Is Japanese HPC another Galapagos? - Interim Report of MPI International Survey -

Atsushi Hori, George Bosilca, Emmanuel Jeannot, Takahiro Ogura, Yutaka Ishikawa

Is Japanese HPC another Galapagos?
- Interim Report of MPI International Survey -

Atsushi Hori[1] George Bosilca[2] Emmanuel Jeannot[4] Takahiro Ogura[1]
Yutaka Ishikawa[1]

[1]Riken-CCS
[2]University of Tennessee
[3]Inria

July 24, 2019

Abstract

We have been conducting questionnaire survey targeting MPI users of the whole world. At the time of this writing, we get more than 800 answers from more than 40 countries. We analyzed the currently available answers and have found some interesting results which indicate that the Japanese MPI users are different from the MPI users of the rest of the world. This paper focuses on those possible specificities of Japanese MPI users and warns the future of Japanese HPC community based on the resuklt of the survey. Since the survey is still open and accepting answers, this is an interim report of the survey. [1]

1 Overview

In 2017, ECP[2] conducted a survey for MPI users in the ECP project to reveal how MPI would/should be integrated with ECP applications in the future[1]. It ws a coincidence that another survey was conducted in Japan targeting HPCI[5] users which included several questions asking about MPI[6]. HPCI is an infrastructure for HPC users in Japan connecting major supercomputers owned by universities and governmental research institutes. If both questionnaire surveys would have the same questions, then we could compare the answers to reveal the differences between US and Japan. Unfortunately we could find only one similar question related to MPI in both surveys.

We decided to conduct another MPI survey targeting MPI users in the whole world. Since MPI is being widely-used vehicle for high-performance computing, this relatively large-scale questionnaire survey would be beneficial for deciding the future direction of MPI. Additionally, we could find the differences of MPI users among countries and/or regions of the world.

This project is a part of JLESC[3] which is an international research collaboration framework. The international nature of this survey matches the concept of JLESC. Each co-author is a member of JLESC and responsible for the country and/or region where he belongs.

Our MPI International Survey, ECP survey, and HPCI survey are summarized in Table 1. The major differences between our survey and the others are;

Target The ECP survey targeted ECP members who are leading researchers and application developers in USA. The HPCI survey targeted HPCI users who are programming HPC applications and using

[1]The word "Galapagos" in the title is widely used in Japan to indicate special technologies developed only in Japan and endangered by the other countries. "Galapagosnized" which is a passive voice of verbed "Galapagos" is often used to express such endangered technology. The analogy between Galapagos and Japan is that both consist of islands and species (technologies) evolves independently.

Table 1: Survey Comparison

Survey	Target	# Questions	# Answers
ECP	US	64 (max.)	77
HPCI	Japan	75 (max.)	105
Our survey	International	30	800+

Japanese supercomputers. Thus, both surveys targeted high-end users. Whereas our survey targets all MPI users from novices to experts.

Number of answers The number of answers of ECP survey and HPCI survey are 77 and 105, respectively. Because of the wider target of our survey, in terms of the scope of MPI expertise of participants on a global scale, the expected number of answers would be larger than those of preceding surveys.

Number of questions The number of questions in our survey is 30 which is much smaller than those of ECP and HPCI surveys.

As a result of our survey, although it is still accepting answers, we got 800+ answers from 40+ countries at the time of this writing. This number of answers allows us to conduct cross-tab analysis between questions on each countries and/or regions.

At the time of this writing, the survey is still open and accepting answers to have more participants from the world. This is an interim report of our survey.

2 Questionnaire

2.1 Design

The points we kept in our mind while we were designing the questions are;

Minimizing the number of questions The number of questions must be less than around 30 to keep participants concentrated[2]. The maximum numbers of questions in the ECP survey and HPCI survey are 64 and 75, respectively. Thus, the number of questions in this survey is far less than those. We focused on MPI itself to reduce the number of questions. For example, the questions asking about the tools are intentionally excluded simple because this may add several tens of questions. Another survey must be conducted for the topics not included in our survey.

