>> Welcome, Ewa. Ewa is here to talk about Pegasus and its sustained science impact. And, I'd like to begin this introduction with a personal story regarding Pegasus. Way back in 2003, a colleague, Mike [inaudible] and I were putting together computational pipelines for bioinformatics. I, myself, had a workflow technology back then, and I was explaining to Mike how useful it would be to have a system to quickly configure these pipelines, a workflow system that would also map the pipeline to whatever network computing resources were available. We had almost decided to build the workflow system out ourselves, and we were sure, cover ourselves with fame and glory. But, Mike did some research and gave me the bad news that actually Pegasus already existed, and it pretty much did what we wanted a workflow system to do. So, no fame and glory for us, but we did get some good science done, and done faster, and at a greater scale thanks to Pegasus. To give some background about Ewa, Dr. Ewa Deelman received her Ph.D. in Computer Science from the Rensselaer Polytechnic Institute in 1998. Following a post doctorate at the UCLA Computer Science Department, she joined the University of Southern California Information Sciences Institute, ISI, in 2000 where she is now serving as the Research Director and is leading the Science Automation Technologies group which explores the interplay between automation and the management of scientific workflows including the resource provisioning and data management of these workflows. She's also a research professor at the USC Computer Science Department and IEEE Fellow. She pioneered workflow planning for computations executing in distributed environments. Her group has led the design and development of the Pegasus workflow management software and conducts research in job scheduling and research provisioning and distributed systems, workflow performance modeling, provenance capture, and the use of cloud platforms for science. The Pegasus project started in 2001 and has users in astronomy, earthquake science, botany, chemistry, climate modeling, computer vision, genomics, helioseismology, limnology, neuroscience, ocean science, and physics. That concludes my introduction. Before I turn it over to Dr. Deelman, a reminder to everyone, email questions to me, rramnath@nsf.gov. So, with that, over to you, Ewa. Welcome again.

>> Thank you very much, Rajiv. Thank you for this nice introduction and for invitation to this inaugural OAC webinar. It is my true privilege to be here. Today, I will talk about the Pegasus workflow management system and how it was conceptualized, how it was dissolved and sustained since 2001. Since its beginning, Pegasus has been a close collaboration with Miron Livny and his HTCondor team at University of Wisconsin, Madison. One of the projects that we've been involved with since a very long time is LIGO which is a gravitational wave observatory project. And LIGO has been making a lot of great discoveries lately. In particular, in October of last year, LIGO and Virgo made the first detection of a gravitational wave that was produced by colliding neutron stars. What is really incredible about this detection is that, although LIGO detected gravitational waves in the prior year, this was the first detection that resulted from a collision of binary neutron stars which can be visible also by other instruments other than LIGO and Virgo's, other than LIGO and Virgo. So, for example, at the same time that LIGO took this collision, NASA's Fermi space telescopes also detected a burst of gamma rays from the same region of the sky. So, this basically kicks of a new area, new era of multi-messenger astronomy. And, at that time, LIGO and Virgo detected the gravitational wave, they were able to triangulate on a particular area of the sky thanks to the free detectors being

distributed geographically. And, they were, they communicated this discovery to the partners which, where were able to retarget the telescopes to a particular area of the sky. And, you can see on this graph, the images that those telescopes captured of this event several hours and even days after the event occurred. So, you see some event captures the, this region of the sky ten hours after the event. Some on the right, like Chandra. So, that event nine days out. The, this was quite an incredible discovery, and if we look at our partnership between the Pegasus project and LIGO at the time, we started in 2001, and we've been working together, evolving our knowledge of science and cyber infrastructure along the way. In 2001, we developed the first Pegasus prototype to help LIGO scientists, in this case, do a search of known binary pulsars. And, this took us to 2017 when LIGO has won the Nobel Prize in Physics. Over the years, we had varied successes. For example, in 2011, LIGO detected a gravitational wave that came from a blind injection. So, that was a fake signal that was injected by a small group of LIGO collaborators into the data stream. And, the goal of this blind injection was to figure out whether the science, the methods that LIGO has developing and the cyber infrastructure that they're relying on were able to detect a gravitational wave should one occur in the data stream. So, although the LIGO detectors at the time were not sensitive enough to pick up the detection, the science behind it and the cyber infrastructure were able to see the fake gravitational wave signal in the data. So, that gave the scientists confidence that when a real event occurred, they would be able to be there and detect it. And the first detection was announced in 2016. This detection came as a result of