Hacking at the divide between polar science and HPC: Using hackathons as training tools

Jane Wyngaard*, Heather Lynch†, Jaroslaw Nabrzyski*, Allen Pope‡ and Shantenu Jha§
*Center for Research Computing, University of Notre Dame, South Bend, IN
Email: jwyngaar@nd.edu, naber@nd.edu
†Ecology & Evolution, Institute for Advanced Computational Sciences
Stony Brook University, Stony Brook, NY, Email: heather.lynch@stonybrook.edu
‡National Snow & Ice Data Center, CIRES, University of Colorado Boulder, Boulder, CO
Email:allen.pope@nsidc.org
§Department of Computer Engineering, Rutgers University
New Brunswick, NJ, Email:shantenu.jha@rutgers.edu

***Abstract*—Given the current scientific questions of societal significance, such as those related to climate change, there is an urgent need to equip the scientific community with the means to effectively use high-performance and distributed computing (HPDC), Big Data, and tools necessary for reproducible science. The *Polar Computing RCN* project (http://polar-computing.org) is a National Science Foundation funded Research Coordination Network, which has been tasked with bridging the current gap between the polar science and HPDC communities. In this paper we discuss the effectiveness of "hackathons" as a model for implementing both the pedagogical training and the hands-on experience required for HPDC fluency. We find hackathons effective in: (i) Conveying to a science user how and why HPDC resources might be of value to their work, (ii) Providing a venue for cross discipline vocabulary exchange between domain science and HPDC experts, (iii) Equipping science users with customized training that focuses on the practical use of HPDC for their applications, (iv) Providing hands-on training with a realistic domain-specific application in a community of one's peers, but are (v) an incomplete training model that requires supplementation via domain science specific HPDC training materials. In addition to their pedagogical benefits, hackathons provide additional benefits in terms of team building, networking, and the creation of immediately usable products that can speed workflows both for those involved in the hackathon as well as others not involved in the hackathon itself.**

I. Introduction

Climate change in the 20th and 21st century is one of the most pressing issues of our day, and the inherently global scale of the problem creates demand for HPDC resources. Nowhere is climate change more dramatic than in the polar regions [1]. Recent studies indicate accelerated thawing of permafrost, receding Arctic sea ice, and accelerating mass loss from ice sheets (Greenland and Antarctic) and mountain glaciers. Understanding the changing polar regions and connections to global climate involves working with multiple heterogeneous sets of data, from multiple distinct domains of expertise. Oceanography, Climatology, Glaciology, Meteorology, and Biology (along with the many subfields within each) all generate and use their own data sets, which may include field measurements, high-resolution observations from satellites, airborne imagery, and computer model outputs. Computational approaches now support faster and more fine-grained integration and analysis of these and other data types, and provide a better understanding of the complex processes that are rapidly changing our climate.

However, despite these data- and compute-intensive scientific needs, polar science is poorly represented in the use of HPDC resources. Through informal community engagement, a Polar-HPDC workshop in 2014 [2], and the 2016 hackathon reported on here, we have identified two primary reasons for this gap: (1) a lack of community awareness of available compute resources, and (2) a lack of appropriate training for scientists interested in using HPDC for science applications.

There is therefore an urgent need to equip the scientific community to use high-performance and distributed computing (HPDC) and Big Data technologies[1]. The scientific questions facing society increasingly require not only greater computing power but also tools for sharing code and data that minimize duplication of effort and allow for reproducible science. Fortunately, for many applications, the technology for such already exists. Unfortunately, there is a gap in skill set and tool usability that needs to be overcome on a broad scale in order for these methods and tools to be used efficiently and effectively by science domain experts. While rapid development in the field of HPDC has eliminated many of the technical barriers to solving modern scientific problems, the pace of change has made it more difficult than ever for domain scientists to keep pace. Accordingly, we need a renewed focus on training domain scientists to use HPDC resources, methods, and tools effectively.

The above challenges and the resulting gap have now been formally recognized by funding agencies and researchers, not just in the polar sciences. This has led to experiments with new pedagogical approaches. We propose here an alternative-HPDC education approach tailored to equipping domain expert scientists who are traditionally not users of HPDC. We refer to

[1]We use the overused term 'Big Data' and Big Data technologies to represent scalable methods and technologies to address any of the 'V's' – volume, velocity, variety and veracity – noting that 'big' is relative to historic norms and varies tremendously across domains.

this as ALT-HPDC education. ALT-HPDC training is focused on communicating what's Appropriate & Available (A), communicating clearly the Lingo (L) of HPDC, and is Tailored (T) to only cover those aspects of HPDC that are essential for a domain's users.

