for summarising results with further rendering of the large amount of data produced into a format which accurately captures a particular setup's performance and is more easily interpreted.

We are currently in the process of addressing the outstanding tasks we set ourselves previously such as:

- work with switch manufacturers and tune performance through eg judicious management of buffer sizes
- assessment of alternative MPI implementations with TCP-bypass (SCore, GAMMA) and comparison with Scali MPI
- generating benchmark data on other clusters with different server architectures (e.g. Intel Woodcrest and Clovertown)
- benchmarking of TCP offload technologies
- broadening the portfolio of MPI implementations being assessed to include the likes of OpenMPI and MPICH2
- assessment of core/edge architectures and the performance impact of oversubscription

We will report on these efforts in due course and look to extend them further once we have gained access to copper-based 10 Gigabit Ethernet clusters featuring servers with on-board adaptors – a development that should lead to a considerable decrease in the price of 10 GbE and therefore increase in its adoption not only in HPC but also in general enterprise computing.

To reiterate our previous approach to benchmarking: for now, we concentrate on presenting an expanded version of our previous findings from a 'plug and play' exercise with additional data generated on switches from Extreme Networks, HP and Nortel Networks. We have made no attempt to optimize the various system components, eg through Ethernet switch management. Instead we have built libraries with recommended compiler switch settings where available; we have taken Ethernet switches straight out of the box with firmware etc. as supplied to us. So far our focus has therefore been on how we believe such clusters are typically setup and deployed at HPC sites (in the UK at least) by Tier 1 vendors and/or their Integrator/OEM partners, or end-users 'rolling their own'.

These initial studies aim to highlight what we believe are reasonable performance expectations for typical GbE cluster configurations, how this information can then be used as input into a TCO analysis, and how this in turn affects the choice as to whether or not to purchase proprietary MPI libraries, and/or a higher performance interconnect such as InfiniBand instead.

2. System Details: Hardware and Software

The cluster deployed in this benchmarking study is comprised of 32 IBM e325 servers featuring two sockets each with a single-core Opteron CPU clocked at 2.0 GHz. An additional e325 server is used as a head node for management, compilation, job submission etc. Further system architecture details are given in paper I. For the sake of brevity, again we will present a summary of just the 2.6 kernel data generated thus far.

A range of typically deployed 1U 48 port GbE switches from the following manufacturers have so far been assessed (new switches in *italics*):

- Cisco Systems (Catalyst 4948)
- Extreme (Summit48si and *x450-24t*, a 24 port switch, therefore benchmarking of 64 process runs of IMB did not take place)
- Force10 Networks (S50)
- HP (Procurve 2848 and *Procurve 3500yl*)
- Netgear (GS 748T)
- Nortel Networks (5510-48T and *5520-48T*)

A discussion of the architectural factors that differentiate these products in the context of MPI benchmarking will be provided in a future paper detailing our attempts to optimise performance cf. this initial 'plug and play' study.

As before, a mixture of free and commercial MPI libraries was used in this study: MPICH 1.2.7; LAM-MPI 7.1; Scali MPI version 3. Likewise, compilers used were free (GCC 3.3) and commercial (PGI 6.0 and PathScale 2.0). Thus we have benchmarked 7 combinations of MPI library and compiler, namely:

- MPICH with PGI, PathScale and GCC
- LAM-MPI with PGI, PathScale and GCC
- Scali MPI.

A number of tests were carried out in order to ascertain whether or not the cluster was in principle 'fit for purpose'. It was established that BIOS settings, memory types, configuration and performance (through use of the STREAM benchmark), and the performance of Linpack and NASTRAN kernels was consistent across the system and in line with expectations for the e325 server. We made use of the STAB suite from IBM's Egan Ford (see http://sense.net/~egan/bench/).

3. Benchmarking Methodology

Using the above combinations of compilers, GbE switches and MPI libraries, we have benchmarked the widely used IMB (formerly PMB) suite from Intel. IMB is easy to build and run and has become the *de facto* standard for testing the quality of MPI libraries and system interconnects. The data presented in this paper for MPI functions such as `MPI_Allgather`, `MPI_Allreduce`, `MPI_Alltoall`, `MPI_Reduce_scatter` and

`MPI_Sendrecv` at 16, 32 and 64 processes will be shown to clearly differentiate the performance of the various setups under test, and give an indication of the factors that users of MPI applications on GbE clusters need to take into account. Justification for this approach and set of choices is given in paper I. Data for all the MPI-1 functions included in the IMB suite will be available within the group's DBD database
(see http://www.cse.clrc.ac.uk/disco/dbd).

4. Analysis of Results: Methodology

It is clear that 7 flavours of MPI library/compiler combination, 5 MPI functions, 3 different process counts, 6 (and now 9) Ethernet switches and either 22 or 24 different data points for each class of test (depending on MPI function) will result in far too much data for the reader to easily analyse in graphical form. Nevertheless, for the interested reader we have produced log-log plots of average time per MPI function call vs. message size for the above set of tests, for both 2.4 and 2.6 kernels, and these graphs can be found in a separate document (http://www.cse.clrc.ac.uk/disco/gbe_perf.shtml).

For the purposes of this ongoing study, we have devised a scheme that attempts to carry out a balanced and fair averaging of performance for each MPI function tested cf. a baseline metric, thus condensing the data into a more digestible form that allows us to draw conclusions more readily. Below, we re-iterate the scheme presented in paper I:

- For each test (MPI function) we assign an equal weighting to all messages tested, i.e. no one message size is deemed more important than the others. In a multi-user environment with a variety of applications, and various data sets run over time (resulting in a range of message sizes) this would seem to be a reasonable approach. It is unlikely that a cluster would be bought for exclusive use by one user/application running very similar datasets (a very narrow range of message sizes) over the lifetime of its service.

- Pick a baseline configuration with which to normalize the data. We have chosen the Extreme Summit48si (a fairly typical switch), LAM-MPI (widely used as it has a reputation of being the best performing free implementation), and the PGI compiler (popular with users of Opteron-based platforms).

- For each system setup and each message size, compute a ratio of "baseline result"/"setup result". Average performance by taking the geometric mean (the n^{th} root of the product of n values) of these ratios for the range of message sizes tested. (We have limited the range of message sizes to 4 bytes upward due to very small or exactly zero timings being returned by IMB in some cases). Geometric means were also taken across all processor counts generating a single mean value for each switch/compiler/MPI combination (See Figures 1-5). The justification for using the geometric mean approach of paper I and in the extended method for summarising results presented below, rather than other means such as the arithmetic mean, is discussed in section 5 below.

- However, this approach as it is could in principle award a biased higher scoring to a configuration that exhibits evidence of potentially severe performance problems, for example, TCP congestion collapse, at certain message sizes. The symptoms are rapidly varying, spiky log-log plots of average time vs. message-size. More reasonable behaviour and performance over the test range should result in smooth slowly varying plots rising gradually with increased message size.

We feel this is an issue because we have carried out a number of experiments at and around data points that exhibit particularly sharp peaks and troughs on the log-log plots. Using command line options, it is possible to make IMB run a set of user defined message sizes instead of the default set of powers of 2 increases. We have observed, for some switch/library/compiler combinations, extremely rapid and erratic changes in timings, at and immediately either side of, certain message sizes. We feel that configurations that exhibit this behaviour should be penalised, as it is possible that a user will experience severe performance degradation if they stray outside of certain message size ranges through running a slightly different data set, sometimes by as much as several orders of magnitude!

- Therefore we have imposed a further criterion. Firstly for each message size we determine the minimum result (across all configurations tested). Runs for the standard IMB power of 2 progression in message size, which contain results that exceed this minimum by more than an order of magnitude, are then highlighted in a red/orange colour as a health warning in the graphs presented at the end of this paper. (This is of course an arbitrary metric, and further experience may result in a more lax or stringent cut-off approach.)

To further aid both the capture of the true performance characteristics of a particular setup, and the interpretation of results, we propose that the data be rendered further:

- We include minimum and maximum geometric mean data for each switch. These data represent the minimum and maximum geometric mean calculated across all MPI/compiler combinations, and all process counts, resulting in one minimum and one maximum value for each switch for each of the 5 IMB tests. A geometric mean has also been calculated across MPI/compiler combinations and all processor counts. The minimum, maximum and geometric mean data described here are plotted in Figures 1-5. These figures convey the range of results across a particular IMB test (MPI function) for each switch and make it easier to directly compare switch performance cf. the methodology presented in paper I. For completeness and comparison, in Figures 6-10 we re-present the graphical summaries from paper I with additional entries for the three new switches we have evaluated since.

5. Analysis of Results: Discussion

5.1 The Geometric Mean: The Correct Way to Summarise Benchmark Results?

It can be tempting to summarise benchmark results with a single number in order to simplify the process of drawing conclusions about the systems under test. Often this can lead to misleading results, especially when the summary statistic used is inappropriate for the original data.

The pitfalls associated with summarising benchmark results with a single number are discussed at some length by Smith [2] who suggests that the arithmetic mean should be used to summarise performance data expressed as a time, while the harmonic mean should be used to summarise performance data expressed as a rate. Smith offers no use for the geometric mean although Mashey [3] highlights the fact that it is used extensively in established benchmark suites to summarise performance data that has been normalised relative to a given system.

Mashey also indicates that the geometric mean is the correct mean for summarising log-normally distributed data (the lognormal distribution often provides a good fit for data that

has been normalised in the form of a ratio as all data will be positive and is usually positively skewed).

Furthermore, Fleming and Wallace [4] present the geometric mean as the only valid mean for summarising normalised data and go on to provide a proof of this claim. This proof provides the justification for using the geometric mean to produce the summary statistics used in Figures 1-10. A simpler example of why the geometric mean should be used for normalised data (in the form of a ratio) is shown below.

	System A	System B	System C
Run-time in secs	40000	4000	400

Table 1: Example benchmark results.

	System A	System B	System C
Normalised run-time	0.1	1	10

Table 2: Normalised results with system B as the baseline.