Easy-to-answer We designed our questions to minimize the stress of participants as much as possible. None of our questions requires free descriptive answers. Additional investigations and/or efforts are minimized. If we would ask participants about the LOC (lines of codes) they have ever programmed, they would run the `wc` command on their programs.

Avoiding ambiguity The above LOC question may introduce ambiguity into its answer; 1) large numerical applications may have more than hundreds thousand LOC, but the LOC for MPI function calls is much less than the whole LOC, 2) participants might or might not include the LOC of supplementary code such as `Makefile`, `configure` and others. We also tried not to have MPI-IO related questions because of not only the number of questions but also the difficulty for novice MPI users to identify the root cause; the MPI standard, MPI implementation, MPI-IO implementation, system configuration, or underlying file system.

[2]This number was advised by Prof. Marshall Scott Poole at Illinois Univ., and Prof. Iftekhar Ahmed at Univ. of North Texas, they are also the members of JLESC[3], from their social science viewpoints.

Table 2: Questions

Q1: What is your main occupation?

Country: Select main country or region of your workplace in past 5 years

Q2: Rate your overall programming skill (non-MPI programs)

Q3: Rate your MPI programming skill

Q4*: What programming language(s) do you use most often?

Q5: How long have you been writing computer programs (incl. non-MPI programs)?

Q6: How long have you been writing MPI programs?

Q7*: Which fields are you mostly working in?

Q8*: What is your major role at your place of work?

Q9: Have you ever read the MPI standard specification document?

Q10*: How did you learn MPI?

Q11*: Which MPI book(s) have you read?

Q12*: Which MPI implementations do you use?

Q13: Why did you choose the MPI implementation(s)?

Q14*: How do you check MPI specifications when you are writing MPI programs?

Q15: What is the most difficult part of writing an MPI program?

Q16*: Which MPI features have you never heard of?

Q17*: What aspects of the MPI standard do you use in your program in its current form?

Q18*: Which MPI thread support are you using?

Q19*: What are your obstacles to mastering MPI?

Q20: When you call an MPI routine, how often do you check the error code of the MPI routine (excepting MPI-IO)?

Q21: In most of your programs, do you pack MPI function calls into their own file or files to have your own abstraction layer for communication?

Q22*: Have you ever written MPI+"X" programs?

Q23: Is there any room for performance tuning in your MPI programs?

Q24*: What, if any, alternatives are you investigating to indirectly call MPI or another communication layer by using another parallel language/library?

Q25: If there were one communication aspect which is not enough in the current MPI could improve the performance of your application, what would you prioritize? Or is MPI providing all the communication semantics required by your application? If not, what is missing?

Q26*: Is MPI providing all the communication semantics required by your application? If not, what is missing?

Q27*: What MPI feature(s) are NOT useful for your application?

Q28: Do you think the MPI standard should maintain backward compatibility?

Q29: In the tradeoff between code portability and performance, which is more or less important for you to write MPI programs?

2.2 Questions

Table 2 shows all questions in our survey. The number of questions is 30 including a question asking the country of participants. Note that this question does not ask the nationalities of participants but workplace for recent 5 years. The answers of this question are used to categorize answers into the countries and/or regions in the following analysis sections. The question suffixed by an asterisk (*) allow participants to select multiple answers.

We conducted prerelease testing on MPI Forum[4] attendees and Riken-CCS researchers. An interview with a R-CCS researcher was also conducted to debug and tune the questions at the very final stage of the questionnaire design.

3 Distribution

The questionnaire is implemented by using Google Forms and distributed by sending e-mails to major mailing lists such as `hpc-announce@mcs.anl.gov`. We started distributing the survey from 17th of February, 2019. All data in this paper is as of 10th of May. Figure 1 shows the number of answers since then.

Soon after we started distribution, we realized that the major mailing-lists did not work well as we expected. So we started asking our friends to help us to distribute the survey to their local communities. In Figure 1, the number of answers increases stepwise. This is because our friends re-distributed the survey to their local communities at each step. As shown in this figure, this hierarchical distribution to reach local communities worked well.