Taking a lesson from the long running and widely successful introduction to scientific computing offered by Software Carpentry (SWC) [3], we believe there should be training materials focused on only those components of HPDC that are absolutely "need to know" for our audience. This approach recognizes that domain scientists using HPDC may lack the time or "mental bandwidth" to learn any details irrelevant to running their compute jobs. The three stages of our ALT-HPDC training program focus on creating a scientific community that is HPDC-aware, HPDC-literate, and HPDC-trained, respectively, at the expense of more in depth training in hardware and algorithms typical of current HPDC education. Researchers who are able to use HPDC resources successfully even at a basic level will feel empowered and motivated to learn more, and this can most easily be done if the initial time requirement for getting started is minimized. If, at a later stage, greater in-depth knowledge is required for code optimization, HPDC-engaged researchers will be equipped to seek out and gain from further training, such as is already widely available through the existing documentation for HPDC resources.

We need to target both graduate students and established researchers given that HPDC resources and the reasons for their need are a recent development in many domains. Effective mid-career training is particularly difficult to achieve as it is difficult for PIs to carve out enough time for learning new skills. To address this challenge, lessons need to be broken up into deliberately small chunks and formatted such that they can be completed in the margins of other commitments. By focusing only on those skills that are "need to know", and by re-packaging that information in small chunks, we believe it possible to overcome some of these hurdles. In the polar community specifically, it is often easiest to carve out time for such training when researchers are in the field, since polar deployments can involve long periods of uninterrupted "downtime". This means, however, that these researchers are also often working with limited to zero connectivity, a factor which must therefore be considered when designing training materials. Further, even for non-field researchers or those without significant downtime, breaking training into smaller, independent chunks will facilitate uptake across the community.

Recognizing the importance of this knowledge gap, and particularly in the polar sciences, the National Science Foundation has funded a number of initiatives tasked with addressing this issue. Amongst them is the Research Coordination Network (RCN) that the authors lead. The remainder of this paper follows with: Section II describes the reasons behind the described gap's persistence, Section III reviews some of the novel approaches being undertaken to address these such as short intense courses and hackathons, Section IV describes our use of a hackathon-based model, and finally Section V presents our conclusions.

II. Challenges perpetuating the divide

As discussed, through surveys and community workshops we have identified two primary reasons for the relatively low uptake of HPDC by the polar community.

1) *Polar science professionals are (in general) poorly equipped to use HPDC resources: We attribute this to five causes.*
    a) **Rate of technological change:** Priority in training scientists is unavoidably given to the domain science itself. Yet, the rate of technological change means easy-to-use tools of abstraction and formal undergraduate and graduate HPDC training generally lags far behind.
    b) **Momentum:** In our experience senior researchers are sometimes the slowest to adopt new technologies in their own workflows, which makes their adoption into formal coursework slow. They are also unable to train graduate students in these skills and may discourage students from using workflows that "break with tradition" in terms of curent lab methodologies.
    c) **Limited Time:** The time available for established researchers to become HPDC proficient is extremely limited, an issue that is exacerbated in the polar community for researchers that spend considerable amounts of time in remote field camps. For this subset of the community, training and analyses have to be completed in the relatively small window between the conclusion of one field season and the start of planning for the next.
    d) **Inappropriate material:** Most HPDC resources have detailed documentation, often accompanied by optional training manuals and courses (both in face to face formats and offered as remote or recorded materials). However, these materials generally assume a relatively high level of programming, computer sophistication, and understanding of terminology that is often inappropriate for a general science audience.

    The materials' inaccessibility is in part due to a missing common vocabulary. In order for engineers and computer scientists to work with science users on applying HPDC resources to science challenges, a degree of common vocabulary is necessary. While HPDC terminology can be taught to all users, interpreting each science domain's terminology into common language requires partnership and direct engagement. Even a small number of HPDC-fluent scientists in each domain can have an enormously positive impact on their local communities, because they can explain terms and procedures in a way that is tangible to the community, and can address language or software specific questions.

Training materials also often prematurely focus on the details of the hardware, and fail to communicate why these details are of relevance. Emphasis should instead be given to running any job successfully, with details on computing efficiency coming only after a user is comfortable with the system and motivated to delve deeper.

Finally, due to the focus on hardware and optimization, the materials are forced to gloss over the practical elements of job submission (ignoring the possible hurdle of even using a terminal interface first), to focus on theoretical elements aimed at current users interested in optimizing their code or gaining greater efficiency.

e) **Limited HPDC support staff on campuses:** As noted above, advanced HPDC ecosystems are continuously improving but therefore also changing. Support staff for advanced HPDC has proven to be a critical consideration for campuses which is often lacking both in presence, and when present in allocation of time to training HPDC-beginners. The NSF has recognized this problem and is seeking to rectify it. One such program is "The advanced cyberinfrastructure research and education facilitators virtual residency: Toward a national cyberinfrastructure workforce" [4].