Consider the benchmark results in table 1. In table 2 the same results have been normalised with system B as the baseline. The values in table 2 represent the performance ratio for each system relative to system B. If we take the arithmetic mean for system A and system B we get the following:

$$\frac{\sum_{i=1}^{n} x_i}{n} = \frac{0.1+10}{2} = 5.05$$

This arithmetic mean tells us that the mean normalised performance of systems A and C is 5.05. Now consider the geometric mean:

$$\left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}} = \sqrt{0.1 \times 10} = 1$$

This geometic mean tells us that the mean normalised performance of systems A and C is 1. Remember that the original results are in the form of a performance ratio relative to system B. System A has a value of 0.1, which can be understood to mean ‘ten times slower than system B’, while system C has a value of 10, which can be understood to mean ‘ten times faster than system B’. One would expect a mean ratio of system A and system C to be half way between ‘ten times slower’ and ‘ten times faster’, in other words ‘The same speed as system B’. The value of 1 given by the geometric mean represents ‘The same speed as system B’. The value of 5.05 given by the arithmetic mean is actually just a value half way between 0.1 and 10. It does not represent anything useful when considered as a performance ratio relative to system B.

As well as covering the uses of the various means, Smith [2] also discusses the normalisation of data and concludes that if performance is to be normalised with respect to a specific machine, then an aggregate performance measure (such as total time) should be calculated before any normalisation of the data takes place. This is the method used to generate Figures 11-16. Total run-times are calculated and then the totals are normalised to give a relative measure of performance for each switch across multiple processor counts and MPI functions.

5.2 New/Further Findings

Examination of Figures 1-16 provides a number of further conclusions to those presented in paper I (note: based on performance alone and without consideration of the price of these products):

- Figures 1-5 clearly show the variation in switch performance over each of the five MPI functions focused on here. According to these results the HP Procurve 3500yl is the switch that performs most consistently across all five MPI functions as it is the only switch to rank in the top 3 for all five tests. It is a considerable improvement over the 2848, which, along with the Force10 S50, is at the other end of the performance scale.

- As concluded previously in paper I, most of the configurations in Figure 6 are shown to be inadequate. LAM-MPI performance is acceptable up to 32 processes (in terms of passing the 'smoothness' test defined in paper I if not always in absolute performance terms) for the Netgear and Cisco switches. Clearly the best configuration is still the Scali MPI on the Cisco switch, with the Nortel and Extreme switches offering similar performance. LAM-MPI with all three compilers does rather well on the Cisco switch only. In general though, we can conclude, as before, that `MPI_Alltoall` is one test that does 'sort the men from the boys'. The Extreme and Nortel switches show clear differentiation between the commercial/TCP bypass (Scali) and free/TCP-based (MPICH and LAM) MPIs.

- As in paper I we note the trend across Figures 1-10, namely that of virtually overlapping performance of the Nortel 5510, Extreme Summit48si and now Nortel 5520. Casual inspection suggests the numbers are identical – they aren't – and we are confident that our analysis has been carried out without error. This similarity in performance can in part, we believe, be traced to commonality of components – ASICs, backplanes etc. We shall discuss architectural features in more depth in subsequent studies. However, Figures 11-15 clearly illustrate that there are differences between the performance of the Extreme Summit48si switch and the two Nortel switches, particularly for 64 processor runs where the Summit48si performs relatively poorly. In fact, Figure 16 shows that the overall performance of the Netgear GS 748T is very similar to that of the three switches mentioned above.

- Performance for the x450-24t is competitive with the Cisco switch and the HP Procurve 3500yl for 16 and 32 processor runs. As the x450-24t is only a 24 port switch it has not been possible to generate data for 64 processor runs with the cluster configuration used in the two studies so far. In future revisions of this report we hope to be able to present data for 64 processor runs using the 48 port x450-48t, as this architecture does look promising and ought to perform well beyond 32 processes. Indeed, Ladd [5] has carried out MPI benchmarks on a number of switches, the findings of which appear to back up this assertion.

6. Conclusions

In this paper we have presented further IMB data for Gigabit Ethernet switches from a number of manufacturers deployed in 'plug and play' mode. This is part of an ongoing effort that will now turn to focussing on the management of these and other switches with a view to optimizing the performance of synthetic and real MPI applications, and ascertaining at what

point HPC users really do need to abandon GbE for higher priced interconnects such as InfiniBand, Myrinet and Quadrics.

As the volume of benchmark data generated has increased, and therefore the ease of interpretation has decreased, we felt it was necessary to explore ways of refining the geometric mean summarising approach presented in paper I which would result in further rendering down of the data into a more digestible format. These amendments to our original approach result in the performance of the configurations under test being more easily understood. Moreover, there appears to be a degree of statistical rigour underlying the method with further justification provided by the earlier findings of Smith, Mashey et al.

7. Acknowledgments

We are grateful to Arif Ali of OCF plc for continued systems support. We wish to thank further vendors for loaning switches to us and providing technical assistance: Nortel Networks (Martin Wolfenden), HP Procurve (Alan Albrecht, Jeremy Arnold, Wim Groenveld, Nick Hancock), Extreme Networks (Stephen Jamieson).

8. References

[1] http://www.cse.clrc.ac.uk/disco/publications/DL_TR_2006_009.pdf

[2] 'Characterizing Computer Performance with a Single Number', Smith, James E, *Communications of the ACM, Vol 31, 1202-1206 (1998)*

[3] 'War of the Benchmark Means: Time for a Truce', Mashey, John R, *ACM SIGARCH Computer Architecture News, Vol 32, No 4 (2004)*

[4] 'How not to lie with Statistics: The Correct way to Summarize Benchmark Results', Fleming, Philip J and Wallace, John J, *Communications of the ACM, Vol 29, No 3 (1986)*

[5] http://ladd.che.ufl.edu – follow the *Beowulf cluster* link.

Figure 1: Geometric mean across Alltoall runs (Ext/lam/pgi baseline)

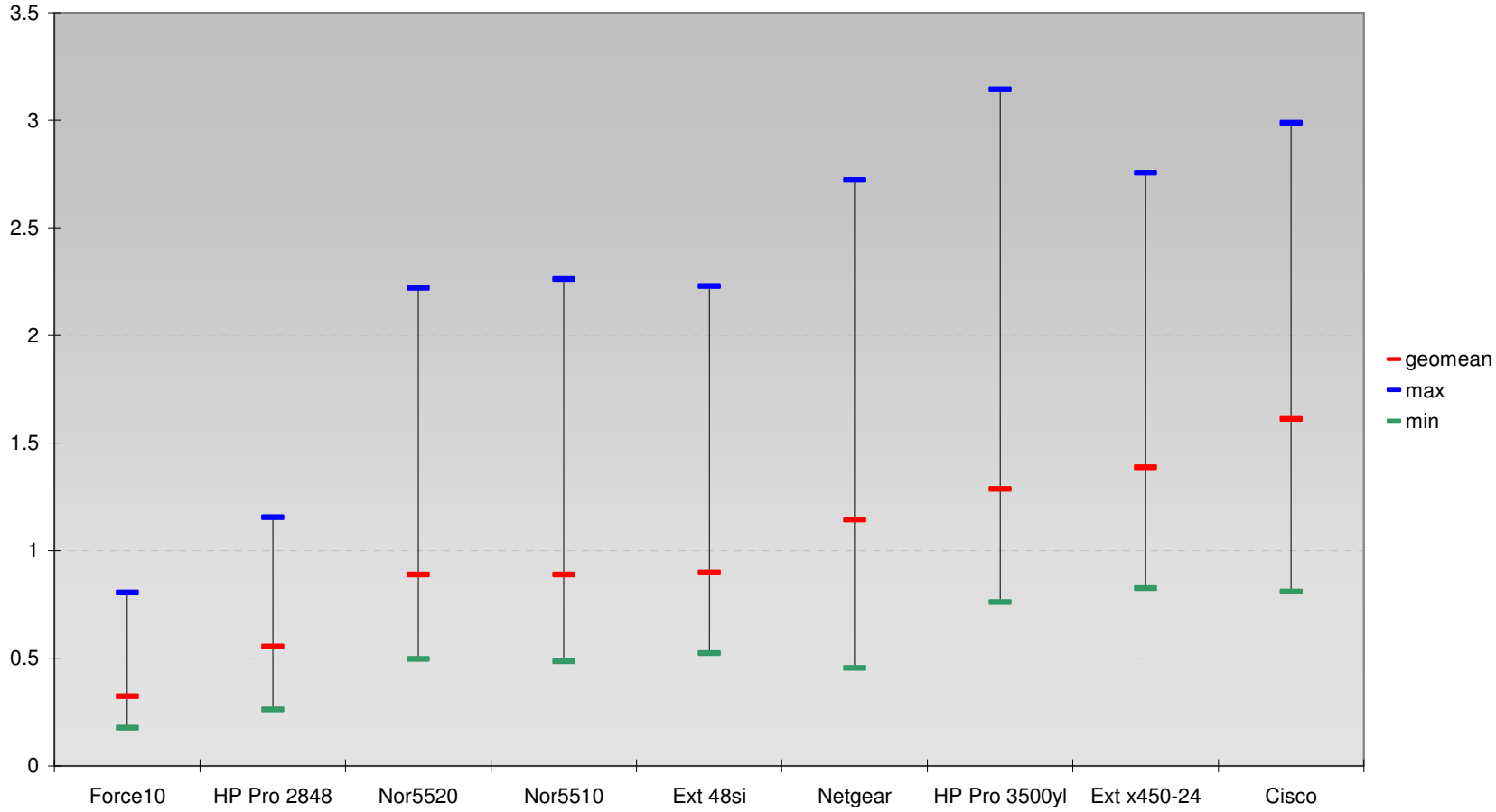


Figure 2: Geometric mean across Allreduce runs (Ext/lam/pgi baseline)

