DL-TR-2006-009

CCLRC

Gigabit Ethernet – an HPC interconnect? I: Findings from synthetic benchmark studies

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

October 2006

Council for the Central Laboratory of the Research Councils

Enquiries about copyright, reproduction and requests for additional copies of this report should be addressed to:

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?
I: Findings from synthetic benchmark studies

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.

For mid-range/capacity HPC, clusters of servers based on commodity technologies (x86 typically) and commodity interconnects such as Gigabit Ethernet are widely deployed. Whilst higher performance (and higher price) interconnects such as Myrinet have been shown to be essential for some classes of HPC applications, it is thought that Gigabit Ethernet is an adequate, cost-effective approach for a wide range of codes.

Our investigations show that the performance of MPI applications on typical Gigabit Ethernet clusters can vary dramatically depending on a number of factors concerned with choice of hardware and software, and that in some cases performance can be so poor and unpredictable, at even modest process counts, that a more sensible and cost-effective strategy may be to invest either in proprietary MPI libraries, very specific Ethernet switch architectures, or higher performance interconnects instead.

[*] Distributed Computing Group, CCLRC Daresbury Laboratory, Warrington, WA4 4AD, UK.
[**] IBM UK Ltd, ATS Dept., MP 9x, Inverkip Road, Spango Valley, PA16 0AH, UK.
[§] Present Address: Barclays Capital, 5 The North Colonnade, Canary Wharf, London E14 4BB, UK.

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.

During this period the HPC community – users and vendors – came together to define and implement, and then refine and expand on, a standard for the message passing model for distributed memory parallel computers, namely MPI-1[1] followed by MPI-2[2]. Coupled with these developments, the advent of free, or relatively cheap, open source operating systems (Linux in particular) and associated compilers, libraries, middleware etc. has led to HPC sites procuring systems with increasingly lower up-front capital costs for a given level of *peak* performance. (Whether or not such facilities improve scientific productivity and total cost of ownership (TCO)/return on investment (ROI) is dependent on a range of other factors and requires detailed further analysis.)

These trends are reflected in the statistics for entries in the '*top 500'* list[3]. The earliest entry on the list for a GbE system was November 2001 with 1 machine (0.2% of the list). The latest list (June 2006) has 256 GbE entries (51.2%). During this period we have seen market share gains followed by losses for both Myricom and Quadrics. Near-future lists are likely to feature a few systems at the top end with proprietary interconnects (e.g. Cray's XT3 and follow-on, IBM's Blue Gene etc.), and the vast majority of systems deploying GbE (and increasingly10 GbE) or InfiniBand, with Quadrics and Myricom perhaps set to lose further market share (although the 10 GbE interoperability their recent products offer may help stave off decline).

Similarly the adoption of Linux has been rapid and has gone from 7.8% of systems in November 2001 to 73.4% of systems in June 2006. Statistics for day-to-day usage of these systems are not as informative, due to a lot of sites not wishing to disclose this information for a variety of reasons – commercial and classified – with 63.2% of systems deployed for unknown applications. Therefore it is hard to draw conclusions as to how the GbE clusters featured on the list are used in production mode, and whether or not the production applications run on them are effectively a stress test for the GbE interconnect.

A considerable amount of benchmarking data has been produced for GbE clusters previously, and attempts have been made to understand factors that influence performance and to tune the appropriate system components. For the sake of brevity, we will not review all of this work here, but remark that these studies have tended to have a different or more narrow scope than the present effort. For example, that of Celebioglu et al.[4] concentrates on comparing performance of three GbE adaptor device drivers but with only one flavour of MPI library (MPICH-1.2.5), a single GbE switch and kernel version. Other studies such as that of Vinter[5] concentrate on comparing three implementations of MPI: LAM, MPICH and MESH-MPI[6], but without exploring different kernel versions, compiler types etc. Vinter's paper does attempt some TCO/ROI analysis and provides estimates of the intrinsic value a good quality MPI implementation can bring per server.

---

[1] http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html
[2] http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html
[3] http:/www.top500.org
[4] http://www.linuxclustersinstitute.org/Linux-HPC-Revolution/Archive/PDF04/23-Celebioglu_O.pdf
[5] http://wotug.org/paperdb/send_file.php?num=113
[6] http://www.meshtechnologies.com/text/PDF/MESH-MPI_White_Paper.pdf

Some previous efforts carried out in this area have ignored collective operations and concentrated on maximizing performance of the marketing numbers (PingPong latency and asymptotic bandwidth), and/or not included high enough process counts (it is difficult to draw useful conclusions from benchmarks carried out on just a handful of nodes).

Of those studies which have looked at the performance of collective operations for more than a handful of nodes, the conclusions have tended to suggest that closely-coupled (communication intensive) applications will not scale on Gigabit Ethernet connected clusters much beyond 32 and certainly 64 processes, and users with such needs should invest in alternatives such as Myrinet, Quadrics, or in more recent times, InfiniBand. One aim of the present study is to recheck these assertions.