In addition to the systemic factors above, there are also many further obstacles which contribute to making available training materials often inaccessible for polar scientists. For instance, relatively simple things, like command line scripting, can be a major barrier for new HPDC users that lack a formal computer science background. Similarly, requesting time on HPDC resources often requires users to answer questions that they are ill prepared to answer, such as: "How many nodes do you need?", or "How many compute units are requested?".

In other words, whereas HPDC appears at the end of a comprehensive program in a traditional computer science curriculum, domain scientists are looking to use HPDC resources with relatively little programming experience outside of the specific language or program they may be using for their research. If learning HPDC takes too long, scientists under tight research deadlines are likely to fall back on highly inefficient local compute solutions (their laptop, for example). Therefore, there is a great need for HPDC training materials that are stripped down to only those elements that are "need to know" for a first time user of a specific domain. These initial elements are focused on the practice of HPDC with a bare minimum of HPDC theory or the specific hardware of the system they are using.

2) *A lack of knowledge of what resources are available:* In many cases HPDC resources (including: hardware, software, and human support) exist specifically to serve the domain sciences, and yet some specializations may have little to no knowledge of what is available.

Project such as US NSF funded XSEDE [5], are working to overcome this gap using such mechanisms as Campus Champions. Given this, we believe that after solving the training problem, researchers will be equipped to find the resources they need.

We believe a focus on the ALT-education elements will address many of the pedagogical issues discussed. And that this training should go beyond theory to incorporate relevant hands-on experience using HPDC resources.

III. Related Work

The NSF has long funded broad work in graduate training for interdisciplinary expertise via, among other mechanisms, the Integrative Graduate Education and Research Traineeship program [6]. This program ran from 1997-2013 and has more recently been succeeded by the NSF Research Training program [NRT] [7]. Both aim to develop institutional level multidisciplinary training tracks across all the sciences. These have been judged successful as far as their high level goals of creating "more interdisciplinary educational experiences", equipping students with "the professional skills relevant to working in the 21st century" and preparing students for "a wide range of careers"[8]. However, these programs do not specifically call out the need for smaller scale interdisciplinary training as discussed here. Nor do they extend to early- and mid-career scientists looking to learn new skills. Within the NRT program, there is also currently no project specifically targeting the interdisciplinary needs of polar sciences.

Looking at HPDC training programs specifically, Louisiana State University developed "A practical and comprehensive graduate course preparing students for research involving scientific computing" [9]. The curriculum covers a range of topics appropriate to an introductory level course, including SSH, OpenGL, version control, networks and data, simulations and application frameworks, scientific visualization, and distributed scientific computing. Alternatively, the University of Oklahoma developed "Supercomputing in plain English" [10] that has been used for many years by Oklahoma and other institutes [11]. Their curriculum includes the even more advanced topics of: shared memory, multi-threading, multicores, storage hierarchies, Instruction Level Parallelism, and compiler optimizations. Finally, XSEDE itself offers both off-line and online training focused on systems and software supported by their service providers. Topics offered include: "high performance computing, visualization, data management, distributed and grid computing, science gateways, and more." [12].

In the Louisiana State University course reported on in [9], attendees were largely drawn from computer science, systems science, or civil engineering graduates, yet it was found that even these domains required additional preparatory training. "Supercomputing in plain English", on the other hand, was targeted at non-programmers, but organizers still found that a 1 hr weekly expert follow-up with attendees was also necessary for up to two years following the course. Finally, a review

of the XSEDE training materials also reveals that it also assumes a base level of Computer Science knowledge that is unreasonable to expect of most polar scientists.

These lists of curricula topics offers a view into the complexity and depth that is considered necessary in an introductory level course to HPDC. Yet the implementation and real world experiences of teaching these courses confirms that traditional HPDC training materials are largely inaccessible to non-programmers. A counterexample to this common problem is provided by The Software Carpentry Foundation, which has been a dominant and successful pioneer of scientific programming training for people with non-computer science backgrounds. Over the past 2 decades SWC has "evolved from a week-long training course at the US national laboratories into a worldwide volunteer effort to raise standards in scientific computing" [3]. They have found that an investment of 25 hours of lectures, plus practical work, can improve productivity of non-CS graduate students by 20% [13]. Further, they report 80-90% of attendees are glad they attended and would recommend it to others. In 2013 alone, their materials were used to train 4300 scientists [3].

Software Carpentry, and its sibling Data Carpentry [14] - which focuses more on skills for analysis than programming - have used their extensive experience to refine the model into its current format: 2 days, host driven, face-to-face, essentials-only, practical, and feedback intensive. To enable use-scalability and efficient community updating and improvement, all materials are published under a Creative Commons license on Github. Their course content, however, does not currently include HPDC specific topics, although the concept of a 'HPC Carpentry' has been raised many times within the community [15] and beyond [16], and is the subject of multiple discussion threads online.