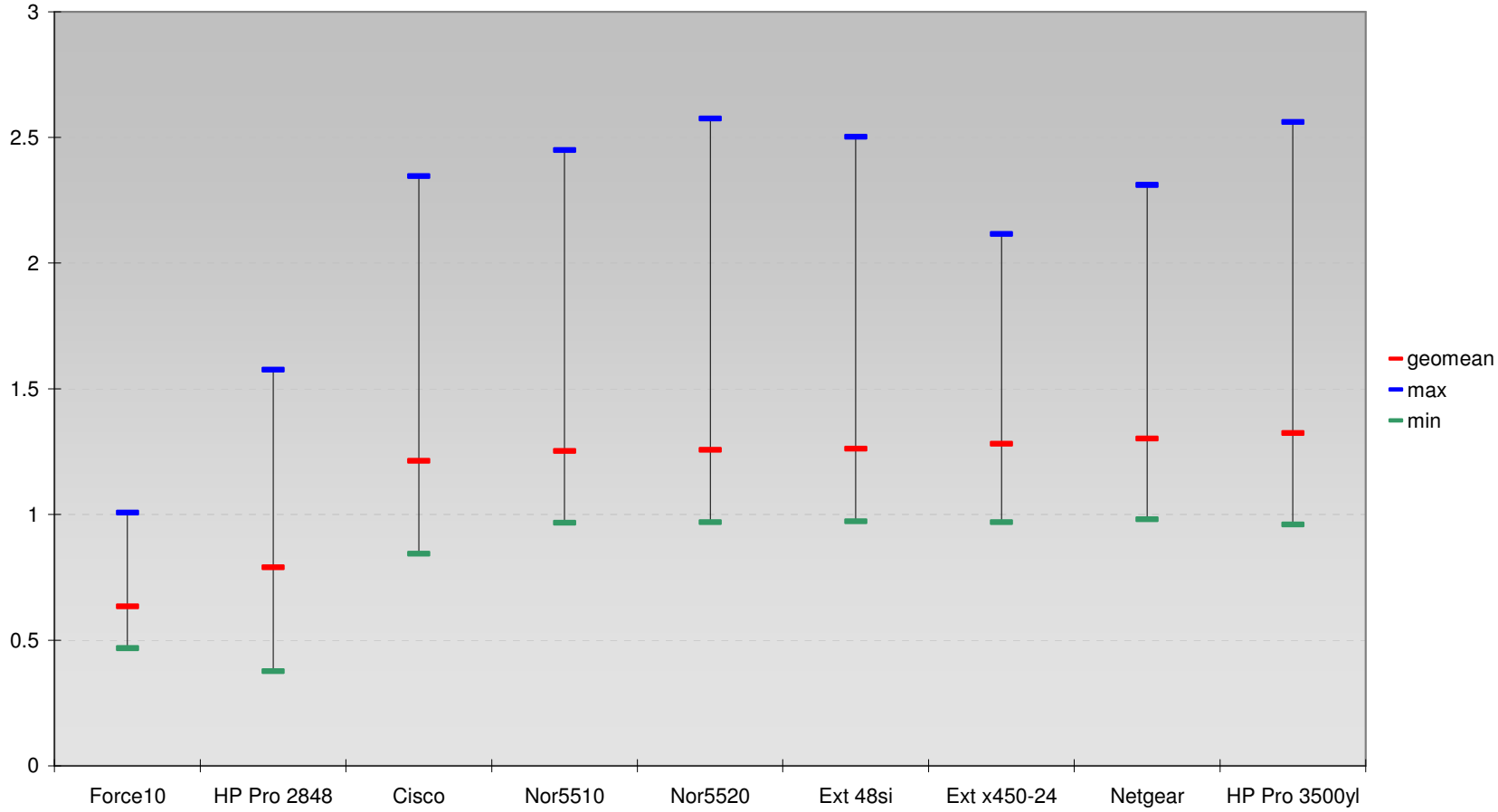


Figure 3: Geometric mean across Allgather runs (Ext/lam/pgi baseline)

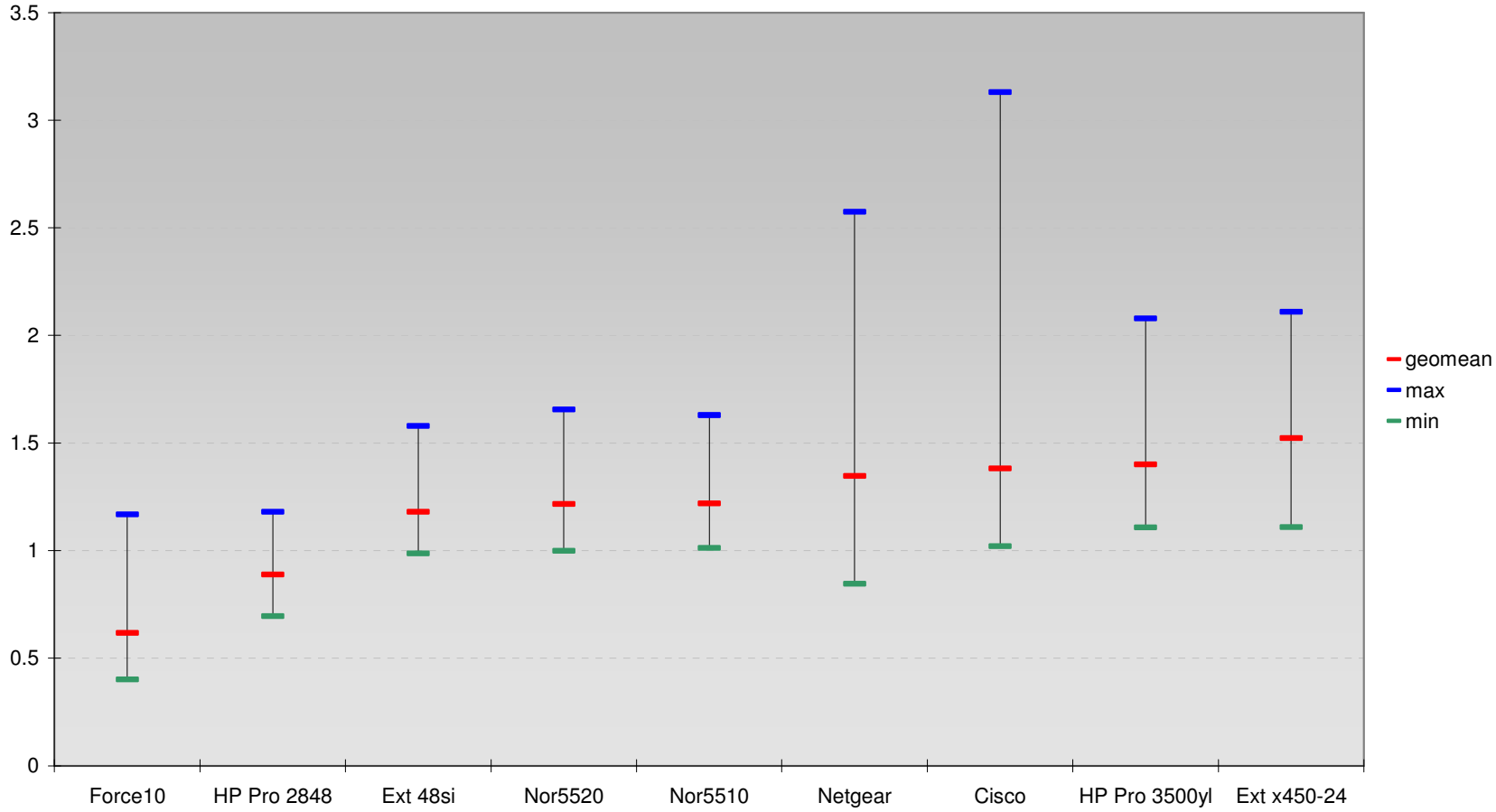


Figure 4: Geometric mean across Reduce_scatter runs (Ext/lam/pgi baseline)