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

This study will not include an assessment of recent GbE technologies which promise lower latencies (which implies improved performance for the MPI functions we are testing). Examples of such technologies include the Level5 Networks/Solar Communications' EtherFabric PCI-X/PCIe NIC (which will of course incur increased capital outlay). We may go on to look at such adaptors in a future study, but it is far from certain whether such approaches will be accepted by the market. Already we have seen the demise of Ammasso Inc., a company with offered an offload technology to provide Ethernet with a RMDA capability.

We intend to monitor further developments in the area of RDMA for Ethernet, the most promising probably being the iWARP approach[7], now part of the Open Fabrics Alliance effort[8] which aims to drive the adoption of common software stacks in the Ethernet and InfiniBand arenas. Right now however, the technology is probably a little immature for benchmark assessment purposes.

In terms of MPI libraries we have examined widely used open source implementations – MPICH[9] and LAM[10] – along with a commercial, proprietary alternative – Scali MPI[11]. These are of course not the only options available to us, but time and resource constraints dictate that for now our efforts be directed towards assessing these libraries. Future efforts will rectify this and look the likes of Open MPI[12] in particular as it is built from elements of LA-MPI, LAM-MPI, FT-MPI and PACX-MPI and will eventually supersede these. Additionally we acknowledge the potential performance gains due to lower latencies offered by the likes of the SCore MPI libraries[13] and OS-bypass approaches such as GAMMA (The Genoa Active Message MAchine)/MPI[14]. We will rectify this in a future study, and examine SCore MPI in particular.

We will in future also provide data for benchmarks carried out on more up to date hardware including clusters based on Intel platforms, especially as Intel produces its own GbE adaptors and associated drivers – factors that are sure to affect performance. However, as the HPC

[7] http://www.iol.unh.edu/services/testing/iwarp/
[8] http://www.openfabrics.org/
[9] http://www-unix.mcs.anl.gov/mpi/mpich1/
[10] http://www.lam-mpi.org/
[11] http://www.scali.com/
[12] http://www.open-mpi.org/
[13] http://www.pccluster.org/
[14] http://www.disi.unige.it/project/gamma/

market's adoption of Opteron-based servers with Broadcom GbE adaptors over the last 2-3 years has been rapid, we feel the present choice of server platform for benchmarking analysis is an appropriate one.

We focus on assessing typical configurations of GbE compute clusters based on 1U 'pizza box' x86/x86-64 servers with on-board generic GbE adaptors, common GbE switches and widely used combinations of MPI libraries and compilers, for a range of MPI collective functions used in a wide variety of applications, and at modest process counts: 16 and 32 (a regime in which GbE performance might be expected to be adequate); and the more taxing – 64 processes.

For now, we have made no attempt to optimize the various system components above. 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. Our focus has 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'. In fact, we will report in a follow-on paper that such optimization efforts are not necessarily straightforward and may be time consuming – work that may ultimately have to be done on a per application basis.

In addition we will in the follow-on paper on how 'real-world' applications' performance correlates with the findings in this present study. Moreover, in a further paper we will report our findings from an in-depth study of the statistical properties of MPI network traffic at the packet level and its implications for performance.

In the present study we 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. We have produced data for two versions of the SuSe OS – 8 and 9 – which are based on the Linux 2.4 and 2.6 kernels respectively. The upgrade was carried out in order to not only keep the system current, patched etc., but to compare how the TCP implementations within the two versions of the Linux kernel affect performance. Due to space constraints we will present a summary of just the 2.6 kernel data, but mention comparisons with 2.4 performance of interest to the results' discussion. All data for both kernels can be viewed in graphical form at http://www.cse.clrc.ac.uk/disco/gbe_perf.shtml. The 2.4 kernel data is not as complete as that for the 2.6 kernel due to some switches only being available to us for short periods of time. In addition we will provide a database of all results (2.4 and 2.6) for this study and others carried out by our group[15] .

The cluster has three interconnects for message passing: InfiniBand 4x/SDR (Mellanox HCAs and switches), SCI from Dolphinics, and GbE. The nodes have dual Broadcom Ethernet adaptors – one used for the message passing network, the other for management, NFS traffic etc. - and two PCI-X slots (100 MHz not 133 MHz).

[15] (see http://www.cse.clrc.ac.uk/disco/dbd)

For this study a range of typical 1U 48 port GbE switches from the following manufacturers were assessed: Cisco Systems (Catalyst 4948), Extreme (Summit48si), Force10 Networks (S50), HP (Procurve 2848), Netgear (GS 748T), and Nortel Networks (5510 48T). We are in the process of compiling additional data for switches from these and other manufacturers, and will report on this in due course. (We may in future extend our studies to larger clusters requiring more than 1 x 48 port switches, and investigate the performance impact of using core/edge switch topologies and the impact on performance of oversubscription.)

A mixture of free and commercial MPI libraries were used in this study: MPICH 1.2.7, LAM-MPI 7.1 and 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 and Scali MPI. MPICH and LAM-MPI were configured to use shared memory within a node and built with recommended compiler options where available.

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[16].