Believing it to be one of the several important components required to overcome the barriers between polar science and HPDC, we would like to see such a course developed in the near future. In its absence, however, as an RCN in 2016 we turned to another model of short and intense training, that of "hackathons".

Originally the domain of Silicon Valley software companies, hackathons were used to collocate normally disparate team members for a period of intense coding. The concept, however, has since been eagerly adopted by a wide range of fields, such that it has now evolved to more often refer to collocating a group of diversely skilled people (coders, designers, artists, scientists, and many other professions), for a similar short intense period of work.

Events are generally themed and focused on a given tool, challenge, or technology. Examples include the commercial and academic sectors engaging with each other and other professional communities outside their normal sphere such as: Barclays' competitive Financial Hackathon [17] that sought innovative technology based financial products, or the annual non-competitive MIT Grand Hack [18] that tackles health care challenges, or NASA engaging anyone who is interested in their annual Space apps challenge [19]. Alternatively, concerned citizens and bodies have run hackathons focused on specific social issues [20], [21], while the maker/hacker-space community use hackathons for everything from introducing K-12 children to basic electronics [22], through to the Science Hack Day [23] events which bring professional scientists and any other interested parties together to hack science problems. Many other examples exist, of groups using the hackathon model to target different demographics, and cover all age groups.

While hackathons were initially envisioned to produce innovation and productivity, a prominent and often noted by-product is the unique cross domain exchanging of skills and networking that can happen at such events. Students attending Major League Hacking [24] (a competitive programming hackathon league for university students) for example, equate going to events with going to the gym for a skill set workout [25]. Alternatively, [22] advocates for the model to be formally recognized as a practical means of supporting cross disciplinary collaboration in academia. Supporting this perspective, post-event surveys from participants at DataViz [26] (a 2014 polar datathon) rated the trans-disciplinary networking opportunity as the best part of the event.

IV. Our Experience with using a hackathon

Given our task and the challenges reviewed, as a first step in exploring solutions and gaging community response, we hosted a hackathon in July 2016 co-located with the XSEDE annual meeting in Miami, Florida. The experience taught us that the hackathon model is an effective but imperfect and incomplete means of bridging the divide between HPDC and non-programmer domain specialist users (in this case in the polar sciences). Specifically, as is elaborated on later, dedicated training input is still required, but we see the hackathon model as a valuable means of facilitating a path from taught theory to an accessible and applicable tool of relevance to specific users work. This section discusses the event, its successes, challenges, and the lessons learned.

A. Preparation

The RCN began by advertising a call for polar science proposals for a two day hackathon that would help the proposer use HPDC resource. Suggested example project domains included: running slow code on faster hardware or using parallelization techniques, visualizing large data set, and exploring statistical computing and data manipulation centric problems. In post event reviews, however, it was made clear to us that this was insufficient. That is, the "what" and "how" of HPDC technologies being of use to domain scientists needs to be better conveyed than just listing the above possibilities. Domain relevant specific example case studies at least, should have instead been given.

This advertisement was circulated to multiple organizational list including: APECS, NSIDC, Cryolist, ESIP, INSTAAR, and the RCN steering committee members. Additionally, eight key polar community members were requested to distribute

the advert to their networks, and the RCN's own members' networks, twitter account, and website were used.

We debated amongst ourselves whether or not to request proposals publicly (perhaps as Github issues as other hackathons have done, or a public Google word document), or privately. In the end we requested they be private, using EasyChair. In retrospect and based on attendee responses this was the correct decision for this community, as there is a valid fear of being "scooped" scientifically. We also believe personal emails to relevant professors would likely see more early career polar scientists hearing about the RCN activities and perhaps go someway to allaying fears.

In response to our advertisements we received seven proposals approximately two months prior to the event. Notably, three of these concerned a common challenge, performing segmentation and classification processing on remotely sensed sea ice imagery. Given the processing burden this analysis can incur it was determined that there is a real community need for a set of open source and accessible image processing tools to carry out such in a scalable manner on HPDC resources. These three teams were therefore invited to join a single team tasked with tackling this jointly.

Of the remaining four additional submissions, one was judged to be highly appropriate but indicated that the proposer already had access to and experience with HPDC resources. While it may well have been beneficial to have such a team join us, their experience, budgetary limits and the applicability yet lack of experience in other applications, lead to a decision involving their exclusion with the remaining three final submissions being accepted.

In light of the accepted proposals an allocation request was submitted to XSEDE requesting, 15,000 core-hours on Comet and 15,000 core-hours on Stampede per team. 25,000 core-hours were requested for the Polar RCN team to use for testing and development in the run-up to the main event. In total 180,000 core-hours were requested. The project has been granted 25,000 hours on each of the system totaling to 75,000 core-hours. This was more than enough for the event itself, leading to teams being able to continue using the systems for up to a year later.