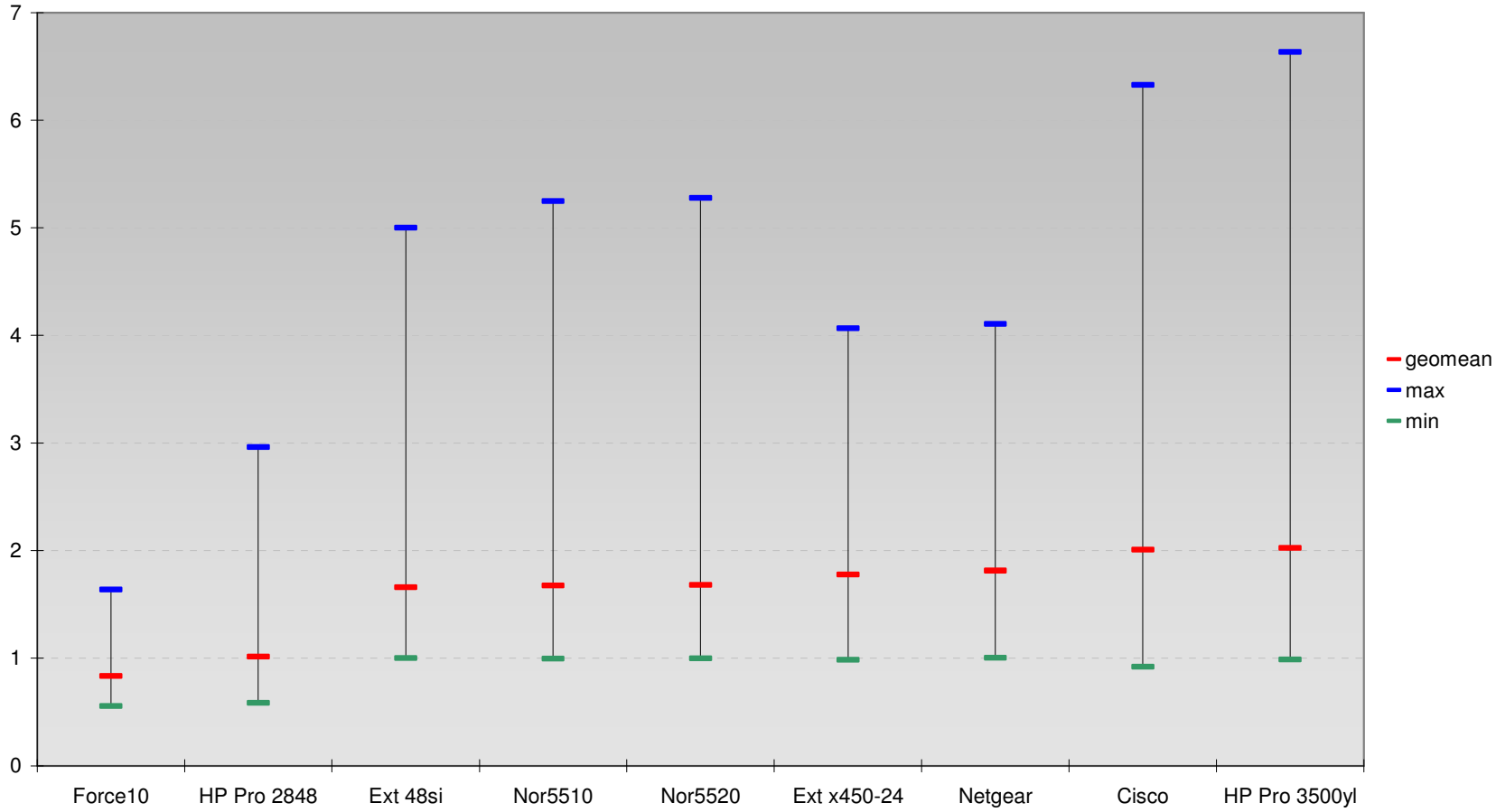


Figure 5: Geometric mean across Sendrecv runs (Ext/lam/pgi baseline)

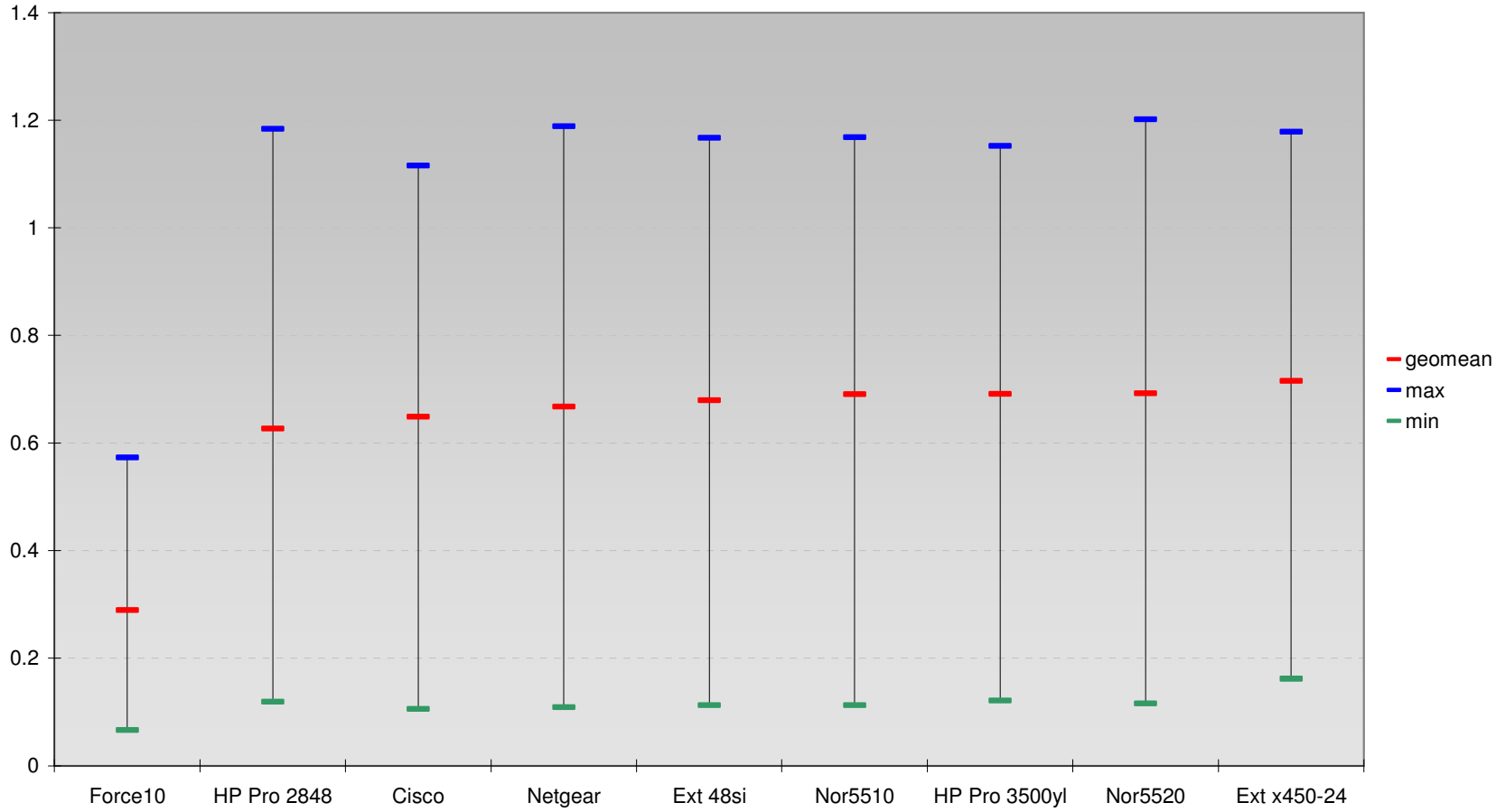


Figure 6: Alltoall geometric mean (Ext/lam/pgi baseline)

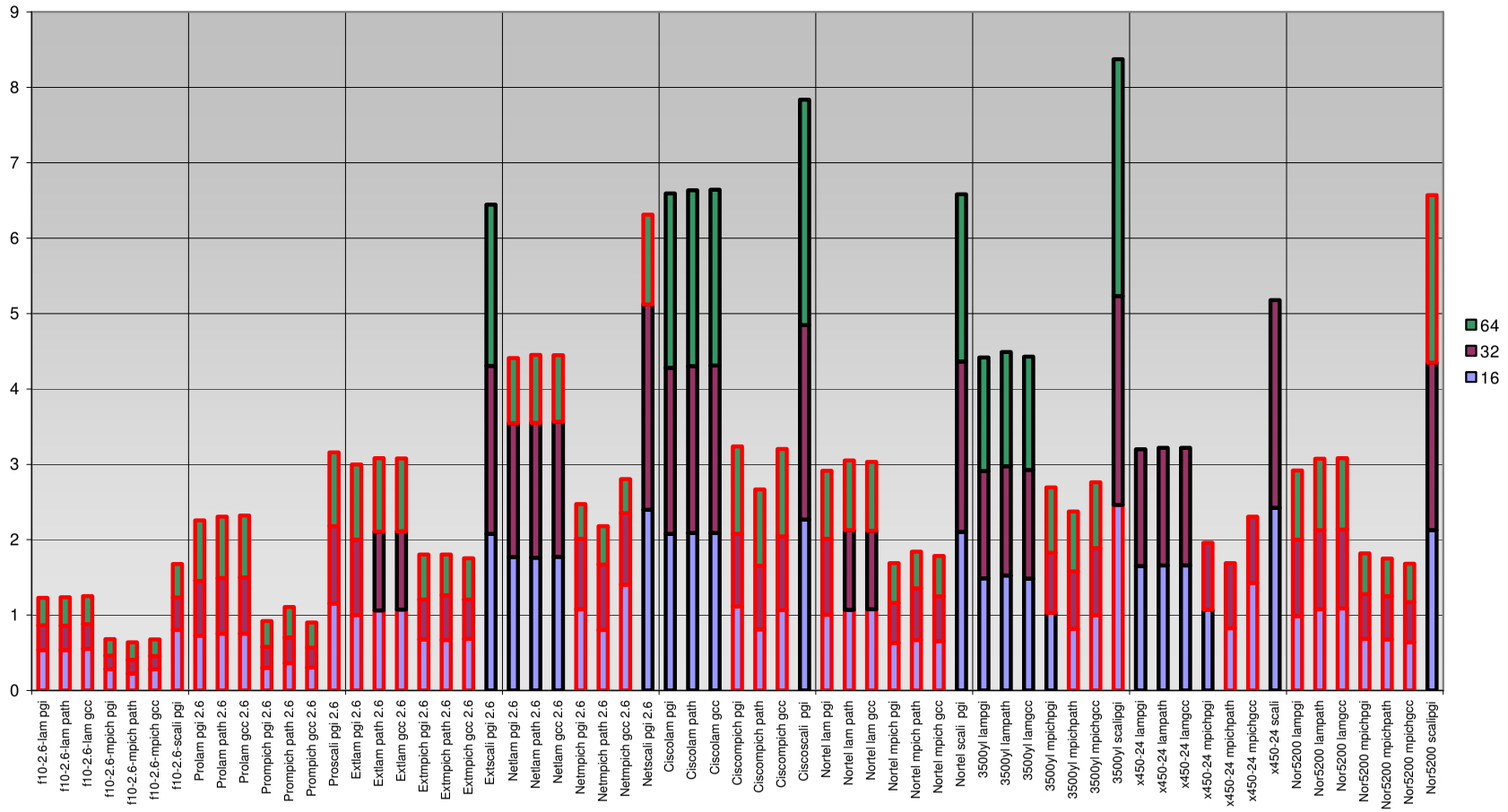


Figure 7: All_reduce geometric mean (Ext/lam/pgi baseline)