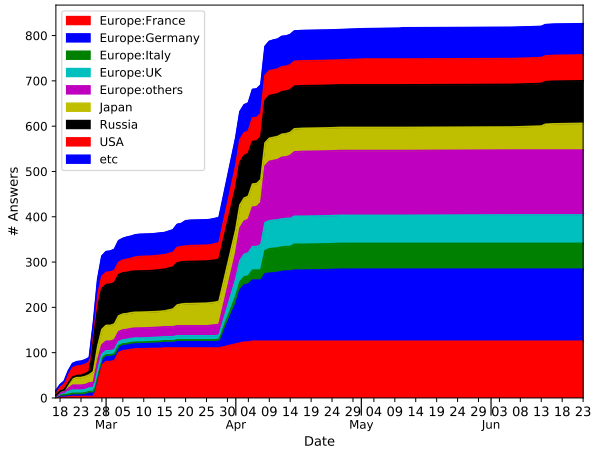

Figure 1: Time series

The answers are automatically collected by Google Forms. So a questionnaire survey in this way (using Google Forms and sending e-mails) is very easy, fast, and no charge. However, we could have answers only from the mailing-list members and these responses may have the possibility of having some bias.

We developed a Python program to analyze the resulting CSV files. For cross-tab analysis, this program outputs graphs of all possible combination of two questions excepting the ones where both questions allow to have multiple answers. All graphs in this paper were created by this Python program.

4 Profile of Participants

Table 3 shows the number of answers of top-10 countries. Again, it should be noted that the question asked the workplaces of participants for recent 5 years, not the nationality. As of this writing, we got only 4 answers from China. This is because Chinese government does not allow its people to access Google.

Table 3: Top 10 Countries

Rank	Country	# Answers
1	Germany	159
2	France	125
3	Russia	94
4	UK	62
5	USA	57
6	Italy	57
7	Japan	51
8	Switzerland	40
9	Korea, South	27
10	Austria	26

41 countries, 817 answers

Table 4 shows the top-10 countries of the system share in Top500 list as of November 2018[7]. Comparing Table 3 and 4, it is ver obvious that the numbers of answers from China, USA, Japan and Canada in our survey do not reflect the system share in Top500. This is the reason not to close the survey until the gap between Table 3 and 4 becomes close enough.

Table 4: Top 10 Countries in Top500 System Share[7]

Rank	Country	System Share [%]
1	China	45.4
2	USA	21.8
3	Japan	6.2
4	UK	4.0
5	France	3.6
6	Germany	3.4
7	Ireland	2.4
8	Canada	1.8
9	Italy	1.2
10	Korea, South	1.2

As of November 2018

From this time onward, the countries and regions (a set of countries) having more than 50 answers are focused in this paper (the top-7 countries in Table 3). The countries and regions having less number of answers may contain large bias and they are inadequate for cross-tab analysis.

Figure 2 shows the survey result of Q1 in Table 2 asking the participants' occupations. Although we can see some diversities, most answers, around 80%, come from research organizations (universities and governmental research institutes). We do not think this diversities did not reflect the characteristics of the countries, but came from the biased questionnaire distribution.

This profile may bias the analysis in the following sections. Thus, readers must keep this in their mind on the above situations. The following sections will show the results of analysis of our survey data. In this paper, we chose the ones which reveals the specificities of Japan.

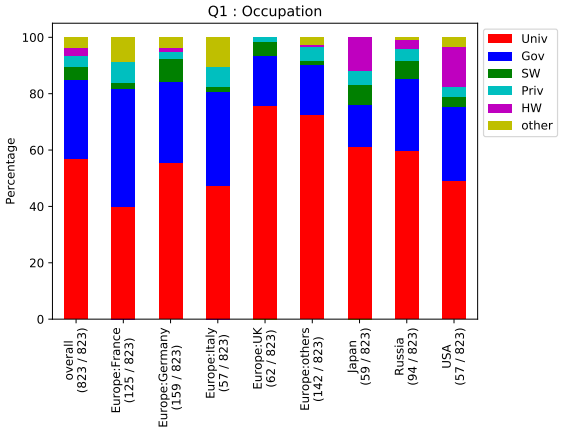

Figure 2: Q1: Occupation. (**Univ**: University, **Gov**: Governmental institute, **Priv**: Private institute, **SW**: SW vendor, **HW**: HW vendor)

5 Simple Tabulation

Figure 3 shows the simple-tab results asking how many years for writing programs (including MPI) and Figure 4 shows the results asking how many years for writing MPI programs. Each bar represents a country or region having more than 50 answers. "whole" represents the whole data and "Europe:other" represents the sum of other European countries. The numbers following the column titles represent the number of answers of that country/region and the number of total answers of the question.