With science team selection concluded the RCN team moved to advertise for programmers with HPDC experience to volunteer their time at the hackathon. This proved to be our largest challenge. We had hypothesized that: collocating our event with the XSEDE annual meeting, offering to cover costs, and the opportunity to network and develop new collaborations, would be enough to attract programmers. Further co-location would also allowing the domain scientists to attend the various HPDC related tutorials run at the meeting. We were, however, wrong on all accounts, and co-location with XSEDE was an expensive endeavor that was possibly not the best value for money option available to us.

Without knowing this prior to event, the above resulted in difficulty in recruiting programmers to attend. We had a very limited response after advertising for programmers via multiple XSEDE mailing lists, with the Computer Science departments at the Florida State University and Miami University, and through direct emails to targeted XSEDE Campus Champions at Universities with known strong polar science expertise. Further efforts were therefore required involving emails to those within the RCN team's own networks were needed. This was less than ideal as many of the people invited to come via this second round of advertising are already known to us and therefore aware of the polar sciences. Ideally, we would like to have enabled the recruitment of programmers new to the polar sciences so as to most effectively grow the community.

In the future therefore, we propose to instead either: seek collocation with an event that has the task relevant programmers already in attendance (such as a Python, Machine Learning, or Image processing conference for instance), or - learning from the SWC community - to host the event at a venue with ready access to potential programmers and a specific HPDC resource, such as a Computer Science department or national HPDC laboratory.

Once we had sufficient programmers recruited, however, we were able to form teams. In other hackathon contexts, for various reasons, team formation can be left to only happen at the event, and may allows attendees to freely move between teams throughout. While in the end we also saw some changes to team make-up on the first day, based on the experiences reported in [26] and the nature of HPDC work, we chose to largely pre-form and assign programmers to teams. The theory behind this being that, given the context of HPDC problems, the teams would then have time to design, and prepare and download data.

Once teams were finalized, by way of introduction and kick-off the RCN hosted initial teleconference calls with each group separately. We also sent out links to tutorials on the basics of Github, bash, HPDC, and specifically XSEDE use. The kick-off calls did serve to initiate discussion, Github repository creation, and email threads, however, for the most part practical preparation of even high-level design work was left to take place at the event. Based on post-event responses we attribute this limited preparatory effort primarily to team uncertainty regarding how the hackathon was supposed to work, along with everyone already being very busy. Amongst the domain scientists too, the lack of an existing relationship with the programmers assigned to them and unfamiliarity with common software development practice, led to a reluctance to share project concept and planning information publicly online.

B. The event

We gathered 10 polar scientists and 11 programmers in four teams for two days of hacking, separated by one day of XSEDE tutorials. Attendees were welcome to then leave or stay on for the final two conference days.

The day before, those attendees who arrived early enough were able to share a meal and begin networking, but we began formally first thing in the morning with shared breakfast and an ice-breaker game. Despite the lack of trust we found in the

initial kick-off teleconferences, this meal and game - which were intended as a team building exercise - was rated in post-event surveys as the least valuable activity.

As an opening, team science proposers were all invited to give a five minute introduction to their project to the entire group. It was in response to this that the last team adjustments were made, after which people stayed in their teams for the remainder of the week. The remainder of this first day was then dedicated to hacking. At the lunch break we made time for a short report back and discussion with the whole group, looking for outsiders to spot potential sticking points that might be circumvented. Tea and Coffee were available at all times.

Due to the scheduling of the XSEDE meeting we were forced to break the following day for tutorials, however, teams met on their own to keep working in the evening regardless. On the following and final day of hacking the RCN gave no input up until lunch time at which point, with only one final hack session remaining, we briefly outlined some potential avenues for the future. These included upcoming possible paper and funding opportunities, the on-going availability of the XSEDE resources, and the availability of XSEDE Campus Champions. At a closing session, which was opened and advertised to the rest of the XSEDE meeting, each team presented their final project status and discussed successes, failures, and challenges.

The following briefly summaries these results for each team. More details and all work carried out can be found under their respective Github repositories hosted at Ref. [27] (http://Github.com/polar-computing).

1) Aerosol-Delta: Aerosols deposited on snow and ice can darken reflective surfaces, increase solar absorption, and subsequently enhance snow and ice melt rates. This project sought to map aerosols over Earth's cryosphere using the global land ice identification mask and monthly mean MERRA-2 aerosol data, by plotting seasonal and annual totals from 1980-present. Over the two days the team was able to develop the job submission and ingestion scripts necessary to process several Earth system science data sets including: aerosol observation, modeled/reanalysis data, and land ice masks. Once ingested data set statistics, and visualizations could be created using Python libraries. Use of HPC systems resulted in a 10x speedup in processing times. However, as the data sets under consideration span 11TB, the majority of data could not be loaded onto the system within the hack-period. Leaving much more work to be done, and a lesson learnt regarding the need to download data to a system prior to the event.