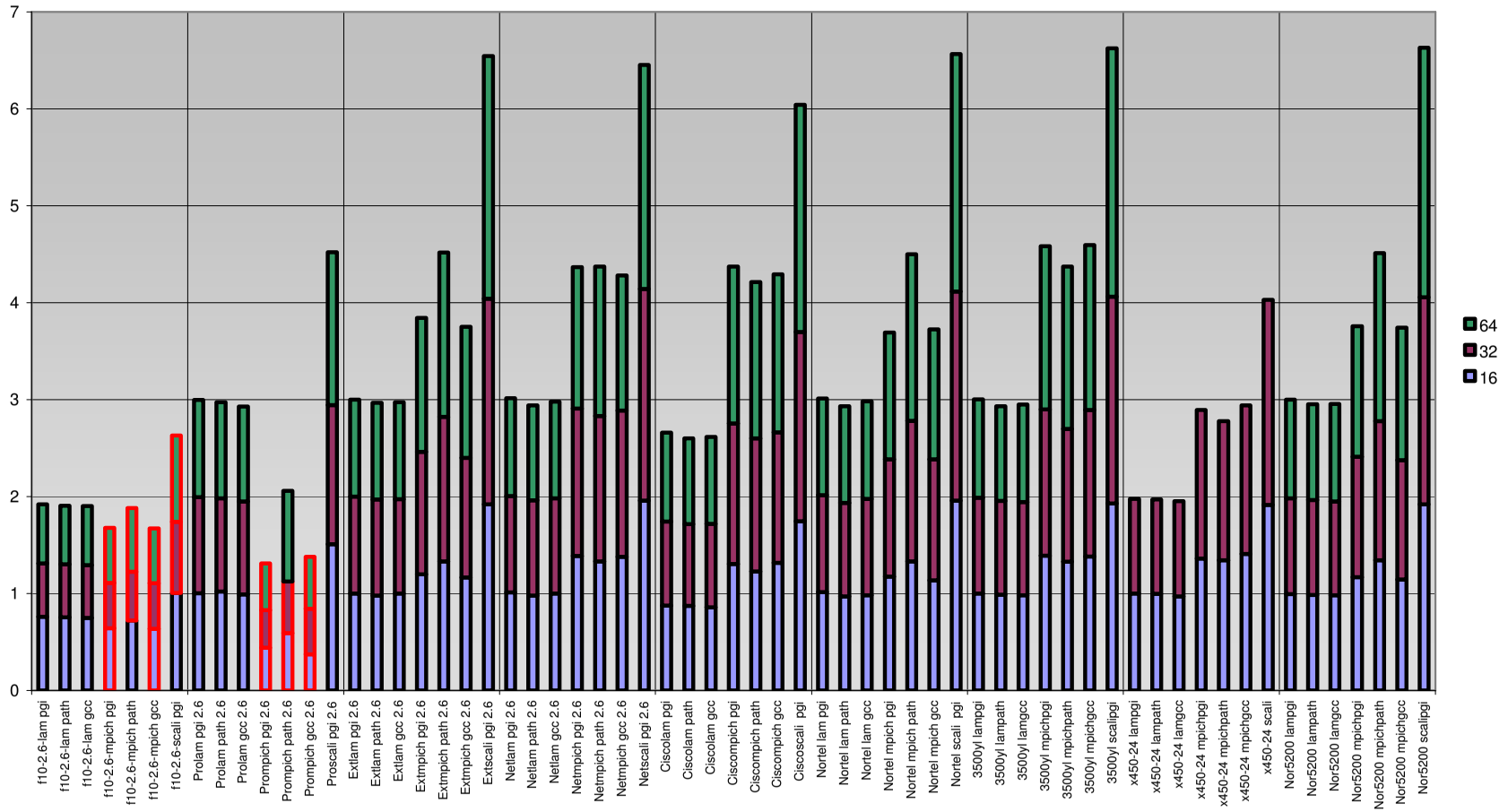


Figure 8: All_gather geometric mean (Ext/lam/pgi baseline)

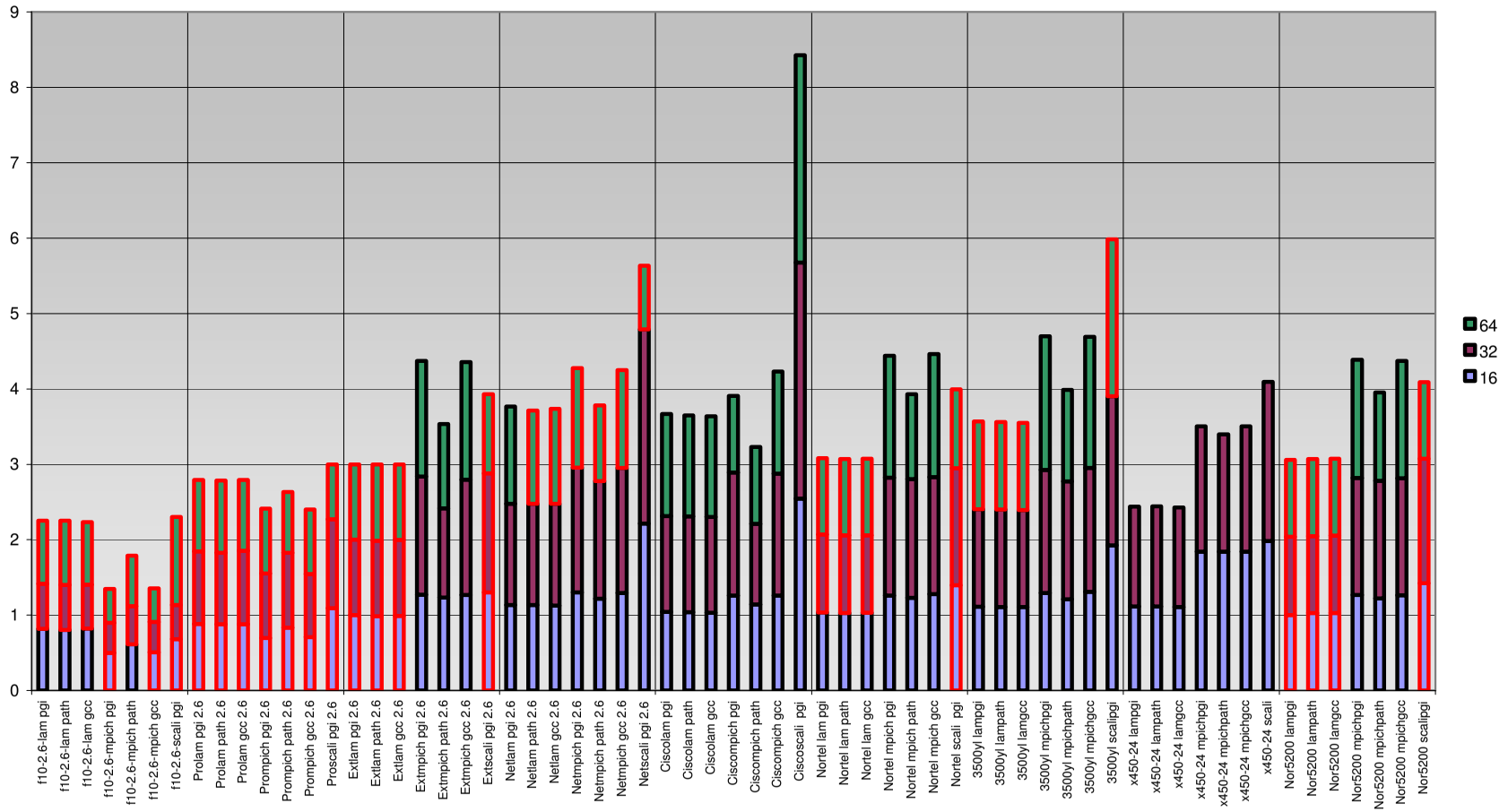


Figure 9: Reduce_scatter geometric mean (Ext/lam/pgi baseline)