a collision of two massive black holes 1.3 billion years ago. And then, since then, they also detected other gravitational waves coming also from black holes and recently from binary neutron stars. If we look at what it takes to actually conduct analysis of such a detection, and in particular this is looking at the analysis that was put in place for the first gravitational wave detection that occurred. The signal was first seen in September of 2015, and the announcement came out in February of 2016. What it took to make the decision that the signal was actually a real signal, it took basically three main elements. It took the science. The science workflow that the LIGO scientists have developed, and this workflow was designed to take the data from the two barometers that LIGO had operational at the time and measure the statistical significance of the signal that we were seeing in the data. To do that, LIGO had to run almost 21,000 of these workflows, and cumulatively, the tasks, the workloads, were combined number of tasks. In these workflows was about 107 billion. So, in order to be able to manage these tasks from distributed infrastructure, they used the Pegasus workflow management system to manage the execution of individual tasks and the movement of data in and out of the computational pipeline. LIGO also relied on the distributive power of their own computing infrastructure. So, they run clusters in U.S. and in Europe. They also relied on the power of Open Science Grid, the resources of XSEDE and Blue Waters. So, we, three key factors were contributed to LIGO's discovery of a first gravitational wave and subsequent discoveries. So, looking back at the Pegasus project and what I think has made us successful in building dependable cyber infrastructure, I really believe that it takes three key elements. One is computer science research. So, we ground our ideas and concepts in computer science. We engage with different domains of science, basically want to make sure that the software we develop is relevant to the needs and meet the needs. And, we also want to work with multi domains because we want to have a breadth of different use cases to work with. And, obviously, it's also great to have impact on these different domains in return. And, then, the third element is the development of the actual sub infrastructure, and in important ingredient of that is the reuse of existing and dependable cyber infrastructure technologies. This basically allows us to focus on the innovative aspects of what we want to do, rather than having to reinvent the wheel. And, obviously, all these three ingredients need to work together to bring together a product that's usable by a broad community. Another ingredient that's key to this are people. So, building, we build a team of over the years of dedicated software developers. We also have graduate students that contribute to our project. Karan Vahi, for example, has been with me since 2001. So, he's been the lead architect on Pegasus on the developer for the last 17 years. And, others contributed to this effort quite heavily as well. In particular, we've had a number of different developers that came to our group, GRAs that were doing, that did their Ph.Ds. in the group and have contributed to the research. We also have postdocs, a number of masters students, and what we thought was very useful is to have visitors come to our group from different groups, from different other group, research groups so that we have a cross pollination of ideas. Many of the people that contributed to Pegasus now work at Amazon, Google, NetApps, and SpaceX. And, in addition to people that have come through group, we also collaborate quite heavily with other computer scientists and domain scientists, and I believe that these collaborations really enrich our projects. So, how did Pegasus start? Well, we started with a concept that was, came out of database community, in particular view materialization. And, this concept was, the virtual data grid model in which a user could come to the system, and they could ask, for example, for certain amount of data, certain type of data. And, they, the system would deliver to, this data to them without the user needing to know whether the data was already precomputed someplace or whether it needed to be computed on demand. So, we went from this high level concept, and then the challenge was how do you translate this valuable concept to something that meets the science needs. So, in the case of LIGO, we came up with this scenario where a LIGO scientist would come into the system. They will, say, conduct a pulsar search on the data collected during a particular timeframe. And then, the system for each of these requests would try to understand what the request means in terms of how do you materialize it. It would determine if this request was already instantiated. So, maybe somebody already asked for that particular data, and it was already available on some storage. And, if not, it would have to plan out the data movement and computations required to obtain the results and to execute the plan. So, then, we, back in 2001, we developed this interface for LIGO where the scientist could come in, and if there was put in the various timeframes that we're interested in. And, they'll say where they wanted the output of the data to go. So, our challenge became how to translate this high level request into something that would be executable by a piece of software. And so, at the time, we explored various AI planning techniques to do this translation from a high level request to what we call the recipe of what became really a workflow that you see on