3. Benchmarking Methodology

Previous studies in this area, e.g. that of Chen and Latouche[17] on very similar e325 Opteron-based clusters, have tended to concentrate on establishing the difference in raw network performance for GbE (using TCP) through use of the Netpipe suite[18], and the performance observed when running MPI over GbE. Netpipe's functionality (for now at least) only allows users to probe the performance of point-to-point type communication. For example the difference between raw TCP PingPong bandwidth and latency cf. MPI over TCP bandwidth and latency can be ascertained thus giving an indication of the efficiency of the MPI implementation.

Whilst studies such as these are very useful, we feel that further work is required to complement such efforts, in particular in the area of assessing the performance of more communication intensive collective operations used in a wide variety of HPC applications. GbE offerings must be able to perform adequately, i.e. not show a dramatic increase in time to solution cf. proprietary or more expensive interconnects, for these types of communication requirements if they are to be considered for general purpose production use in HPC environments. Users must feel assured that they are not sacrificing more performance than is reasonable going on the difference in price between Ethernet and other options. One aim of this study is to ascertain crossover points (i.e. number of processes) between acceptable and unacceptable system performance for typical GbE cluster deployments. In addition, we would argue that even if a cluster is procured for an initial set of applications which are not that communications intensive, this picture may change as new users make use of a facility or existing users develop codes further, and therefore procurement teams should be mindful of the potential pitfalls when using GbE.

Using the above combinations of compilers, GbE switches and MPI libraries, we have benchmarked the widely used IMB (formerly PMB) suite from Intel[19]. This benchmark is easy to build and run and has become the *de facto* standard for testing the quality of MPI

[16] http://xcat.org/doc/
[17] http://www.redbooks.ibm.com/abstracts/redp3866.html
[18] http://www.scl.ameslab.gov/netpipe/
[19] http://www.intel.com/cd/software/products/asmo-na/eng/cluster/clustertoolkit/219848.htm

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. Data for all the MPI-1 functions included in the IMB suite will be available within the group's DBD database (see above).

4. Analysis of Results: Methodology

It is clear that 7 flavours of MPI library/compiler combination, 5 MPI functions, 3 different process counts, 6 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 once more these graphs can be found at http://www.cse.clrc.ac.uk/disco/gbe_perf.shtml .

For the purposes of this paper, we have devised a scheme which 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 which allows us to draw conclusions more readily. The scheme we have devised is as follows:

- 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.)

- However, this approach as it is could in principle award a biased higher scoring to a configuration which exhibits evidence of potentially severe performance problems (due to 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.)

5. Analysis of Results: discussion

MPI performance for 16 processes/8 nodes with a GbE interconnect is expected to be at least adequate, and the performance summarised in Figures 1-5 in general supports this assumption – but with some notable exceptions. `MPI_Alltoall` (Figure 1) in particular is known to be an exacting test of interconnects. The function takes a buffer from every participating process and scatters it to all other processes. If an interconnect performs well on this task, it is probably fair to expect good performance for other IMB tests and 'real-world' applications as well.

Inspection of Figure 1 shows that very few of the configurations tested make the grade. Of particular surprise is the very poor showing of the Force10 S50 switch – no combination of MPI and compiler provides acceptable performance even at 16 processes, and this trend becomes even worse as we go to 32 and then 64 processes. We should mention that better performance for the S50 was observed when running these tests with the 2.4 Linux kernel. However, we suspect that there are other issues that need to be addressed, and that perhaps a firmware upgrade may improve performance when running under 2.6. We will hopefully be able to provide an update on this in future papers.

Examination of Figures 1-5 provides a number of further initial conclusions:

- Most of the other configurations in Figure 1 are shown to be inadequate. LAM-MPI performance is acceptable up to 32 processes (in terms of passing the 'smoothness' test if not always in absolute performance terms) for the Netgear and Cisco switches. Clearly the best configuration is 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 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 (Scali) and free (MPICH and LAM) MPIs.

- Another trend of note across all five sets of graphs is the virtually overlapping performance of the Nortel 5510 and Extreme Summit48si. Casual inspection suggests the numbers are identical – they aren't – and we are confident that our analysis has been carried out without error. This strongly suggests common components produced by third party manufacturers within these switches, and this apparently is the case having consulted with the vendors in question.

- Details of the architectures of the switches under test and how these features affect performance will be examined in detail in a future paper, along with the affect of

adjustments one can make to managed switches cf. the performance of unmanaged switches. But architectural factors to consider would appear to include:

  - The capabilities of the internal switching ASICs and the numbers of ports that each ASIC supports;
  - the latency added going from ASIC to ASIC[20];
  - the forwarding supported in hardware and which layers of the OSI model are supported at this level;
  - how the architecture impacts on IP multicast and broadcast performance (if the MPI library implementation attempts to make use of such features);
  - the size of receive and transmit buffers, the number and type of receive and transmit queues and the thresholds for dropping packets;
  - Etc.

- To reiterate, the Force10 S50 is consistently the poorest performing switch. This is due to its particularly egregious showing for large messages for all tests which cancel out reasonable performance for small messages. We expect there are ways to improve performance (perhaps a firmware upgrade) and will report on this in the future.