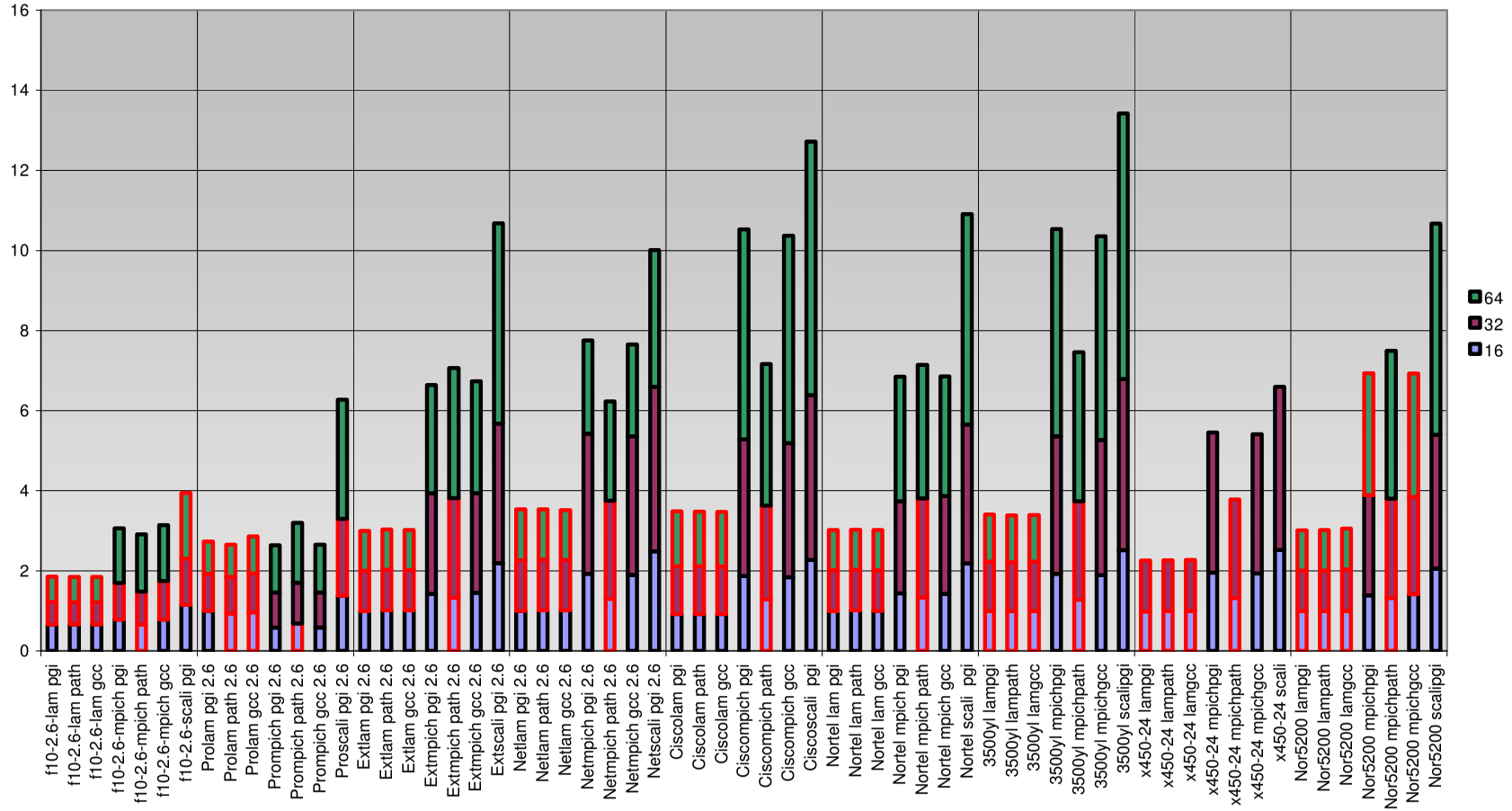


Figure 10: SendRecv geometric mean (Ext/lam/pgi baseline)