the right. And to, then, determine how to step through this recipe that takes the raw data from the instruments. It transforms it into short timeframes. Then, coalesces it into transferred into longer timeframes and looks for the signals of gravitational waves in this data. So, as we propose these techniques to LIGO scientists, it turned out that they really didn't want the interface that you see on the left, this high level description, but really wanted to interact with the workflow that you see on the right and tune it, attune the science based on this concept of workflows. So, what was really lost in this translation of going from the idea to a deliverable piece of software was this high level abstraction. So, that was not appropriate for this project. However, it also gave us a new research direction, and we focused on the management of these workflows that you see on the right in distributed environments, environments where LIGO led to other data distributed where they had resources all over the globe. So, we also, at the time it was 2002, 2003, were engaged with other science projects, specifically the Montage project out of IPAC Caltech. And, also, the earth simulation projects, CyberShake out of the Southern California Earthquake Center at USC, and based on these three applications, we saw the challenges to workflow management as follows. So, one thing was that the users needed to describe the complex workflows in a simple way. Not as simple as this web interface, but still a way that allows them to use languages that the computer languages that I used. At that time, it was Java and Perl. They need to access distributed and often heterogenous data and computational resources, and they need to deal with resources and that software changing over time. So, for example, for Southern California Earthquake Center, since 2007 has been working on nine different HPT systems funded by NSF, and they needed to move from one to the next in a seamless fashion. So, given these challenges, our focus has become, then, to look at three main issues. One was the separation between workflow description and workflow execution. So, we really felt that being able to still describe the workflows on an abstract level without putting any information about actual resources, was a useful thing, because, then, you could target this workflow to different types of sub infrastructure. And then, if you have this assumption, then you would have to do some workflow planning to figure out how do you take this abstract workflow and map it onto the resources. And, once you've done the mapping, you need to be able to schedule the actual jobs in a way that performs well and is scalable. Once you have, you're down to the actual job execution, the task execution, then you need to deal with issues of monitoring photons and debugging. And, if we look at, you know, since I've been talking about workflows as being a useful concept for science. So, this is a slide from the earthquake scientists out of USC, and basically, he summarized the benefits of scientific workflows in the following way. First of all, it allows you to conduct a series of computational tasks so that you can put larger computations together based on building blocks. You can change, chain these together so the output of one computation becomes the input of another. And, this replaces the manual handoffs that people were often using. Basically, sending email to the colleague and saying, "I did the calculation, and the data is available at this url." It also gives you ease of use. So, if you have an expert putting together workflow in a scientifically meaningful fashion, they can, then, share this workflow with other colleagues or with graduate students so that they can reproduce the results. And, provides, also, a framework to host and assemble these community codes so you can take expertise from various people and put them together into a larger whole that can be also multidisciplinary. So, now, you can cross boundaries between different disciplines within a particular workflow. And, in general, because you see, if you see the benefit of putting these community codes together, then this also provides a framework and encourages people to design common formats and standards which enable this type of composition. So, if we look at the typical environment that the scientist has access to, they have their own local resource. They have some idea of what they want to do. So, this workflow, work definitions which we often find on a piece of paper or on a white board. They have some local data that they might want to integrate with other data sources, and they have, potentially, an instrument in the lab. In addition to the local resources, they also have the power of the national cyber infrastructure such as Blue Waters, XSEDE. They have the campus clusters. They have the Open Science Grid, some DOE facilities, and commercial and academic clouds. So, the question, now, becomes how do you bridge this local environment to that global infrastructure? I'm not using workflow technologies, and you have a simple workflow that you want to execute, and you see it on the top left of the workflows, the Hello World. Hello takes file m.a and generates file m.b which is, then, passed to the computation world which generates file m.c. So, if you take this workflow and, for example, you have access to tax Wrangler, there are several steps that you need to follow. The first of all, you need to log onto the machine. Then, you need to write a script that you'll submit to the queue, and you have to define various parameters. The maximum [inaudible] time, for example of a