- The HP Procurve whilst a better performer in terms of its normalized geometric mean score consistently shows erratic, spiked log-log plots and therefore seems to be a very poor choice for MPI applications. As with the Force10 switch, it may be that there are ways of improving this situation dramatically, but when used as supplied, it seems only right to highlight the severe problems one faces.

- Scali MPI is consistently the best performing library over the series of tests and for the various process counts, both in terms of performance and smoothness of log-log plots leading to some degree of confidence in its capabilities for HPC application workloads. Of course there is a cost associated with this choice and potential purchasers are advised to carry out further benchmarks with 'real-word' applications to try and ascertain likely increases in productivity in a production environment before taking the decision to choose this option.

- Users of free open source MPI libraries – LAM, MPICH etc – should be aware of the role the compiler may play in determining performance. We have built these libraries with recommended compiler switches where available, but our data, summarised in Figures 1-5 and available on the group's web site in the more common format of log-log plot of average time vs. message size, clearly illustrates the possibility of different compilers giving vastly different performance over a wide range of message sizes for the MPI functions we have investigated. One example of this is LAM-MPI with the PathScale compiler for `MPI_Sendrecv`.

- LAM-MPI tends to outperform MPICH for a lot of the tests, however there are notable exceptions where MPICH comes out on top – `MPI_Allreduce` and `MPI_Reduce_scatter` for Extreme, Netgear, Cisco and Nortel switches.

- `MPI_Sendrecv`, a combination, as its name suggests, of a `send` and `recv` combined into a single call to do `send` and `recv` simultaneously thus avoiding deadlocks, is still a point-to-point operation as opposed to a collective (but is often used in implementing collective operations). The performance data for this test

[20] Latency numbers for 64 byte transfers – i.e. the latencies typically quoted in switch manufacturers' literature – are not a reliable indicator of MPI performance. For example, Netgear quote 20 μs, and HP quote 6 μs for the switches we have tested.

confirms the need to push networks beyond the demands placed on them by point to point operations – there is little differentiation between a host of switches and MPI implementations, the exceptions being MPICH built with the PathScale compiler (atrocious performance) and of course the Force10 S50 data. Even LAM flavours with the HP Procurve do quite well on this test.

- Whilst we do not have accurate list prices to hand for the switches under test in this study (and list prices are only a rough guide to what customers may end up paying), it would appear that there isn't a particularly strong correlation between price and performance. We are certain that the Netgear switch is considerably cheaper than the rest (but that its performance is more due to accident than design as Netgear tend to concentrate on addressing the needs of the low-end office networking market).

- Right now we are not in a position to make definitive statements when it comes to choice of MPI library. Whilst Scali does appear to be the leader certainly in performance (and possibly price/performance), we reserve final judgement until completion of our attempts to carry out optimized runs for all libraries under test.

- And of course we need to remind ourselves of caveats already mentioned above. We have not tested anything like an exhaustive list of possibilities when it comes to choice of MPI library or Ethernet switch. But we feel we have accumulated enough data across a range of typically deployed options to start to draw meaningful conclusions which should stand up to further scrutiny and be reflected in 'real-world' applications' performance.

6. Conclusions

It is clear that there are numerous factors which contribute to the performance of MPI functions on GbE interconnects, that these factors interact in complex ways, that performance can be very poor even for low process counts, and that the quality of performance for a given MPI function can vary wildly as a function of message size. Whilst this study has not presented data from attempts to tune performance through, for example, algorithm selection as a function of message size, or adjustment of buffer sizes via environment variables, or tuning of TCP parameters, we do not expect to report in future that such efforts will cure some of the pathological performance problems highlighted in this study.

In considering GbE interconnects for message passing, HPC users would be advised to carry out a thorough requirements analysis, ascertaining the communication patterns of their applications, and through profiling gain some indication of the message sizes for typical workloads if these are known. Thorough benchmarking of these applications in conjunction with full runs of synthetic benchmarks such as IMB over a wide range of message sizes ought to highlight potential problems where adequate performance crosses over to very poor performance and completely congested networks resulting in much increased time to solution and lower productivity. Repetition of such exercises should be an integral part of acceptance testing.

Procurement decisions for GbE clusters based on benchmark exercises with at most one or two applications which involve mostly point-to-point communication, and with one or two data sets apiece, are quite unlikely to spot the regimes in which the cluster configuration under test runs into severe performance problems either when running different data sets, or new more communication-intensive applications altogether.

Our study seems to indicate that there is a strong case to be made for *some* HPC sites to reassess their approach to procurement budgeting and not concentrate almost solely on maximising their compute server count (i.e. HPL numbers), and instead purchase commercial software such as compilers, and in particular commercial optimized MPI libraries, and more robust and thoroughly tested interconnects, even if this means making sacrifices when it comes to node/core count.

Indeed, it may be that users of MPI codes with reasonably intensive communication may be better served by investing in higher performance. We have vacillated somewhat during this project between writing off Ethernet as an HPC interconnect and regarding it as a viable option (with caveats) depending on the configuration under test at the time. In answering the question: 'Gbit Ethernet – an HPC interconnect?', we must at this stage say the jury is still out. We hope to offer more concrete answers in future papers.