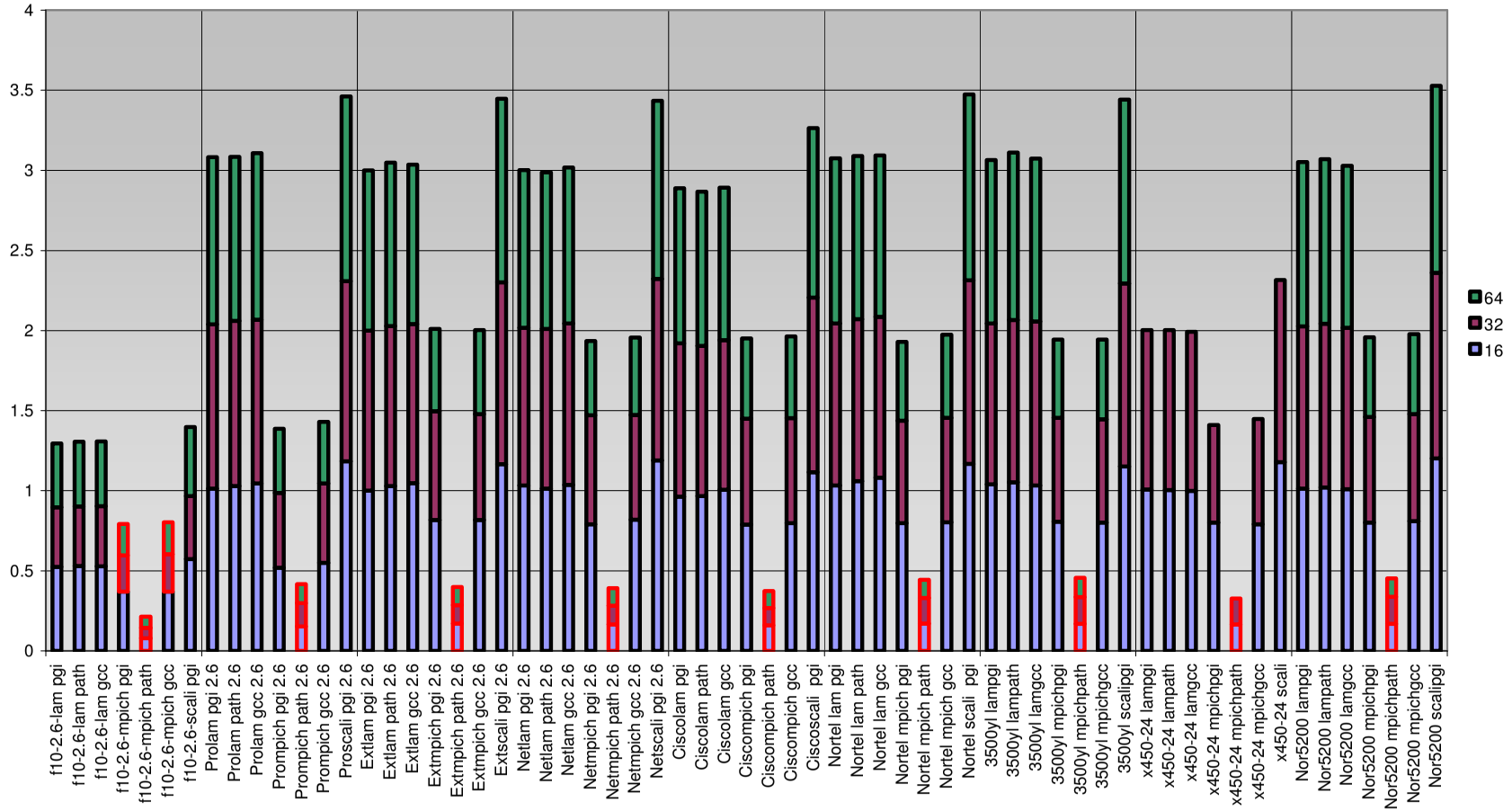


Figure 11: Aggregated runtime Alltoall

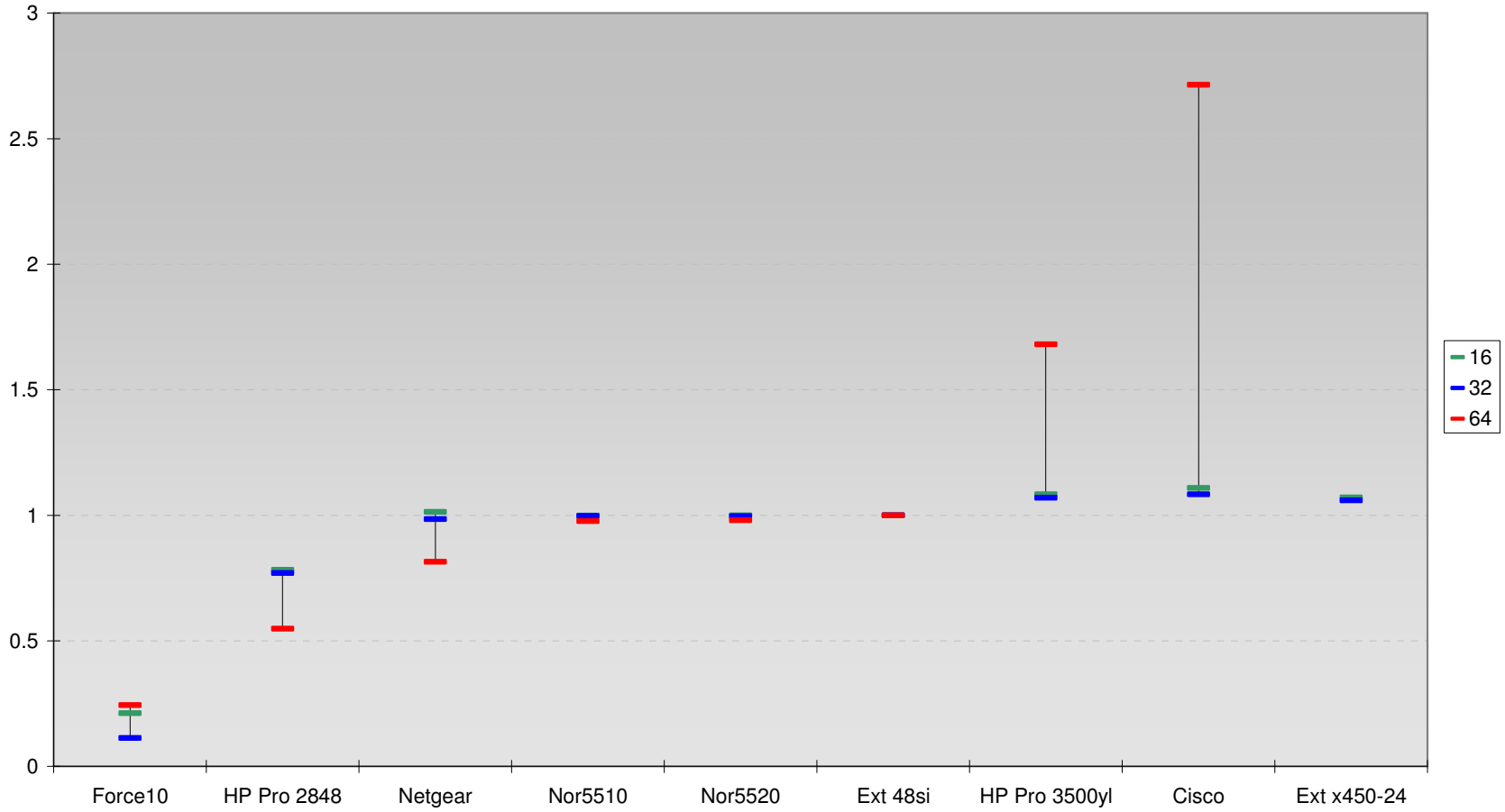


Figure 12: Aggregated runtime Allgather

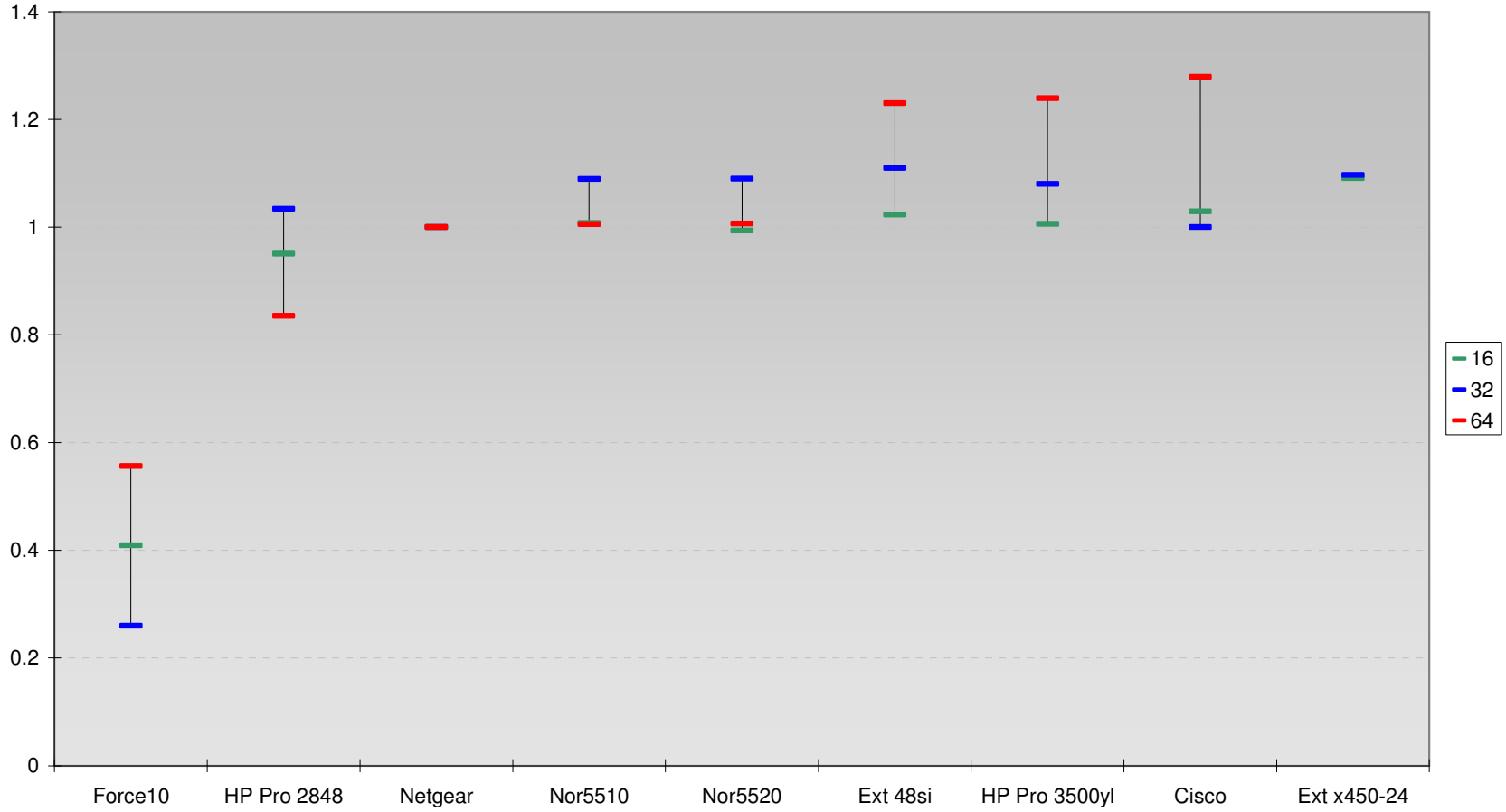


Figure 13: Aggregated runtime Allreduce

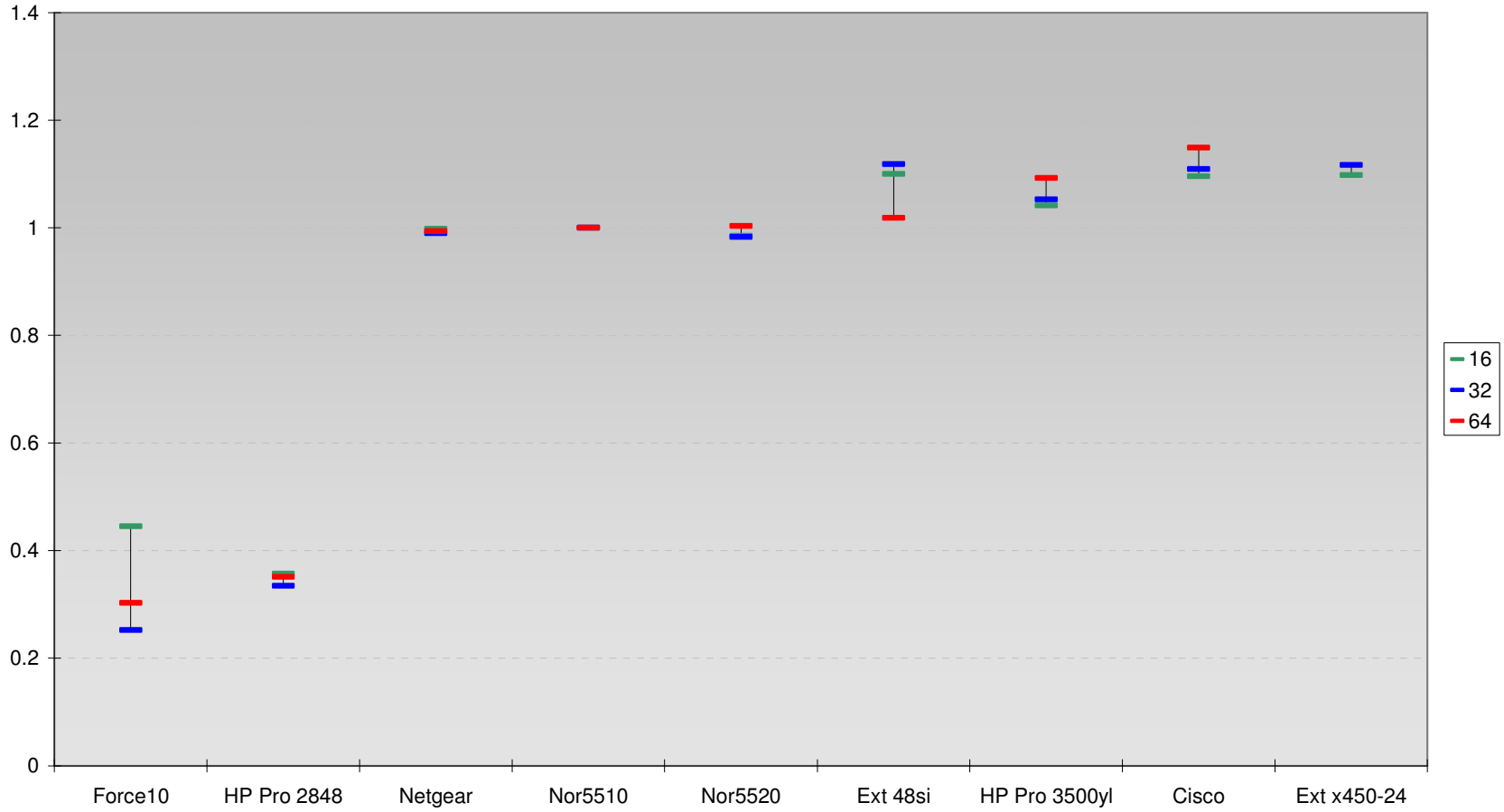


Figure 14: Aggregated runtime Reduce_scatter

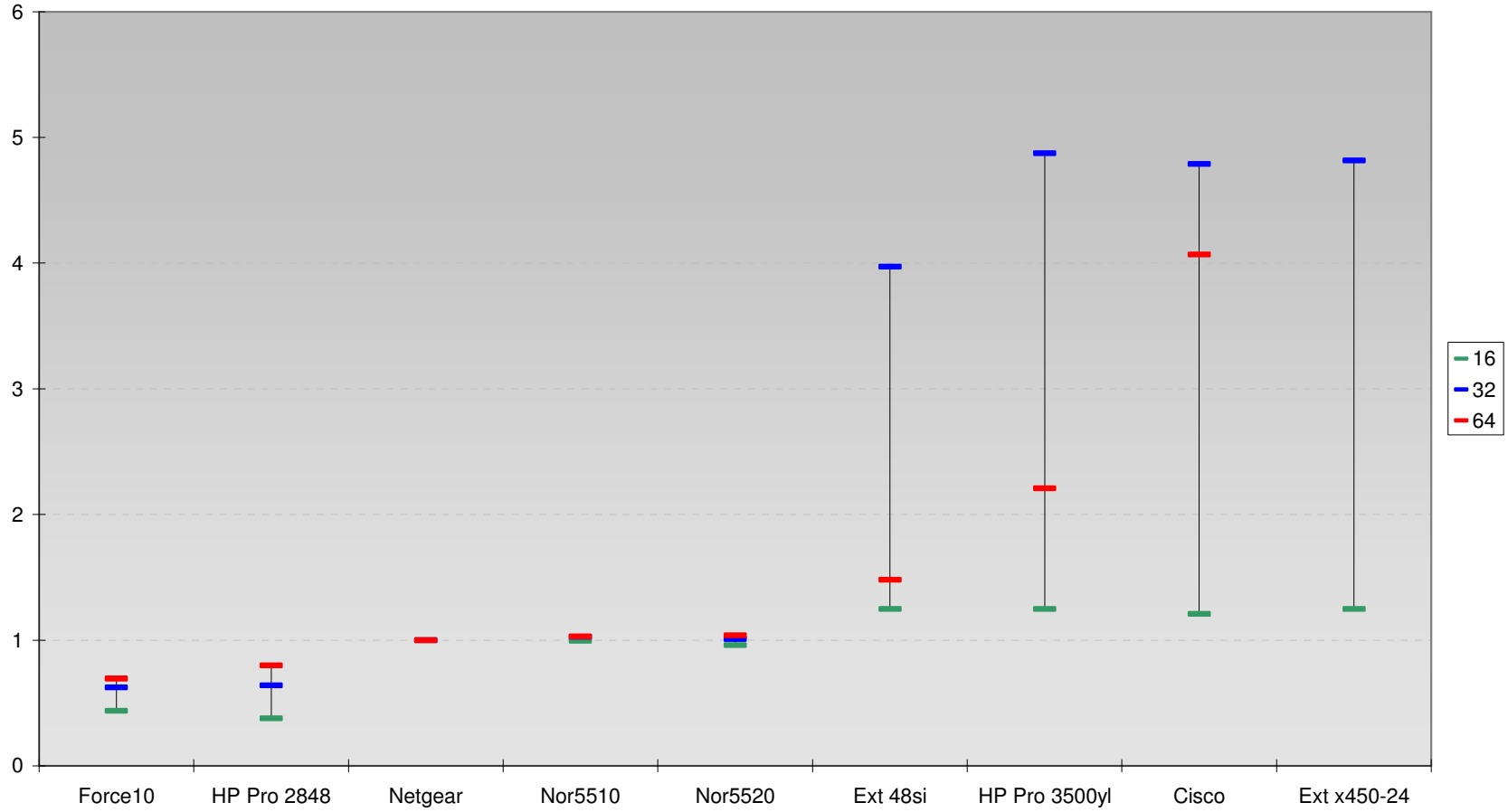


Figure 15: Aggregated runtime Sendrecv

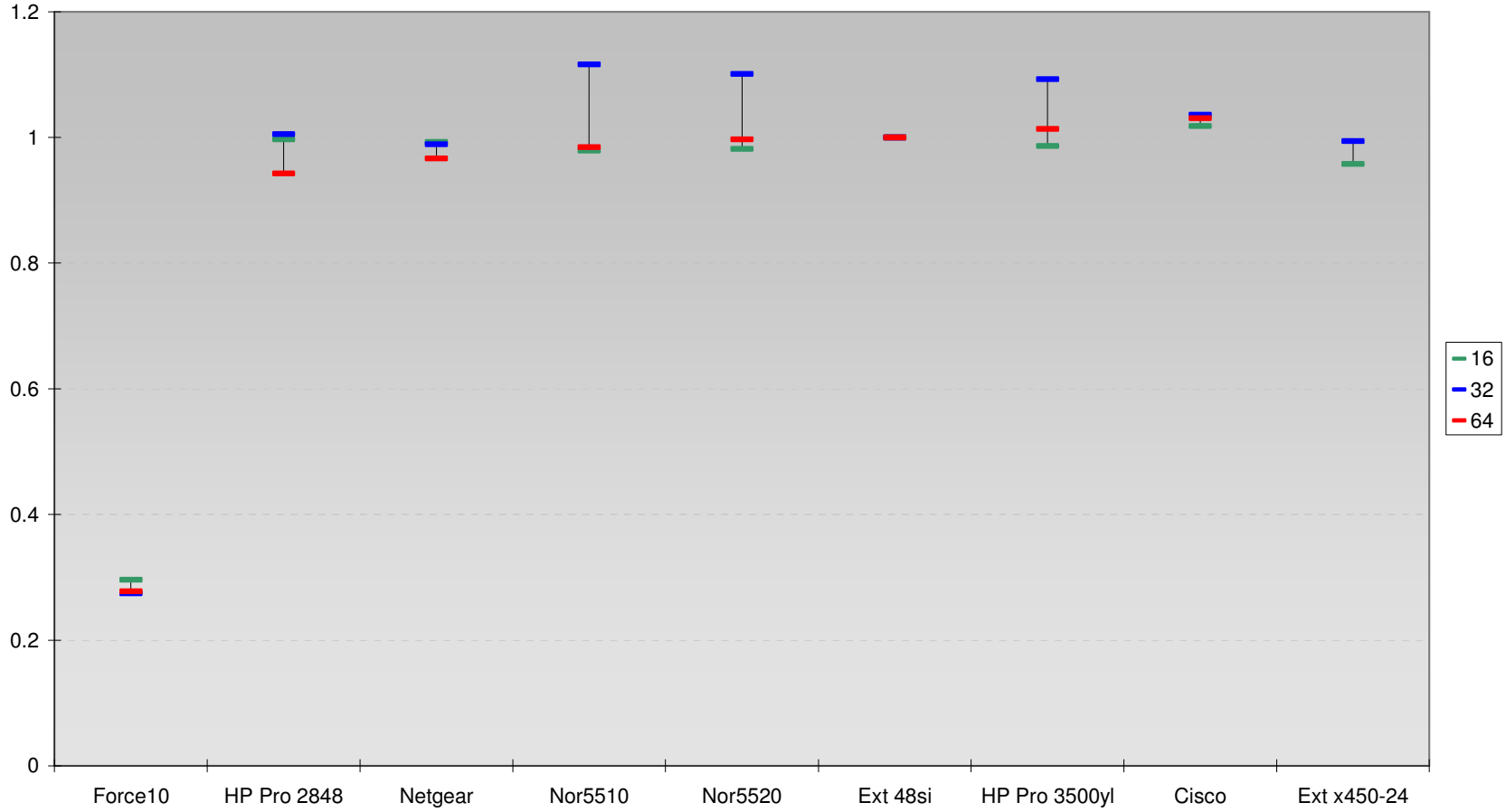
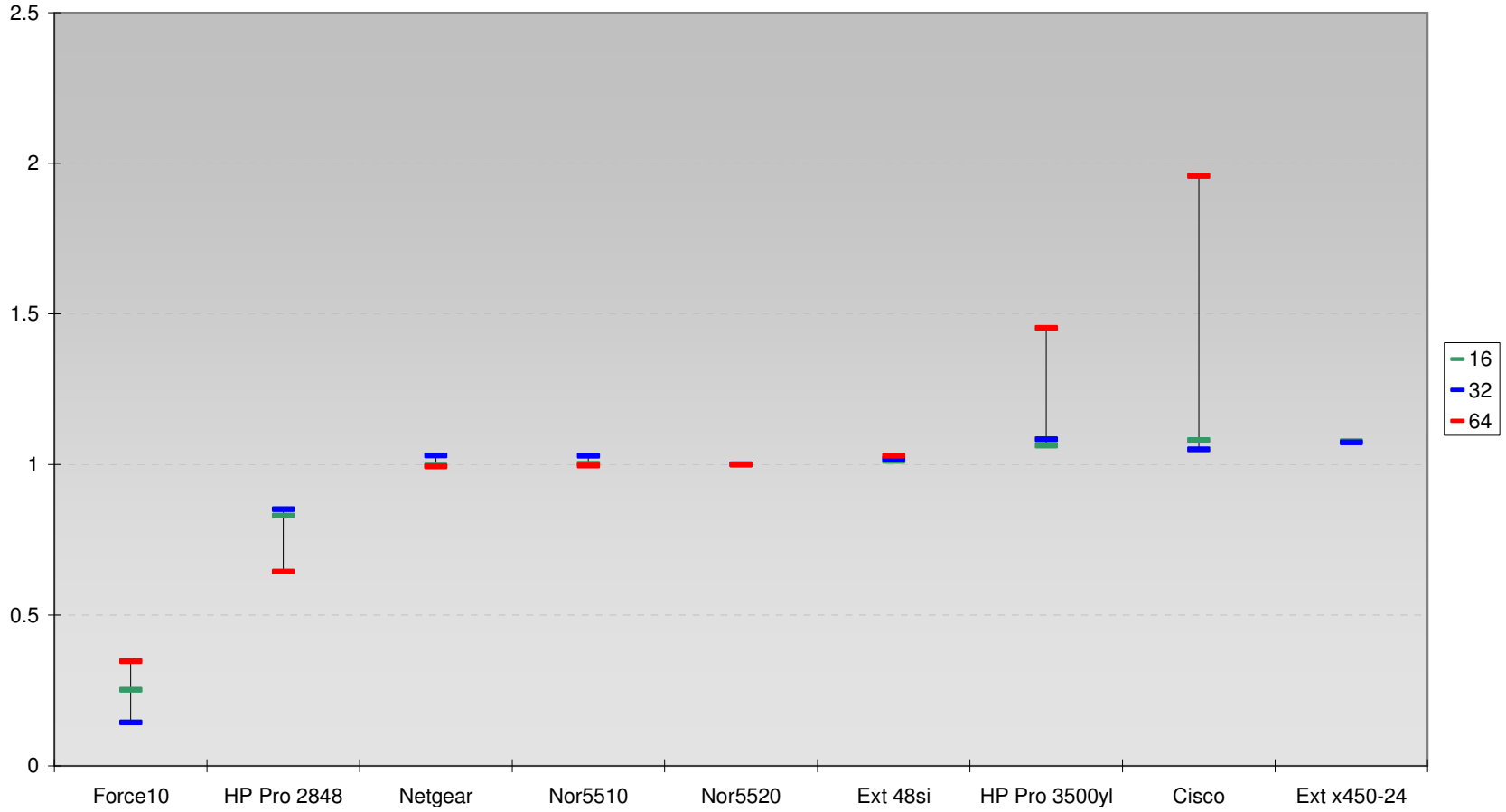


Figure 16: All MPI functions



Council for the Central Laboratory of the Research Councils
Chilton, Didcot, Oxfordshire OX11 0QX, UK
Tel: +44 (0)1235 445000 Fax: +44 (0)1235 445808

CCLRC Rutherford Appleton Laboratory
Chilton, Didcot,
Oxfordshire OX11 0QX
UK
Tel: +44 (0)1235 445000
Fax: +44 (0)1235 44580

CCLRC Daresbury Laboratory
Keckwick Lane
Daresbury, Warrington
Cheshire WA4 4AD
UK
Tel: +44 (0)1925 603000
Fax: +44 (0)1925 603100

CCLRC Chilbolton Observatory
Drove Road
Chilbolton, Stockbridge
Hampshire SO20 6BJ
UK
Tel: +44 (0)1264 860391
Fax: +44 (0)1264 860142