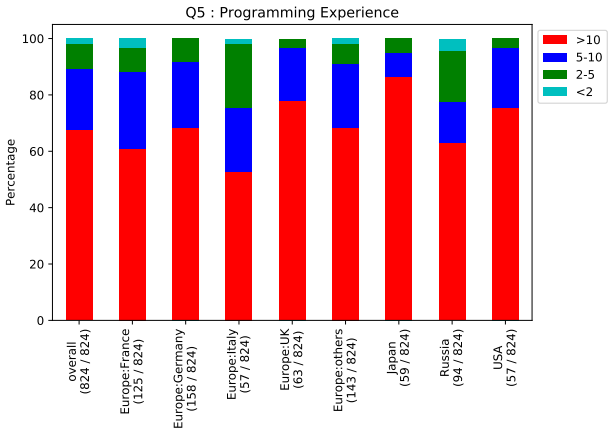

Figure 3: Q5: Programming Experience

It is very interesting that the Japans' percentages of writing MPI and non-MPI programs more than 10 years are highest among the others. This looks like the Japanese HPC researchers and programmers are well-experienced. However, it can also be said that only little young researchers and programmers are writing MPI programs. Contrastingly in Germany and Russia cases, novice users, intermediate users, and experienced users are almost equally distributed. These look more ideal than the Japanese case.

Figure 6 shows the answers asking about the MPI difficulty. In Japan, the ratios of people having the

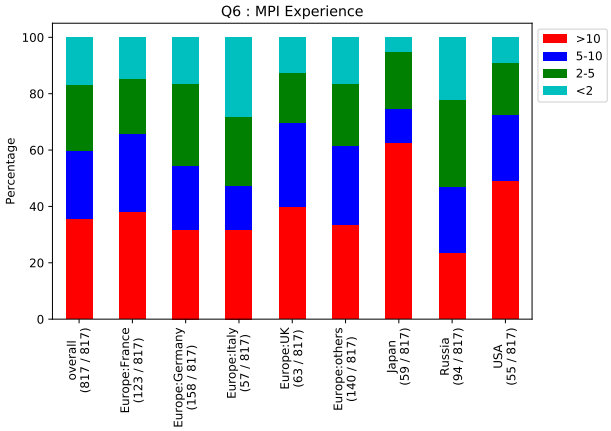

Figure 4: Q6: MPI Experience

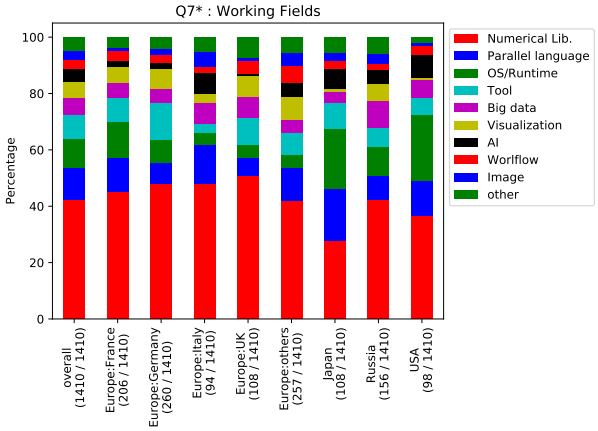

Figure 5: Q7: Field

difficulty for debugging and tuning are the highest among the other countries. And the ratios for algorithm selection and domain decomposition, which are apparently higher levels than the levels of debugging and tuning, are least among the other countries/regions.

These results may come from the role of participants. Figure 7 shows the role of participants. Unlike the other countries, more Japanese MPI users are working on research and development of OS and runtime, and less users are working on tools. This Japan's specificity may affect the result of Figure 3 and 5, because the OS and runtime code are harder to debug and tune than that of applications in many cases. It is also assumed that algorithm selection and domain decomposition are the roles of its users.