computation, the location where the executables are. You need to invoke the executables, and then you might want to copy the outputs out to different directory. Once you've written the script, then you need to find where your data is and bring it in to the platform so it's ready for execution. You need to, then, submit your script to Wrangler and put the results in a particular queue. And then, finally, you have to wait until the execution finishes, and then you can stage out the data for further analysis or to share to somebody else. Now, what happens if Wrangler goes down? For example, for maintenance or if it gets a commission? What if your job crash during execution? What about if you want to run on multiple platforms? So, these are the challenges that scientists often face. So, our solution to that is to place a workflow management system on a local resource, this local resource that's available to the user. We sometimes help users structure their workflows in a coherent fashion, and you see the definition, in this case, of a Montage collected framework on the left. And the, Pegasus works from this local environment through the interfaces to HTCondor to the national cyber infrastructure and other distributed resources. So, it's scheduled to work onto these resources. It also manages the data movement from the local storage and also across the different storage available in the distribution environment. So, Pegasus today. How does it look like? So, today, scientists describe the processes for workflows at the logical level without including the details of our underlying cyber infrastructure. And, they use various languages that they're familiar with to do so. So, we provide APIs in Java, Perl, [inaudible], and recently, we also provided interfaces is the Jupyter Notebook to facilitate the compositions such as Hello World that you see on this slide. The workflows operate at the level of individual files and individual codes. And then, Pegasus takes this high level workflow and maps it on, finds out what resources are available to the user, and then, it maps this abstract workflow to what we call an executable workflow which you see on the right. And, during this mapping, it infers the data transfers. So, it adds notes to the workflow to stage data in and out of a computation. It creates the directories on the remote system. So, to isolate better workflow execution. It also registers data in the data registry if figures use such an instance. And then, finally, once we have this executable workflow, Pegasus runs it on a variety of resources that I already described in the workflow. It also, of course, generates the submit files that I needed for this execution, and does all this management on behalf of a user. If errors occur, Pegasus tries to recover from them. It prepares ways. It tries to retry individual jobs or replan the entire workflow if necessary. Oftentimes, users also don't see the actual workflow that they're using. So, sometimes, the workflows are hidden behind a user-facing portal. So, for example, Pegasus is part of NanoHub and also HUBzero. Also, Pegasus provides the workflow management capabilities to other workflow composition tools such as, for example, Wings out of USC. So, we built Pegasus on very specific computer science principles. The main concept that we use in Pegasus is the concept of directed acyclic graphs. And, basically, there's a lot of, we've structured the workflow at a DAG, and given that structure, we can, then, reuse a number of graph traversal algorithms that are already well published in the literature, note clustering algorithms, pruning, and other complex graph transformations. We also use hierarchical structures in DAG, so we can put a DAG within a DAG. And, this allows us to achieve scalability, so we can run workflows that have millions of nodes that if we put the workflow with the workflow, then the Pegasus assigning engine just needs to work at a subworkflow level, not at the entire one million jobs at a time. We also use recursions. So, being able to have a DAG within a DAG allows us to have a deep, hierarchies in the workflows. And, we also are enabled, enabled rewriting of the subworkflows that have not been executed yet. Which gives us a dynamic behavior that, with which we can mimic, for example, loops and other types of constructs as well. As a result of using directed acyclic graphs, we also developed new algorithms and task clustering and data placement, sufficient use of data in workflows, data reuse, resource usage estimation, resource provisioning, and recently, we've also been looking at supporting in situ workflows in HPC, high performance computing environments. And, this is work done in collaboration with [inaudible] out the of the University of Delaware. Over the years, we published quite a bit. So, these are the publications related to Pegasus since 2001, and the colors indicate the different types of areas that we've published in. I believe that publications are important for dissemination and education of the next generation of cyber infrastructure researchers and developers. They also help with workforce development, and they put you on a good career path. Especially being a faculty, research faculty in the computer science department, one of the ways I'm being judged is on my publication record. And obviously it helps with funding as well, and you can bring in research funding into cyber infrastructure efforts and enhance the funding that you obtain from various sources. Another key to success, I believe, is leveraging a proven cyber infrastructure component. And, in particular, we've leveraged HTCondor's capabilities of