

This website uses cookies to ensure you get the best experience. More information is available in our [Privacy Policy](#). [Dismiss](#)

IEEE Cluster 2022 ▾

[To-Do List](#)[My Account](#) [Sign Out](#)

IEEE Cluster 2022

Yicheng Li

[My Submissions](#)[Make a New Submission](#)[My Conflicts](#)

All Reviews of pap265 [details](#)

Decision: Reject**Submission:** MARs: Memory Access Rearrangements in Open MPI Datatype Engine**Contributors:** Bosilca, Li, schuchartKey for the below column headings [show](#)

Summary of Reviews of pap265s1

Reviewer	Rel ⓘ	Sound ⓘ	Import ⓘ	Novel ⓘ	Rec ⓘ	Expert ⓘ
Reviewer 1	High (4)	Low (2)	Medium (3)	Low (2)	Weak Reject (2)	Medium (3)
Reviewer 2	Medium (3)	High (4)	Medium (3)	Medium (3)	Weak Accept (3)	Low (2)
Reviewer 3	Very High (5)	Very Low (1)	Low (2)	Low (2)	Reject (1)	High (4)
Reviewer 4	Medium (3)	Medium (3)	Low (2)	Low (2)	Reject (1)	Low (2)
Averages:	3.8	2.5	2.5	2.2	1.8	2.8

Sections Below

[Full Reviews](#) • [jump down](#)[Committee Comments & Notes](#) • [jump down](#)

Full Reviews

Review of pap265s1 by Reviewer 1

[top](#)

► Summary

monospace

The authors goal is to improve the pack/unpack functions in the MPI datatype engine. They do this by using memory access rearrangements (MAR) modifying the datatype description using IOVEC. They conclude that the number of instructions is the main issue for performance. MAR reduces the number of instructions. They focus on the OpenMPI implementation.

► Strengths and weaknesses

monospace

Using IOVEC to minimize the calls to memcpy, which is expensive is a good optimization. In general the work is useful in that it optimized data movement

There is no application motivation for the datatypes tested.

► Detailed Comments

monospace

I find graphs with multiple y-axis hard to interpret. I think most people do.

There is too much data that is very similar and not enough explanation of the results.

It also seems from the data that cache misses are as large an impact as number of instructions since the paper doesn't have a general solution to the pipeline problem

► Suggestions for improvement

monospace

abstract: is what limiting -> is what is limiting the

The only explanation of the figure 9, is: "The calculations showed the number of instructions could affect performance and it is proven from Fig. 9." I think the figure needs more explanation. These figures take up a whole page and there is no real description of why you used the different datatypes or what the results are showing.

The 9 factors on page 5; did you look at Principle Component Analysis or something similar? ie figure out the more impactful factors.

Instruction sets vary with processors, do you think your analysis will hold across processors. You should have tried more than one processor, intel + arm etc.

It would be good to have an analysis of some application datatypes tied to the experiment datatypes.

Scored Review Questions

Rel ⓘ Sound ⓘ Import ⓘ Novel ⓘ Rec ⓘ Expert ⓘ

pap265s1 High (4) Low (2) Medium (3) Low (2) Weak Reject (2) Medium (3)

Review of pap265s1 by Reviewer 2

[top](#)

► Summary

monospace

Authors revamp the datatype description of Open MPI to improve the performance of pack/unpack functions. Significant performance improvement is observed from the proposed scheme on datatype benchmarks.

► Strengths and weaknesses

monospace

Strengths:

(1) The proposed datatype description significantly reduces the number of instructions and the calls of memcpy.

Weaknesses:

(1) No real-world application is used for evaluation, but only datatype benchmarks.

► Detailed Comments

monospace

(1) There are several typos in paper, please go through the paper to rectify all of them. For example, "how master outperform IOVEC in the next section", "Fig. 11 present how MARs".

(2) It's would be more interesting to see the experimental results on real applications, such as stencil computation.

► Suggestions for improvement

monospace

See the comments.

Scored Review Questions

Rel ⓘ Sound ⓘ Import ⓘ Novel ⓘ Rec ⓘ Expert ⓘ

pap265s1 Medium (3) High (4) Medium (3) Medium (3) Weak Accept (3) Low (2)

Review of pap265s1 by Reviewer 3

[top](#)

► Summary

monospace

This paper discusses optimizations to the datatype engine in OpenMPI. It presents an IOVEC based approach and an access rearranging approach (MAR). In some cases, the alternate approaches perform better than the baseline.

► Strengths and weaknesses

monospace

The authors found optimizations that improve the performance of some datatype processing.

It is not clear exactly what the MAR method does. What is presented does not look correct.

► Detailed Comments

monospace

The method of using an IOVEC for a datatype is well known - and well known to scale poorly with many datatype elements. It takes too much storage to describe the datatype.

