Hacking at the divide between Polar Science and HPC: Using hackathons as training tools

Jane Wyngaard*, Heather Lynch†, Shantenu Jha§, Allen Pope‡ and Jaroslaw Nabrzyski*
*Centre 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, Boulder, CO, Email:allen.pope@nsidc.org
§Departmet of Computer Engineering, Rutgers University
New Brunswick, NJ, Email:shantenu.jha@rutgers.edu

***Abstract*—Given the current scientific questions facing the globe, 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 other tools necessary for reproducible science. As a National Science Foundation funded Research Coordination Network, we have 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 customised 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 ones peers, and v) to be an incomplete training model that requires supplementation via Polar 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 useable products that can speed workflows both for those involved in the hackathon and others not involved in the hackathon itself.**

I. Introduction

There is an urgent need to equip the scientific community to use leading 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 allows 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 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 effectively.

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 [**?**]. 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 [1], 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.

The above challenges and the resulting gap have now been formally recognised 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 for domain scientists, which we refer to as ALT-HPDC education. ALT-HPDC training is focused on communicating whats 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 domains users.

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

Taking a lesson from the long running and widely successful introduction to scientific computing offered by Software Carpentry (SWC)[2], 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 already available through the existing documentation for HPDC resources.

We need to target both graduate students and established researchers, since many domain scientists realize mid-career that HPDC resources may be required. 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 on those skills that are only need to know, and re-packaging that information in small chunks, we believe can overcome some of the hurdles that are preventing a more pervasive uptake of HPDC technologies in the domain sciences. In the Polar community specifically, it is often easiest to carve out such time when researchers are in the field, since polar deployments can involve long periods of uninterrupted downtime. However, this means these researchers are also often working with limited to zero connectivity, a factor which must therefore be considered when designing training materials.

Recognising the importance of this gap particularly in the Polar sciences, the National Science Foundation has funded a number of initiatives tasked with addressing it, amongst them being a Research Coordination Network (RCN) that we lead. The remainder of this paper is as follows; Section II describes the reasons behind the HPDC-Polar gaps 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 the hackathon model for the Polar domain, 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 science training generally lag far behind.
   b) **Momentum:**Senior researchers are often 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 traditio" in terms of 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 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.
   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 recognised 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"[3]

In addition to the systemic factors above, there are many illustrations of 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 (such as How many nodes do you need?, How many compute units are requested?) that they are ill prepared to answer.
The inaccessibility of the materials 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 domains 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. In addition to this language barrier, HPDC training materials usually prematurely focus on the details of the hardware, and fail to communicate why these details are of relevance. Emphasis should 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, training materials often 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.

In other words, whereas HPDC appears in the traditional curriculum at the end of a comprehensive computer science training program, domain scientists may be 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 only to 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.

National bodies, such as the USAs XSEDE [4], tasked with curating HPDC resources, 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.

Therefore, 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 [5]. This program ran from 1997-2013 and has more recently been succeeded by the NSF Research Training program [NRT] [6]. Both aim to develop institutional level multidisciplinary training tracks across all the sciences. These appear to have been successful where run, however, they do not incorporate 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 targeting the Polar Sciences interdisciplinary needs.

Looking at HPDC training programs specifically, Louisiana State University developed A practical and comprehensive graduate course preparing students for research involving scientific computing[7]. 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 [8] that has been used for many years by Oklahoma and other institutes [9]. Their curriculum includes the even more advanced topics of; shared memory, multithreading, multicores, storage hierarchies, Instruction Level Parallelism, and compiler optimisations. Finally, XSEDE itself offers both offline 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.[10].

In the course at Louisiana State University, 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, yet organizers found that a 1 hr weekly expert follow-up with researchers was necessary for up to two years following the course. And a review of the XSEDE training materials 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 confirm 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" [2]. They have found that an investment of 25 hours of lectures, plus practical work, can improve productivity of non-CS graduate students by 20% [11]. 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 [2].

Software Carpentry, and its sibling Data Carpentry[12] - 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 [13] and beyond [14], and is the subject of multiple discussion threads online.

We would like to see such a course developed in the near future, and believe that such a course will be one of several important components required to overcome the barriers between polar science and HPDC. Currently, however, in the absence of any available training programs in this vein, as an RCN 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 has 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, or broad challenge. Examples include; the annual non-competitive MIT Grand Hack [15] tackling health care challenges, Barclays competitive Financial Hackathon[16] that sought innovative technology based financial products, through to hackathons for community social impact [17], [18], or exploring new angles on science and engineering with NASAs annual Space apps challenge [19]. Finally, the maker/hacker space sees many small to large hackathons, from those introducing K-12 children to basic electronics[20] through to Science Hack Day [21] events that bring professional scientists and any other interested parties together to hack science problems. Many other examples 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 [22] (a competitive programming hackathon league for university students) for example, equate going to events with going to the gym for a skill set workout [23]. Alternatively, [24] advocates for the model to be formally recognised as a practical means of supporting cross disciplinary collaboration in academia. Supporting this perspective, post-event surveys from participants at DataVis [25] (a 2014 Polar datathon) rated the transdisciplinary 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 gauging 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). This section discusses the successes and challenges and consequently the lessons learnt.