Figure 8 shows the result of asking the room for tuning in participants' programs. As shown in this figure, Japan has the lowest ratio of the answer "My programs are (already) well-tuned." At the same time, Japan has the highest ratio of the answer "Rewriting programs is too hard," while they recognize the room for tuning in their applications. Yes, we agree that rewriting a program for tuning sometimes requires lots of work; for example, major data structure changes may affect whole program.

In this question, there is another choice of "I do not have enough resource for tuning." What is the difference between the answers of "rewriting is too hard" and "not having enough resource?" Considering

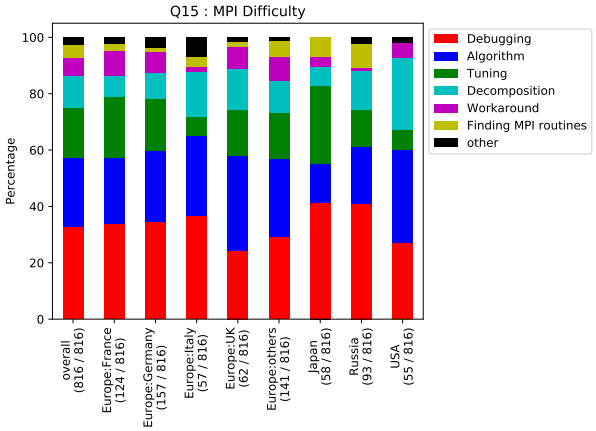

Figure 6: Q15: Difficulty

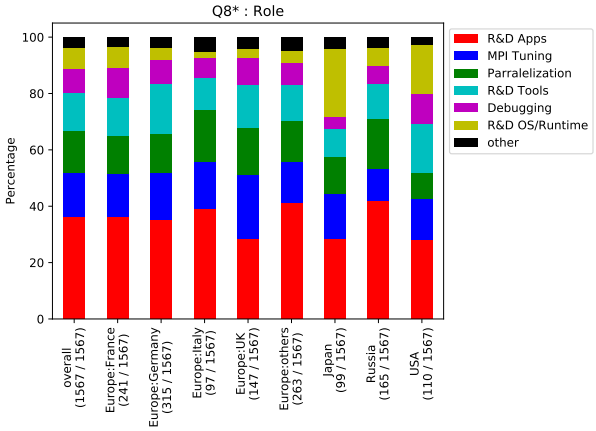

Figure 7: Q8: Role

with the lowest ratio of "My programs are well-tuned" answer, "rewriting..." answer sounds like a trouble or *giving up thinking* while the latter one sounds more aggressive.

6 Cross Tabulation

The cross-tab graphs in this section are the heatmap graphs. The higher (darker) the value (color) of each cell, the higher the frequency of the cell. There are nine graphs in this figure, each graph represents a country or region. All garphs have the same scale. The lower-right graph is the legend of these graphs serving as a color bar, too. The numbers in the cells in the legend graph are percentages. The rows and columns consisting only of the cells less than 4% in all countries/regions are omitted to increase readability.

Figure 9 shows cross-tab graphs between Q3 (asking MPI skill) and Q18 (asking MPI experience). As shown in this figure, Japanese answers are concentrated in the cell with the high MPI skill of 5 (out of 6, larger the number, higher the skill) and having more than 10 years of MPI experiences. Although the Japan's Mhigh peak makes the peaks of the others lower and hard to see, the peaks of the other countries

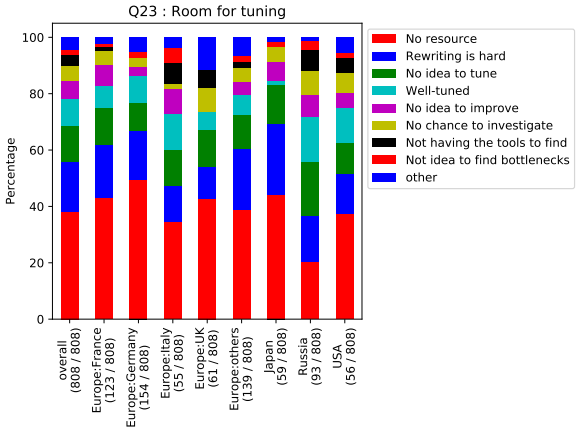

Figure 8: Q23: Room for Tuning

are rather distributed.