7. Acknowledgments

We are grateful to Arif Ali of OCF plc and Egan Ford of IBM for their help in setting up the cluster. We wish to thank the following vendors for loaning switches to us: Cisco Systems (Ray O'Hanlon), Force10 Networks (Richard Machen), Streamline Computing & Nortel Networks (Peter Pearce & Martin Wolfenden). We thank the following individuals for their helpful comments and suggestions: John Taylor (Streamline Computing), Drew Pletcher and Ray O'Hanlon (Cisco Systems).

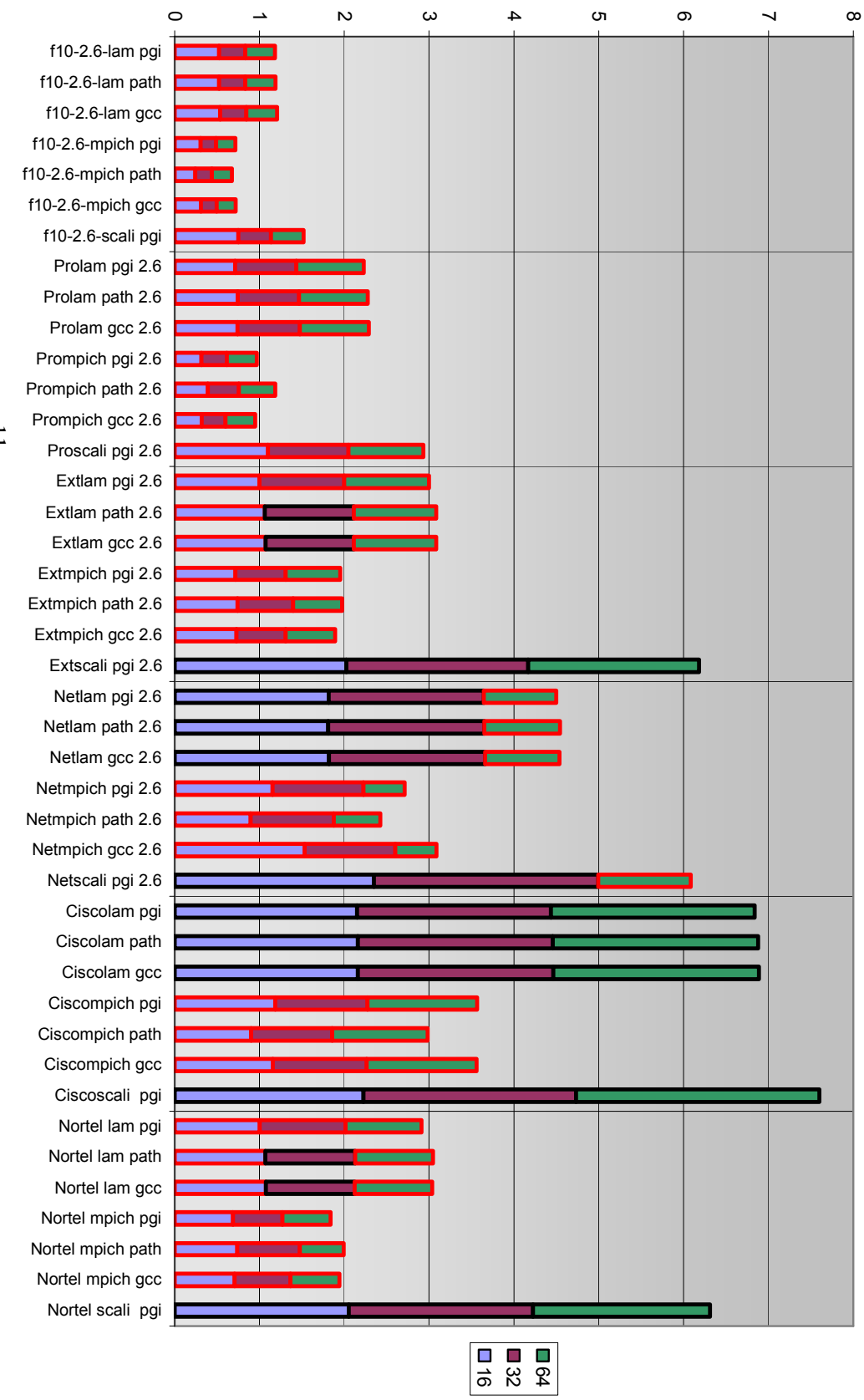


Figure 1: Alltoall geometric mean (Ext/lam/pgi baseline)

Figure 2: All_reduce geometric mean (Ext/lam/pgi baseline)