2) Parallel-OBLIMAP: General Circulation Models (GCM's) are coupled with ice dynamic models to simulate the complex feedback mechanisms that exist between the ice caps, atmosphere, and oceans. A custom-built package (OBLIMAP [28]) has been designed for this purpose and affords a dramatic performance gain that allows for fast embedded on-line coupling of an ice model within a GCM. This project aimed to parallelize a hot spot in the OBLIMAP pipeline. Written in Fortran, this project was the most classically suited to HPC at the hackathon. Consequently the team was able to: rapidly port to XSEDE resources, load balance the processes, optimize the I/O for minimal contention, and parallelism the outer loop on the code's core nested loop's outer loop. While further work remained they immediately demonstrated strong performance scaling results, achieving 65% parallel efficiency on 16 cores.

3) Seal imaging: This project aimed to understand the distribution of Weddell seals on the Antarctic Peninsula and their movements within and among years by tracking them with opportunistic photography sourced from researchers and tourists. Such photo catalogs have proven highly successful for other species. However, given the non-standard characteristics of the dataset, and the fact that individual spot patterns can be distorted and degraded by the posture of the animal, folds in the skin, and even moisture, new techniques not currently used in the pattern matching community but well developed in visual computing were necessary. Over the course of the hackathon substantial progress was made on the pre-processing and segmentation phases of a recognition pipeline. The team used Python and OpenCV for rapid prototyping, and wrote a simple visualization application to review the results. This had the added benefit of boosting motivation through quick visual results.

4) Sea ice: High-resolution satellite and aerial imagery are increasingly used to provide assessments of the spatial/temporal coverage of various surface types (smooth ice, deformed ice, open water, melt ponds, etc) over the Arctic and Southern Oceans. An important challenge for the sea ice community is the segmentation and classification of these images into their constituent surface types. Given the scope of data tools have been developed to automate this process using various machine learning techniques, however, most are written in IDL which requires licensing and is therefore unsuitable for many national HPDC systems. The goal for the hackathon was to develop an open-source, high-performance computing (HPC) compatible toolkit alternative.

Within the hackathon, for supervised classification a Quick-shift segmentation algorithm and a Random Tree classification algorithm were implemented using Python libraries. An interactive feature selection tool was developed to label features for training data. And lastly, readers for several different file formats were developed. While far from complete, the conclusion drawn was that such an open pipeline was both worthwhile pursuing and relatively easily realizable, in so far as all the required quality open source libraries already exist.

C. Outcomes

Table I provides a summary of the event in numerical terms and a social media based account of the event is available at [29]. Regarding intangible outcomes, in the post-event survey the largest request was for more time. Despite the majority of respondents agreeing that the objectives were reasonable given the time allotted, afterwards only 35% (6) of participants considered the allocated time sufficient to reach what were the goals in their view, although 60% indicated that the majority of objectives had been met.

TABLE I

*LINES OF CODE WRITTEN AND COMMITS MADE FROM THE TIME OF HACK REPOSITORY CREATION TILL THE END OF THE HACKATHON-XSEDE WEEK (22 JULY 2016). 3 OUT OF 4 REPOSITORIES HAVE SUBSEQUENTLY SEEN SIGNIFICANT CONTRIBUTIONS BEYOND THIS DATE THE DATA FOR WHICH HAVE BEEN EXCLUDED HERE. **AS AN EXISTING CODEBASE CONTAINING THOUSANDS OF LINES OF CODE, THE LOC COUNT GIVEN HERE REPRESENTS THE DIFFERENCE IN LOC BETWEEN GIT REPOSITORY ON 21 JULY (LAST DAY OF XSEDE CONFERENCE AFTER HACKATHON) AND 16 JUNE (DAY PRECEDING THE HACKATHON)

	LOC written*	Commits*	Languages
AerosolDelta	602	70	Python
SeaIce	706	60	Python
Parallel OBLIMAP	80**	29	Fortran 90, C, Shell, Make
3DSeals	284	75	Python

The majority specifically indicated more hacking time (an extra day or two) would be the most useful addition, with one person suggested scheduling the days so as to not overlap with the co-located event at all. Given that there were also requests for more pre-event training, we surmise that if domain science participants had been exposed to even just a first level introduction to topics such as using the command line, version control, and job submission, progress would have been considerably accelerated.

On average participants indicated that they had made two new collaborations each and, overall, they felt engaged, empowered, and that the event was well facilitated. Some of their suggestions for future events included: having some dedicated relevant-package-specific tutors (such as SciDataKit experts) present, more snacks and budget, more pre-event training, and greater input and clarity regarding possible continuation and follow up plans.

