SC18

Decision: Reject

Submission: Assessing the Performance Impact of an Active Global Address Space: A case for AGAS
Contributors: Amini, Kaiser

Key for the below column headings: show

Summary of Reviews of ws_pawatm110s1: Assessing the Performance Impact of an Active Global Address Space: A case for AGAS

Reviewer	rel ℹ	snd ℹ	imp ℹ	orgn ℹ	pres ℹ	act ℹ	conf ℹ	exp ℹ
Reviewer 1	VERY HIGH (6)	HIGH (5)	MODERATE (4)	MODERATE (4)	MODERATE (4)	WEAK ACCEPT (4)	MODERATE (4)	LOW (3)
Reviewer 2	LOW (3)	MODERATE (4)	MODERATE (4)	MODERATE (4)	HIGH (5)	WEAK REJECT (3)	MODERATE (4)	LOW (3)
Reviewer 3	LOW (3)	MODERATE (4)	MODERATE (4)	MODERATE (4)	LOW (3)	WEAK REJECT (3)	HIGH (5)	HIGH (5)
Averages:	**4.0**	**4.3**	**4.0**	**4.0**	**4.0**	**3.3**	**4.3**	**3.7**

Sections Below

Full Reviews · jump down
Committee Comments & Notes · jump down

Full Reviews

Review of ws_pawatm110s1 by Reviewer 1

top

Comments for Authors:

The manuscripts describes the AGAS capabilites in HPX, and uses the OctoTiger multi-physics AMR HPX application to analyze performace results.

The paper is well organized and easy to follow. Section IV.A could benefit from a more detailed description of the application.

Edits:
* The ordering of the figures is a little confusing.
You discuss figure 4 before discussing figures 2 or 3.
Reversing the figure order will improve readability.
Figure 4 <-> Figure 2

* page 10: or the number of indicated object -> or the number of indicated objects

* page 6. Sec V: In case of AGAS, -> In the case of AGAS,

Scored Review Questions

	rel	snd	imp	orgn	pres	act	conf	exp

ws_pawatm110s1	VERY HIGH (6)	HIGH (5)	MODERATE (4)	MODERATE (4)	MODERATE (4)	WEAK ACCEPT (4)	MODERATE (4)	LOW (3)

Review of ws_pawatm110s1 by Reviewer 2

top

Comments for Authors:

Abstract line 5 "a an" -> "an".

II Related Work, paragraph 2, "Co-Array Fortran" -> "Fortran". (There is no non-coarray Fortran). And I believe the Titanium project is ended.

Scored Review Questions

	rel	snd	imp	orgn	pres	act	conf	exp
ws_pawatm110s1	LOW (3)	MODERATE (4)	MODERATE (4)	MODERATE (4)	HIGH (5)	WEAK REJECT (3)	MODERATE (4)	LOW (3)

Review of ws_pawatm110s1 by Reviewer 3

top

Comments for Authors:

This paper (re)introduces HPX and the AGAS model on which it's based and uses the OctoTiger AMR application for simulating white dwarf binaries to quantify some of the overheads of the AGAS model.

The strength of this paper is that HPX is a very relevant programming model to the PAW-ATM community and has a number of unique features that are worthy of broader study, consideration, and attention. In addition, I believe OctoTiger to be a very interesting and nontrivial application.

The main negative of this paper is that my understanding of OctoTiger comes not from this paper but from previous ones that I've read which makes it feel like a poor match for PAW-ATM's aims. Specifically, rather than being a paper about an application written in an alternative to MPI, it reads more as a paper about an alternative to MPI that happens to be evaluated in the context of an application. OctoTiger is only afforded two short paragraphs of high-level overview, and as such, the reader comes away with very little in the way of insights about OctoTiger itself, what aspects of HPX/AGAS were particularly suitable for it, how easy/hard it was to write, whether HPX would be similarly useful for their application, the degree to which OctoTiger benefited from unique HPX features, etc. By focusing on the evaluation of HPX performance details (yet without providing a comparison to any conventional programming model that the reader might be familiar with) the paper limits the degree to which it will appeal to a broad audience. If you aren't already invested in HPX or OctoTiger, you probably will not learn much from it or gain information to take back to your own work. This all seems regrettable given that I suspect that there could be a great paper that focused on OctoTiger more than on HPX details, but this isn't it—making it a poor match for what I believe PAW-ATM is trying to achieve for attendees and readers. If the paper is accepted, I would strongly recommend that the authors make their at-SC presentation more about OctoTiger and its expression in HPX and leverage of HPX-specific features, and less on HPX itself to appeal to PAW attendees (and if possible, to rewrite the paper in that vein as well). My best guesses as to why this was not done are (a) maybe the authors felt this was already covered in previous publications and didn't want to repeat themselves, (b) maybe the authors had a previously rejected paper that they threw at PAW-ATM with minimum modifications to see if it would stick, (c) maybe the PAW-ATM CFP wasn't written as clearly as it should've been to convey the intended emphasis on applications.