A. Collocation

We hypothesised that collocating our event with the XSEDE annual meeting would attract programmers while simultaneously also allowing the domain scientists to attend various HPDC related tutorials. Interestingly, however, neither concept was true, according to our post event survey. The domain scientists indicated that the tutorials were mostly unhelpful and the programmers simply indicated that free conference attendance - in this case - was not a significant motivator for attending the hackathon. In the future we believe collocation with an event that has the task specific programmers already in attendance (such as a Python, Machine Learning, or Image processing conference for instance) would be more beneficial. Or, learning from the SWC community, that the event should be hosted at a venue with ready access to potential programmers and a specific HPDC resources, such as a Computer Science department or national HPDC laboratory.

B. Teams and Tasks

We gathered 10 Polar Scientists and 11 Programmers in 4 teams for 2 days of hacking, separated by 1 day of XSEDE tutorials. Polar Scientists applied to attend by proposing a Polar Science challenge they believed might benefit from HPDC resources. By grouping 3 applications all proposing to address the same domain problem (sea ice feature classification in satellite imagery) we were able to accommodate 6 of the 7 applications received.

On application to XSEDE, we received an allocation of 75,000 core hours on various XSEDE systems for the event. 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.

The following briefly summarise each teams scientific and computational challenge and progress. More details and all work carried out can be found under their respective GitHub repositories hosted at 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 2 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 visualisations 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::

3) Seal-Imaging:: his 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 visualisation 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 Quickshift 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 whortwhile pursuing and relatively easily realisable, in so far as all the required quality open source libraries already exist.

C. Outcomes

Table IV-C provides a summary of the event in numerical terms and a social media based account of the event is available at [26]. Regarding intangible outcomes, in the post-event survey the largest request was for more time (64% of attendees). This is likely due to the large scope of the projects. However, given that there were also requests for more pre-event training, we also 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. Future event length determinations will therefore likely depend on both factors.

	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

TABLE I
*LINES OF CODE WRITTEN AND COMMITS MADE FROM THE TIME OF HACK REPO CREATION TILL THE END OF THE HACKATHON-XSEDE WEEK (22 JULY 2016). 3 OUT OF 4 REPOS 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 REPO ON 21 JULY (LAST DAY OF XSEDE CONFERENCE AFTER HACKATHON) AND 16 JUNE (DAY PRECEDING THE HACKATHON)

Overall, participants felt engaged, empowered, and that the event was facilitated well, indicating that in general it is a good model to follow in the future. Their suggestions for future events included; having some dedicated relevant-package-specific tutors (such as SciDataKit experts) present, 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.

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 specialised 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 utilise 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 NSF 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 at address this gap between the Polar community and HPDC use, we utilised 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 skill 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 organising 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 reasons. 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 increase 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 2 days, and crucially involve practical and domain specific examples on HPDC resources. That is, such a course should be run with a specific HPDC resource in mind, one that will remain accessible to the users beyond the course. That systems 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.

We conclude by stating that 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.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] "Polar-HPDC Workshop," https://sites.google.com/site/polarhpdc/, accessed: 2017-1-26.
[2] G. Wilson, "Software carpentry: lessons learned," *F1000Res.*, vol. 3, p. 62, 19 Feb. 2014.
[3] 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.
[4] "XSEDE — home," https://www.xsede.org/, accessed: 2017-1-20.
[5] "Introduction to the IGERT program — NSF - national science foundation," https://www.nsf.gov/crssprgm/igert/intro.jsp, accessed: 2017-1-23.
[6] "National science foundation research traineeship program (NRT) — NSF - national science foundation," http://tinyurl.com/z8bte7x, accessed: 2017-1-24.
[7] 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.
[8] 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.
[9] H. Neeman, H. Severini, D. Wu, and K. Kantardjieff, "Teaching supercomputing via videoconferencing," *Proc. TeraGrid*, 2008.
[10] "XSEDE — training," https://www.xsede.org/training1, accessed: 2017-1-26.
[11] 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.
[12] D. Mimno, "Data carpentry," *blog, August, available at http://www. mimno. org/articles/carpentry/, accessed*, vol. 13, 2016.
[13] datacarpentry, "datacarpentry/hpc-carpentry," https://github.com/datacarpentry/hpc-carpentry, accessed: 2017-1-24.
[14] "High performance computing carpentry," https://github.com/hpccarpentry, accessed: 2017-1-24.
[15] "MIT GRAND HACK 2016 - MIT hacking medicine," http://hackingmedicine.mit.edu/grandhack/, accessed: 2017-1-19.
[16] "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.
[17] "Smart communities hackathon," http://www.startupspark.co.za/2016/08/02/smart-communities-hackathon/, accessed: 2017-1-19.
[18] "International open data hackathon," http://opendataday.org/, accessed: 2017-1-19.
[19] "Space apps," https://2017.spaceappschallenge.org/, accessed: 2017-1-19.
[20] "HackHolyoke 2016," http://hackholyoke.com/, accessed: 2017-1-26.
[21] "Science hack day," http://sciencehackday.org, accessed: 2017-1-19.
[22] "Major league hacking," https://mlh.io/, accessed: 2017-1-24.
[23] S. Leckart, "The hackathon fast track, from campus to silicon valley," *The New York Times*, 6 Apr. 2015.
[24] 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.

[25] "NSF polar DataVis hackathon," http://nsf-polar-cyberinfrastructure.github.io/datavis-hackathon/, accessed: 2017-1-24.
[26] "Polar-HPC hackathon 2016 (with images, tweets) r4space," https://storify.com/r4space/getting-started, accessed: 2017-1-25.