Based on the above, what we wish to highlight to the community in this paper, is the potential for using a hackathon as a training and community building event. That is to say, that while many hackathons aim to produce specific products, we found the format to also be an effective (albeit incomplete) means of overcoming some of the challenges discussed in bridging gaps between the HPDC and polar science communities. Clearly, however, pre-event training, trust building, data preparation, and a more appropriate location or co-location event are key issues to be addressed.

V. CONCLUSIONS AND FUTURE WORK

In summary, nowhere is climate change more dramatic than in the polar regions. Studying these regions involves the interactions between an array of specialized domain processes, and multiple heterogeneous data sets that have never been larger. The domain therefore requires HPDC scale resources and yet the broader polar community is largely not currently able to fully utilize them. Prior community engagement and experience have led us to conclude that this is due to a required skills barrier to entry and lack of awareness of available resources. There are other communities addressing the latter, while we have been tasked with addressing the former via our Research Coordination Network.

We have determined that the skills barrier in the polar community is due to: (i) the rate of change in HPDC technologies causing a training lag, (ii) the momentum that exists behind the use of any established methodology, (iii) a lack of domain appropriate training materials, (iv) the fact that polar scientists have an even more restricted time budget for such training than other communities due to required remote field work, and (v) insufficient HPDC support staff on campuses.

While the community is aware of the gap and the broader challenge of equipping interdisciplinary domain scientists - efforts have thus far missed this particular niche. And while efforts such as SWC are successfully addressing the need for introductions to scientific programming, they fall short of teaching the skills needed for HPDC work. Therefore, in our first attempt to address this gap between the polar community and HPDC use, we used the hackathon model to bring HPDC experts and polar scientists together in a face-to-face and interactive environment. This enabled a focused period for skills and knowledge exchanges to take place - in both directions - through practical hands on work, using the problems most relevant to the particular domain scientists.

As a RCN we learnt many valuable lessons about the communities we are trying to bridge. Particularly, the community interaction afforded by organizing and running the hackathon clarified the reasons behind the divide, and the hackathon itself was shown to be an effective means of overcoming some but not all of these challenges. We have concluded, and intend to test in the future, that the use of a short, intense SWC-like introduction to scientific computing remains necessary to maximize the impact of the hackathon.

Following this, we would strongly suggest the creation of a HPC-carpentry course that would introduce the further necessary concepts for HPDC work. Specifically, while there are many available resources for HPDC training, unlike prior efforts we suggest that collectively an ALT-HPDC approach be used. Such a course would develop HPDC Awareness, Literacy, and following the SWC ethos of teaching only the essentials - be Tailored. That is, it should start with the minimal required basics, be no longer than two days, and crucially involve practical and domain specific examples on HPDC resources. Further, it should be run with and customized for a specific HPDC resource in mind, one that will remain accessible to the users beyond the course duration. That system's specifics (resource requests, job submission, job monitoring, etc.) should be taught so as to equip the users for immediate follow on work, with the goal being to ensure that a user is sufficiently conversant with the domain to move forward. We are currently exploring how both a SWC and an HPC carpentry equivalent might be run within the critical time limits polar science field work demands impose on researchers.

Based on prior work and literature, such courses would also ideally be followed up on by regular, direct HPDC-expert input over the next 1-2 years. However, where this is not possible given financial or time resource limits - such

as field campaigns impose - we propose that a follow up hackathon type event might serve as a quicker alternative mechanism. Such an event is short and moves users beyond simple coursework examples to working on their specific daily science problem on an accessible HPDC resource. Further, the format allows for cross domain networking that can lead to future work and facilitation of work via relationships that would otherwise never have formed.

In our experience, the hackathon model is a valuable means of bridging the divide between science users (such as polar researchers) and HPDC resource operators, tool builders, managers, and funders.

ACKNOWLEDGEMENT

This work is supported by NSF ICER 1542110 and associated collaborative awards. The authors would like to thank the many participants of the Hackathon and XSEDE'16 Organizing committee for their support.

REFERENCES

[1] F. Nelson, K. Kobak, and B. Sinclair, *Polar Regions (Arctic and Antarctic)*. Cambridge University Press, 2007, no. January.

[2] "Polar-HPDC Workshop," https://sites.google.com/site/polarhpdc/, accessed: 2017-1-26.

[3] G. Wilson, "Software carpentry: lessons learned," *F1000Res.*, vol. 3, p. 62, 19 Feb. 2014.

[4] H. Neeman, S. K. Ramadugu, A. Romanella, J. Rush, A. H. Sherman, B. Stengel, D. Voss, A. Bergstrom, D. Brunson, C. Ganote, Z. Gray, B. Guilfoos, R. Kalescky, E. Lemley, and B. G. Moore, "The advanced cyberinfrastructure research and education facilitators virtual residency," in *Proceedings of the XSEDE16 on Diversity, Big Data, and Science at Scale - XSEDE16*, 2016.