Other downsides to the paper include:

* In addition to its focus on HPX to the exclusion of providing many details about the application or its use of HPX, the paper is somewhat repetitive, as though chunks of text were put together without care as to whether a good and clear story was being told. For example, the reference counting mechanism was described at least three times by my count, each time giving essentially the same information, as though it was the first such description. This was space that could have been used for content that would be of far more interest to the PAW audience (to say nothing of any reader who got it the first time).

* The related work section felt more like an enumeration of other technologies working close to the AGAS space rather than any sort of detailed comparison between those technologies and HPX (which is what I think a good related work section is meant to do). Charm++, probably the most similar technology to HPX in terms of presenting an object-based abstraction with migration got a few mentions, but the reader doesn't come away with a very good understanding of how the systems differ. The section claims that "only HPX has a global address system that allows objects to be relocated at runtime," though that's my understanding of Charm++ as well, leading me to wonder whether my understanding is wrong or whether there's some minor semantic difference that the authors are alluding to without stating very clearly.

* Other related work characterizations seem sloppy: Charm++, UPC++, Chapel, and X10 are all credited with supporting dynamic load balancing, yet I'm reasonably certain that the latter two don't support it at all (in any sort of way other than completely explicit manual copying of objects), and I suspect (but am less confident) that UPC++ is the same. Similarly, all are credited with supporting multidimensional, sparse, associative, and unstructured data structures when I'm fairly certain that few of these models support most of these options, at least out of the box (i.e., such that the programmer need not write them from scratch him-/herself).

* "AGAS maintains mapping tables to be able to map GIDs to local virtual addresses" -- for me, this begged the question of how much space these tables require relative to the number of localities and objects more than the amount of time required to find remote objects.

* Section 3 would be aided greatly by the introduction of illustrations of these concepts (at least, the first 4 or so), or a concrete example. The current presentation requires a lot from the reader to follow along if they are not already familiar with HPX/PGAS/AGAS. The reference to figure 1 gave me hope that it would help clarify these concepts, but turning the page I found instead a very abstract, very standard PGAS memory model diagram that could apply to almost any of the languages in the related work.

* "we exclude data from the bootstrap and teardown phases since they do not directly provide useful data on how AGAS might impact the performance or scaling behavior at runtime." For an applied scientist (target PAW-ATM attendee), I would think that concerns like this would be very much of interest; perhaps less so to a computer scientist interested in the AGAS model abstractly...

* Sec 3B: This is an example of the repetitiveness of this paper. It's arguably telling us about the environmental setup for the experimental evaluation, but instead starts characterizing features of HPX all over again. Then in section 5, HPX's GC is explained yet again.

* The opening paragraph starts with some opening characterizations but then doesn't seem to lead to any sort of summary, conclusion, or point before shifting abruptly to a new topic in the second paragraph.

* Section 3A, sentence 1 was difficult to parse and would benefit from being reworded. E.g., "Due to the novelty of its abstractions, traditional performance tools such as ... can be a poor match for measuring ..."

* Section 2, paragraph 2: Looks like it should start with "In..."

* The paper felt a little short to me, and by my count was 7 pages long where I believe the minimum is intended to be 8. The CFP is arguably slightly contradictory, saying that appendices do count toward the page limit, but then saying the reproducibility section did not.

Scored Review Questions

	rel	snd	imp	orgn	pres	act	conf	exp
ws_pawatm110s1	LOW (3)	MODERATE (4)	MODERATE (4)	MODERATE (4)	LOW (3)	WEAK REJECT (3)	HIGH (5)	HIGH (5)

Committee Comments & Notes

top

Committee Comments for Authors

None