Figure 10 shows the cross-tab graphs between Q3 (asking MPI skill, again) and Q18 (asking MPI thread level). There is a peak at 'Never used,' this means that many Japanese participants do not explicitly call the `MPI_Init_thread` function. The doubt here is why Japanese MPI *experts* do not call the `MPI_Init_thread` function. In contrast, in the Russian case, the ratio of "Never used" answer increases when the MPI skill decreases. In the USA case, the ratio of using `MPI_THREAD_MULTIPLE` increases when the MPI skill goes up. Those situations are quite reasonable. Thus, the Japanese specificity comes to the front.

7 Concerns about Japanese HPC

In this section, we dare say our concerns about Japanese HPC. The following four points are our hypothesises based on the results of our survey. Our survey might be biased and the hypothesises might be wrong. But these could be our warning messages.

7.1 Aged MPI users

As shown in Figure 3 and 4, the most MPI users in Japan have more experiences than the other countries. This might sound good, but it is not good when you think about its future. This means there are only little *novice* MPI users in Japan. Experienced (old) Japanese MPI users decreases in the future. We call MPI users in Japan *aged*, not experienced. The reason of this will be described in following subsection.

7.2 Less effort on MPI programming

In the previous section we discussed about another Japanese specificity about the use of `MPI_Init_thread` in Figure 10. Considering with the aged MPI users in Japan, one possible explanation of this is that they are still calling MPI functions only appeared in the old MPI standard. And this hypothesis can lead to the next hypothesis; they do not writing MPI programs from the scratch, but changing existing old MPI programs.

This new hypothesis is consistent with the result of Q15 asking MPI programming difficulty. As discussed in Section 5, many Japanese participants have the difficulty with the low-level, early-stage program development; debugging and tuning, whereas they do not dominate that much in the other countries or regions. If an MPI user tries to write a MPI program from scratch, firstly they have to think about parallel algorithm and domain decomposition. Many MPI users in Japan do not suffer from these issues.

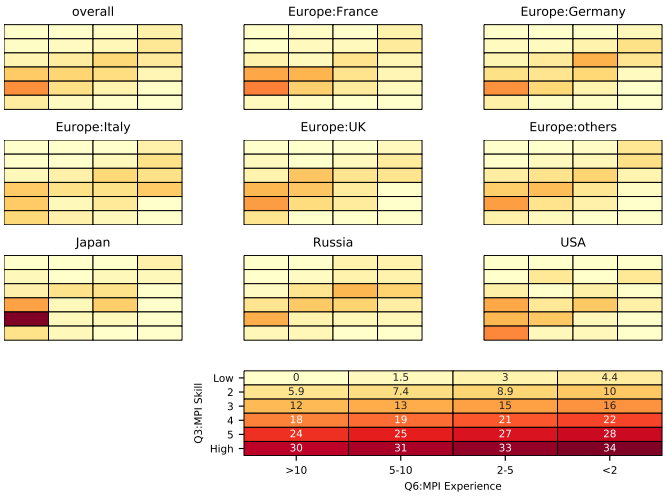

Figure 9: Q3-Q6: MPI Skill and MPI Experience

7.3 Negative attitude to MPI programming

On the question Q23 asking room for performance tuning, there is one Japanese *other* descriptive answer; "Yes, but its an MPI implementation issue." This sounds like that the tuning effort is not his/her job but MPI implementors. This answer is too negative and he/she does not capture the nature of parallel programming at all.

7.4 Are Japanese MPI users really expert?

Taking into account of all above discussions, we dare say "MPI users in Japan may NOT be experts, although they say so." They declare they have long experiences with MPI in the Q6 question, but the other questions in our survey reveal that they may not be well-experienced. As the aged MPI users fade away in the future, it is hard to deny that the Japanese HPC will be in a big trouble.