job submission to heterogenous and distributed resources. The fact, also, that they can manage dependencies that are expressed as DAG, so in DAG. And, they have capabilities for job retry and error recovery. So, having Condor, HTCondor take care of these things for us, we're able to innovate and focus on other aspects of automation. In particular, we looked at and explored workflow planning and replanning in case of failures. We explored automated data management in scientific workflows. We're able to devolve specialized workflow execution engines that take high throughput computing workflows, the single core jobs, and are able to work to execute them efficiently within HPC systems. We're also able to provide users with different workflow composition APIs, and then developed user friendly monitoring and debugging tools that scientists can use to explore the behavior of a workflows as they're running. We also added provenance tracking capabilities that can help with reproducibility. And, recently, we also explored the issues of data integrity within the SWIP project which is a collaboration with Indiana University and RENCI. Another important part of what we did in Pegasus, I believe is the use real applications in our work that provide us a realistic testing ground and a good evaluation of our software. In particular, we've been using Montage since 2002, and this has been a great collaboration with folks at Caltech, Bruce Berriman and John Good. And, Montage is not only an important application for us, but also many other scientists and computer scientists in, scientists in astronomy have used it, but, also, computer scientists and CI developers have used it for their work. And, the reason is that Montage is an open source software. It's robust, and it's scalable. So, the workflows that you design with Montage can be scaled up and down based on what you're trying to test. The data that they operate on which is hosted in various astronomy archives is often open data. So, all this capability is easily available to everyone that wants to use it for testing the algorithms, for example, in workflow scheduling or resource provisioning or provenance tracking. And, people have used it quite extensively. And, we also use it in Pegasus as one of our applications that we test the software, build and test the software nightly. In general, I think it's very important to have also applications that can push the boundaries of what you can do. In past, it's been the state SCEC, Southern California Earthquake Center CyberShake application. And, this is an important application that tries to answer the questions of what will the peak earthquake shaking be over the next 50 years in a particular geographic area. And, in this case, the CyberShake focuses on southern California. And, this information, because southern California is the seismically active area. This information is very useful for building engineers, disaster planners, and insurance agencies. If we look at the recent achievements of CyberShake, in 2017 they used their resources at ORNL's Titan Machine and NCSA's Blue Waters to run workflows which were composed, which in total ran for 2., 21.6 million core hours and generated 777, so almost three quarters of a terabyte of data. And, in general, CyberShake, since 2007, as I believe I mentioned before, has run on nine different HPC systems and has consumed 100 million core hours which is over 11,000 years of computing. As SCEC has had various successes over the years, and I just want to highlight two of them. One was in 2010 where they generated the world's first physics-based probabilistic seismic hazard map of southern California, and that was quite interesting because they saw that the shaking of a basin, L.A. Basin was greater than originally anticipated. And, recently, in 2018, they incorporated an earthquake simulator that can simulate over a million years of California seismicity and feedback into the CyberShake calculation. Over the years, we've done a number of Pegasus optimizations, and I'll mention some in a minute. However, also, the applications themselves have been restructuring the workflow performance, tuning the MPI codes, and importing some of the codes to GPUs. So, Pegasus can be used by large scientific groups that have large problems, but we can also arm individual scientists with Pegasus. And, they can use it to run the, and leverage the power to open science grid. So, this is an example of a Ph.D. student, Ariella Gladstein from the University of Arizona. You see on the left, it's a graph of her execution on the open science grid since May of 2017. The different colors show you the different sites [inaudible] that she executed her calculations. And, the questions that she wanted answered with these calculations was how did humans spread across the world and what were the demographic events that led to where we are today and with diversity that we see? So, she analyzed the genetics of a number of the human population, and she used 342 workflows to do that. And, the workflows had 12 million jobs. They executed a cost for 40 of open science grid sites, and she consumed over seven million wall hours during her Ph.D. thesis. And, also, I believe that cross pollination between domains is highly beneficial. So, if you see, this is a graph of different releases of Pegasus since 2001. On the top of the graph, you see the developments, and, I didn't include other types of features that we included over the years for simplicity. But, on the top, you see the LIGO driven development that we conducted. On the bottom, you see the Southern California Earthquake Center driven