The way the MAR method is presented makes it look like it packs the data in a different order than the baseline. There is only one correct order for packing data. Please provide additional details on how this method works.

The graphs in the paper label the total payload sizes in an unusual way. For example, 100KB is presented as 10^{-1} MB. Please use more typical graph labels.

► Suggestions for improvement

monospace

Present additional detail on the new scheme proposed.

Present the storage cost for the datatype representation

Use more conventional graph labels.

Scored Review Questions

	Rel ⓘ	Sound ⓘ	Import ⓘ	Novel ⓘ	Rec ⓘ	Expert ⓘ
pap265s1	Very High (5)	Very Low (1)	Low (2)	Low (2)	Reject (1)	High (4)

[top](#)

Review of pap265s1 by Reviewer 4

► Summary

monospace

The authors identify issues with MPI datatype packing as it is implemented in OpenMPI, and explore a method for increasing performance by reducing the amount of work (instructions) required to perform packing and unpacking. If I understand correctly, the method involves first using IOVECs as a basis for packing, and then minimizing the amount of 'bookkeeping' required to perform the packing and unpacking (where the amount of bookkeeping is related to the number of times the current default OpenMPI database engine traverses the unoptimized datatype). The authors compare the performance of their method with the default for a set of datatypes varying according to how much they can be optimized. They show their method can, at least in certain circumstances, increase performance by up to 3.2x. They also analyze instruction counts, showing that their method can (for some of the considered datatypes) reduce instruction counts.

► Strengths and weaknesses

markdown monospace

Strengths:

1. MPI Datatypes have long been a problem for MPI performance, so any progress on that front is welcome.

Weaknesses:

1. The work focuses exclusively on the current (?) OpenMPI datatype engine, and as such is prima facie limited in scope. That is, it is unclear to what degree the issues identified in Section III are general to MPI Datatypes or whether they are unique to OpenMPI.
2. There are significant issues with the clarity of exposition that interfere with the motivation of the work and the reader's understanding of the methods. A more careful and detailed exposition would be beneficial to both motivating the problem (e.g., through better explaining the OpenMPI datatype engine) and explaining the proposed solution.

► Detailed Comments

markdown monospace

As mentioned above, it is not clear that the problem addressed in the paper is unique to OpenMPI's datatype engine. A discussion of how it relates to other MPI implementations would help provide context to better motivate the project.

As mentioned elsewhere, grammar interferes with clarity; e.g., missing indefinite or definite articles, incorrect word choice, possessives. In some cases, the grammatical issues obscure the intended meaning (e.g., "However, because of the high overhead datatype engines come with [15] [11], it has prevented scientific applications from adopting MPI datatype...")

The list of potential factors given in section V.C. suggests the research is "in progress". A more careful exploration of these factors would make the paper much stronger.

► Suggestions for improvement

monospace

--- I. Apparently Frontier (at Oak Ridge in the US) has achieved over an exaflop.

--- I. The introduction seems incomplete: That is, the last paragraph says they analyzed the source of the performance hit, and says what the performance increase was, but does not summarize what the authors did to achieve that performance increase. (At the end of section II vector extensions are mentioned as a way to increase efficiency, but it is still not clear whether that is what the authors intend to do.)

--- III. In subsection A, the paper would benefit from a more detailed explanation of the issue raised by figures 1 and 2, as it is not obvious (especially if the reader is not familiar with the output generated by OpenMPI) how the two examples have the same unoptimized datatype definition yet different optimized definitions.

--- III. The exposition in subsection B is also very difficult to follow.

--- IV. This section contains information that helps clarify section III (esp. III.A). Perhaps the two sections could be combined?

--- V. The way the methodology is stated (1st sentence of section V) makes it seem as if more than 7 different datatypes were considered, but the paper is only going to report the 7 that most benefit from the proposed optimizations. It seems to me the actual motivation is to pick a set of datatypes that can benefit from optimization to varying degrees, and those described in the text meet that requirement.

--- V. The list of potential factors given in section V.C. suggests the research is "in progress". One strategy would be to remove the list and discuss these factors in the discussion section, perhaps targeting them for future work.

--- VI. The explanation of how IOVECs are to be used to pack datatypes is difficult to follow.

- VII. Could you say more about the "Open MPI datatype benchmark"? Is it simply a ping-pong using datatypes?
- VII. There are far too many figures/subfigures given the amount of discussion of them in the main text. Could the points be made using just a single example, with more discussion?
- VII. Since the current work focuses on a particular implementation of MPI (OpenMPI), the paper would benefit from a discussion of whether the same techniques would benefit other implementations.

Scored Review Questions

	Rel ⓘ	Sound ⓘ	Import ⓘ	Novel ⓘ	Rec ⓘ	Expert ⓘ
pap265s1	Medium (3)	Medium (3)	Low (2)	Low (2)	Reject (1)	Low (2)

Committee Comments & Notes

[top](#)

Committee Comments for Authors

None