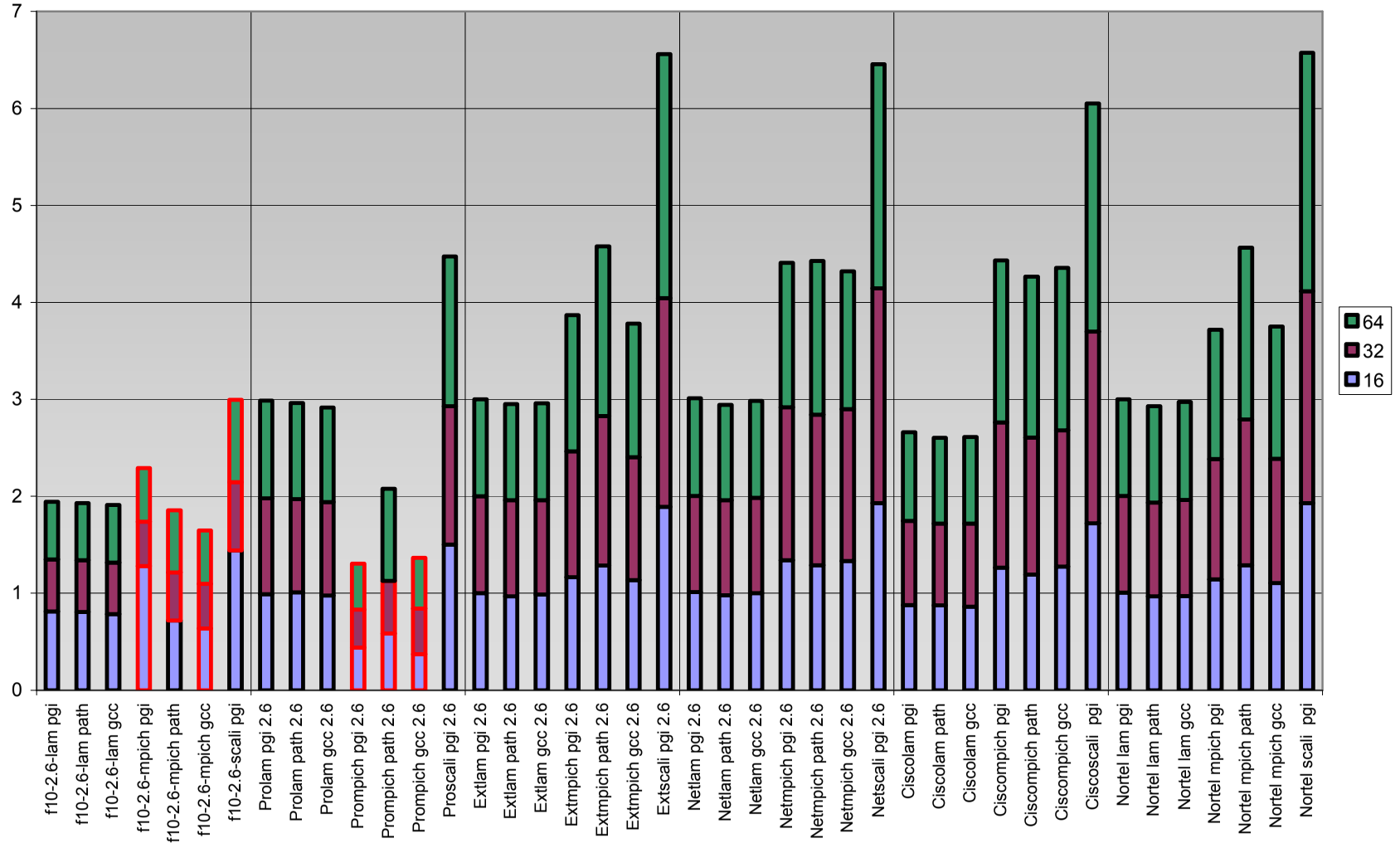


Figure 3: All_gather geometric mean (Ext/lam/pgi baseline)