8 Summary and Future Work

We report an interim report of MPI International Survey. We designed the questionnaire very carefully; 1) 30 questions, 2) easy-to-answer, and 3) minimizing ambiguity. As of this writing we got more than 800 answers from more than 40 countries. In this paper, we focused on the Japanese HPC situation though the questionnaire. Although the answers might be biased, we warned the possibilities of critical situations in Japanese HPC. These are three points; a) aged MPI researchers and developers, b) less effort on MPI programming, and c) negative attitude to MPI programming.

Recently we opened a new survey site for those who cannot access Google Forms by using Microsoft Forms, having the same questions and answer choices. We are trying to get more answers from China which owns the most powerful supercomputers. We are also trying to narrow the gap described in Section 4. When we get enough number of participants, we will publish the final report.

We believe that this international survey in the JLESC framework has worked very well. The other HPC related surveys which we excluded due to the number of questions are planned to follow.

All survey data obtained by Google Forms and Microsoft Forms and the Python program to draw graphs are open and can be downloaded from GITHUB freely (`https://github.com/bosilca/MPIsurvey/`).

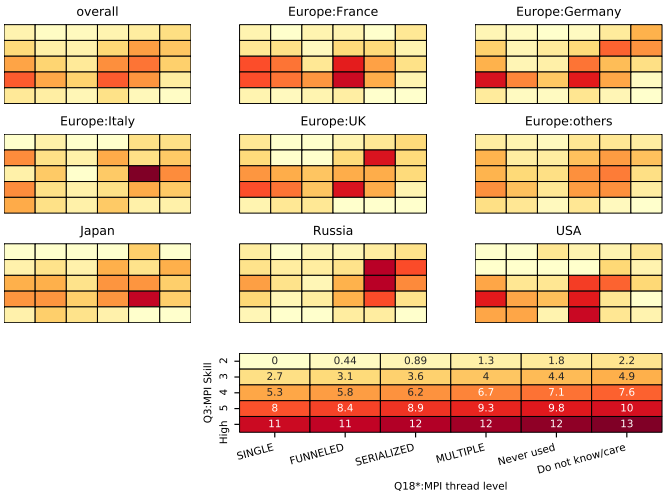

Figure 10: Q3-Q18: MPI Experience and MPI Thread Level

The URLs for the questionnaire are shown in Figure 11. Again, the survey is open and accepting answers. We welcome new participants. If you are interested in this survey, access one of the ULRs in this figure (not both).

Google Forms:
https://docs.google.com/forms/d/
1TQmGoO6xTC1eCNesYRH0z1yTL75Q
D-hSywGxUM4AGIo/edit

Microsoft Forms:
https://forms.office.com/Pages/
ResponsePage.aspx?
id=DQSIkWdsW0yxEjajBLZtrQAAAA
AAAAAAANAAf_cH7pUNE5ONzI0
S1paSEs2WkYyVU9TTFNHOVJSVy
4u

Figure 11: QR codes to access the survey

9 acknowledgment

We thank to those who participated in this survey and those who helped us to distribute the survey to their local communities.
This research is partially supported by the NCSA-Inria-ANL-BSC-JSC-Riken-UTK Joint-Laboratory for Extreme Scale Computing (JLESC, `https://jlesc.github.io/`).

References

[1] David E. Bernholdt, Swen Boehm, George Bosilca, Manjunath Gorentla Venkata, Ryan E. Grant, Thomas J. Naughton, III, Howard P. Pritchard, Martin Schulz, and Geoffroy R. Vallee. A survey of mpi usage in the u.s. exascale computing project. 6 2018.

[2] Exascale Computing Project. Exascale computing project. `https://exascaleproject.org/`.

[3] JLESC. Joint laboratories for extreme-scale computing. `https://jlesc.github.io/`.

[4] MPI Forum. Mpi forum. urlhttps://www.mpi-forum.org.

[5] Research Organization for Information Science and Technology (RIST). High-performance computing infrastructure. `http://www.hpci-office.jp/folders/english`.

[6] Research Organization for Information Science and Technology (RIST). 4hpci . `http://www.hpci-office.jp/materials/k_chosa_4th.pdf?4th`, 2018.

[7] TOP500.org. Top 500. `https://www.top500.org`.