[5] "XSEDE — home," https://www.xsede.org/, accessed: 2017-1-20.

[6] "Introduction to the IGERT program — NSF - national science foundation," https://www.nsf.gov/crssprgm/igert/intro.jsp, accessed: 2017-1-23.

[7] "National science foundation research traineeship program (NRT) — NSF - national science foundation," http://tinyurl.com/z8bte7x, accessed: 2017-1-24.

[8] T. National, S. Foundation, and H. Resources, "EVALUATION OF THE INITIAL IMPACTS OF THE NATIONAL SCIENCE FOUNDATION ' S INTEGRATIVE GRADUATE EDUCATION AND RESEARCH Final Report E VALUATION OF THE I NITIAL I MPACTS OF THE N ATIONAL S CIENCE F OUNDATION ' S I NTEGRATIVE G RADUATE E DUCATION AND R ESEARC," *Education*, no. February, 2006.

[9] G. Allen, W. Benger, A. Hutanu, S. Jha, F. Löffler, and E. Schnetter, "A practical and comprehensive graduate course preparing students for research involving scientific computing," *Procedia Comput. Sci.*, vol. 4, pp. 1927–1936, 2011.

[10] H. Neeman, J. Mullen, L. Lee, and G. Newman, "Supercomputing in plain english: Teaching high performance computing to inexperienced programmers," in *Proceedings of the 3rd International Conference on Linux Clusters: the HPC Revolution 2002*, 2002.

[11] H. Neeman, H. Severini, D. Wu, and K. Kantardjieff, "Teaching supercomputing via videoconferencing," *Proc. TeraGrid*, 2008.

[12] "XSEDE — training," https://www.xsede.org/training1, accessed: 2017-1-26.

[13] G. Wilson, "Software carpentry: Getting scientists to write better code by making them more productive," *Comput. Sci. Eng.*, vol. 8, no. 6, pp. 66–69, 2002.

[14] D. Mimno, "Data carpentry," *blog, August, available at http://www. mimno. org/articles/carpentry/, accessed*, vol. 13, 2016.

[15] datacarpentry, "datacarpentry/hpc-carpentry," https://github.com/datacarpentry/hpc-carpentry, accessed: 2017-1-24.

[16] "High performance computing carpentry," https://github.com/hpccarpentry, accessed: 2017-1-24.

[17] "Hackathon diary — barclays," https://www.home.barclays/news/2016/09/hackathon-diary--36-hours--1-045-developers--and-a-whole-lot-of-.html, accessed: 2017-1-19.

[18] "MIT GRAND HACK 2016 - MIT hacking medicine," http://hackingmedicine.mit.edu/grandhack/, accessed: 2017-1-19.

[19] "Space apps," https://2017.spaceappschallenge.org/, accessed: 2017-1-19.

[20] "Smart communities hackathon," http://www.startupspark.co.za/2016/08/02/smart-communities-hackathon/, accessed: 2017-1-19.

[21] "International open data hackathon," http://opendataday.org/, accessed: 2017-1-19.

[22] J. Aboab, L. A. Celi, P. Charlton, M. Feng, M. Ghassemi, D. C. Marshall, L. Mayaud, T. Naumann, N. McCague, K. E. Paik, T. J. Pollard, M. Resche-Rigon, J. D. Salciccioli, and D. J. Stone, "A "datathon" model to support cross-disciplinary collaboration," *Sci. Transl. Med.*, vol. 8, no. 333, p. 333ps8, 6 Apr. 2016.

[23] "Science hack day," http://sciencehackday.org, accessed: 2017-1-19.

[24] "Major league hacking," https://mlh.io/, accessed: 2017-1-24.

[25] S. Leckart, "The hackathon fast track, from campus to silicon valley," *The New York Times*, 6 Apr. 2015.

[26] C. Mattmann, P. Ramirez, L. Mcgibbney, A. Pope, and J. Wyngaard, "Report on nsf dataviz hackathon for polar cyberinfrastructure," http://nsf-polar-cyberinfrastructure.github.io/datavis-hackathon/, accessed: 2017-1-24.

[27] "Polar Computing — github," https://github.com/polar-computing, accessed: 2017-1-30.

[28] T. J. Reerink, W. J. v. d. Berg, and R. S. W. v. d. Wal, "OBLIMAP 2.0: a fast climate model–ice sheet model coupler including online embeddable mapping routines," *Geoscientific Model Development*, vol. 9, no. 11, pp. 4111–4132, 21 Nov. 2016.

[29] "Polar-HPC hackathon 2016 (with images, tweets) r4space," https://storify.com/r4space/getting-started, accessed: 2017-1-25.