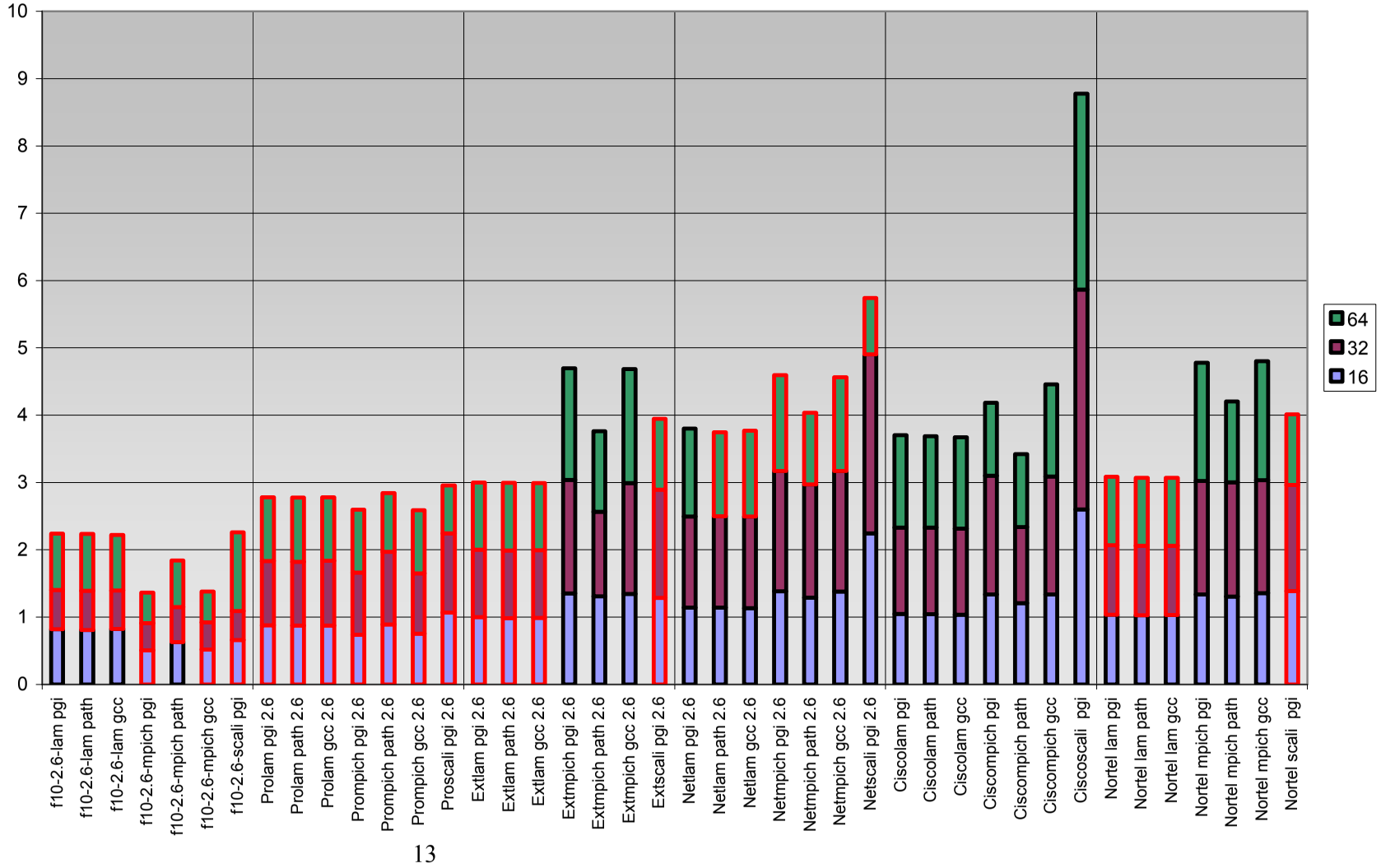
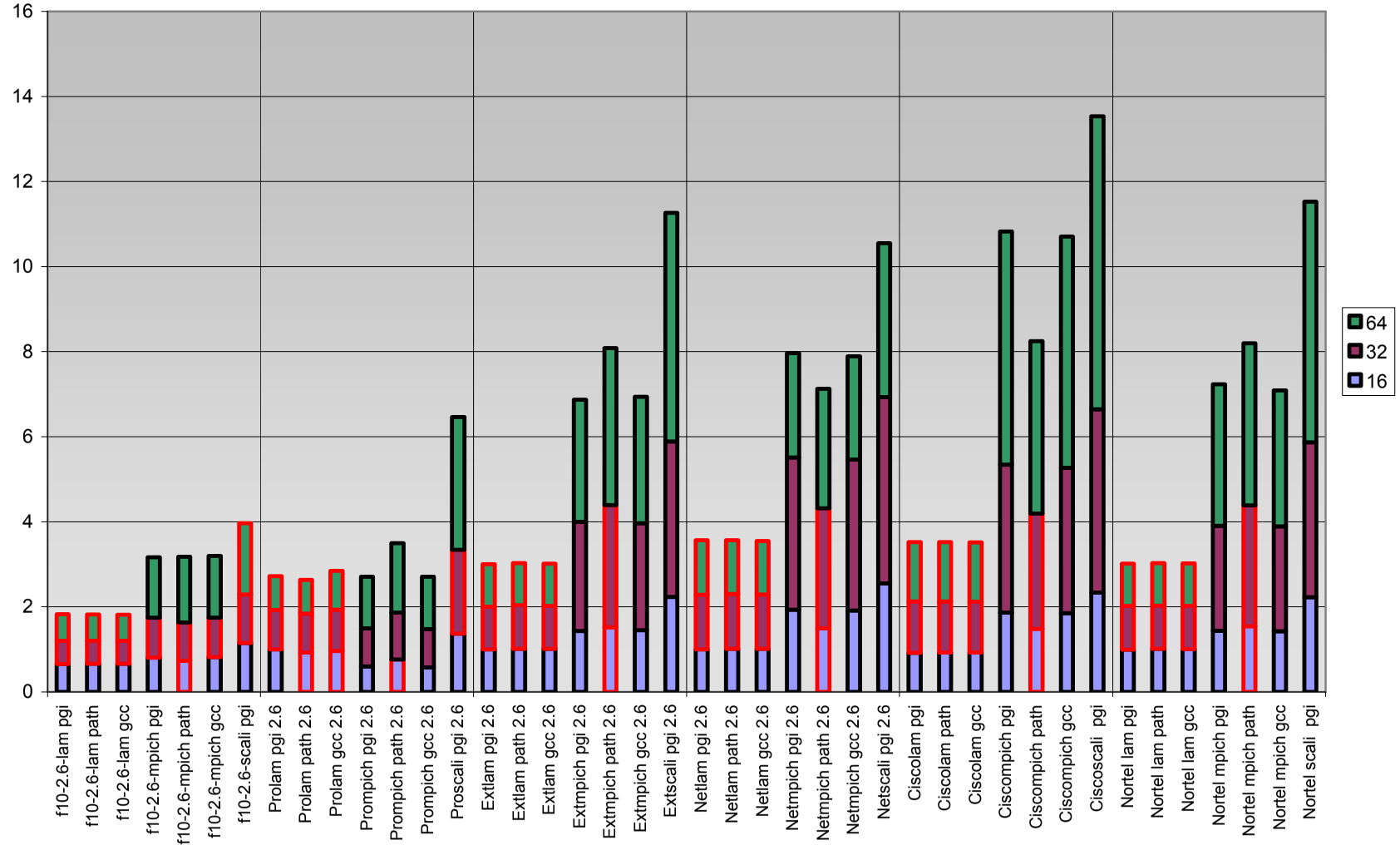


Figure 4: Reduce_scatter geometric mean (Ext/lam/pgi baseline)



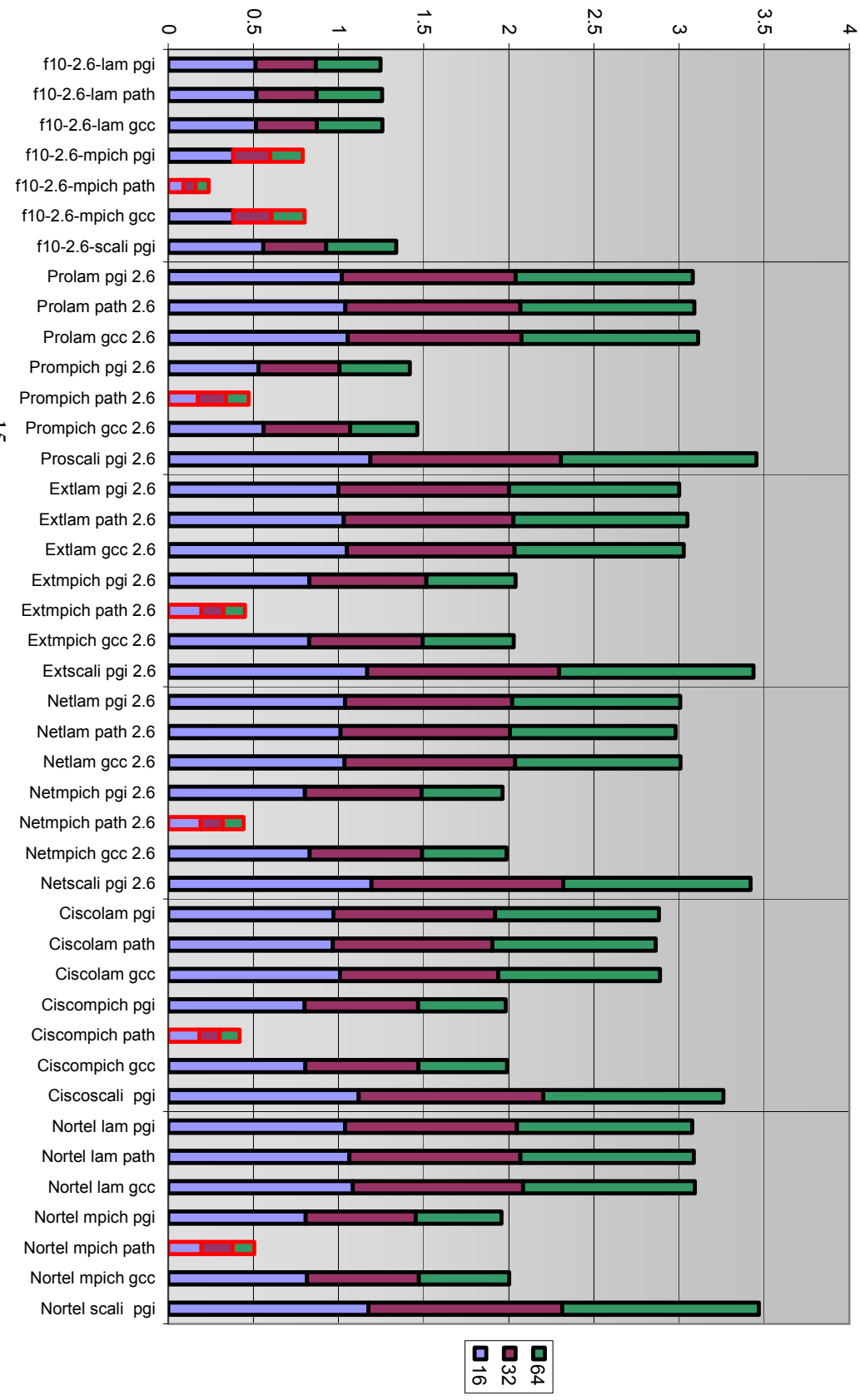


Figure 5: SendRecv geometric mean (Ext/lam/pgi baseline)

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