development. So, for LIGO, we've looked at various data transfer monitoring tools. We implemented data cleanup and other algorithms, but SCEC, and I remind you that LIGO's workflows consist of single core jobs. So, they're really high throughput computing type of computations. So, SCEC which has a mix of MPI codes and high throughput computing computations were developed algorithms that allowed them to run these high throughput computing calculations on HPC systems. And, the reason for that is, two reasons for that. One is that it gives them a good turnaround time to solution to use high performance computing resources. And, also, there is a lot of data that's being transferred between, within a workflow between the HPC and HTC components so that it's often beneficial to collocate these two types of computations within an HPC system. So, with SCEC, we've developed job test clustering algorithms that allow us to put tasks, small tasks into larger holes that are easier to manage. We also developed a specialized workflow execution engine which is shipped with the part of the workflow to the HPC cluster, and Pegasus manages the execution of this subworkflow via MPI using master-worker paradigm. And, what's important about these developments is that if you look at 2015 when, suddenly, LIGO was moving was calculating not only on their own resources which are [inaudible] computing resources based on HTCondor and wanted to go to, in addition to open science grid, they wanted to go to XSEDE and Blue Water resources. They needed the capabilities that we've been developing since 2005 for SCEC. So, the capability of taking that high throughput computing workflows and putting them on the HPC environment. So, you can see that there's a lot of benefit of these application basically cross pollinating of ideas. So, but, however, this also adds a lot of complexity to the software, and that's something that we need to manage as well. So, it also takes funding to support software such as Pegasus, so this graph shows us the funding that's related to Pegasus that my group obtained over the years. In blue, you see the core funding out of OAC. And so, first, we started with SDCI program, and so, subsequently were funded by SSI. We also received funding for research for workflow design which is in orange on your screen, monitoring research provisioning, and more recently, for data integrity and in situ workflows. We also have been receiving application funding, in green. So, this is the funding that applications such as SCEC and LIGO has given us to help them develop workflows and optimize the workflow system to their particular needs. And, finally, we also have, in 2012, started receiving funding from infrastructures such as open science grid and XSEDE to provide, and in this case, we really provide expertise, not only in Pegasus, but in general for in workflow management systems and how do you manage workflows on the various resources. And, throughout, we also have received funding from NIH and DARPA, mostly to use the technologies in their environments. The green diamond that you see on the screen are basically our financial [inaudible]. So, these were the times that I couldn't sleep at night, and I was trying to figure out how I can sustain my group, and if I can't, who do I need to layoff. And, this is a very serious concern because it's very hard, once you layoff somebody, and even if your funding comes in a month later, it's very hard to bring that person back. So, that expertise, that knowledge is gone at that point. But, luckily, we had, were able to overcome these challenges of failures. So, in summary, some of my observations are, so I talked about the three things that need to intertwine, computer science, engagement with multiple domains, and the reuse of the existing CI when you do the development. I also added two more things because of the three of the elements to the left are not possible without the strong team and good collaborations. And, also, obviously, sustained funding. So, if we look ahead at the application trends. So, I think the applications are becoming much more complex, and to use the graph that you see is an application from the station neutron source, and basically, now the workflows are starting to include instruments from the beginning in the workflow process where you might want to look at the instruments data coming off and being analyzed in flight and compare the simulations. And, based on the observations, you may want to, then, feedback guide the instrument in some different fashion. And, you can imagine this type of modality will be also important in multi-messenger astronomy where you have some events happening at the detector and then you want to also notify other telescopes to retarget the areas that they're looking at. Also, so, if the complexity's going, obviously, you want to have pass the time to solution to enable these type of execution. And, also, at the same time, you have more individual researchers that are really in need of significant cyber infrastructure. So, just like Ariella's example using the open science grid for her research, there are many, many people out there that need the type of capability. So, we need to have, obviously, intuitive work for composition interfaces, better monitoring or handling some assisted debugging. But, also, I think the big question is how do you reach out to these people that don't know you're out there? So, how do you really disseminate this information and make sure that people have access to this cyber infrastructure that's available for them to use? And so, we can potentially leverage existing engagement. So, for example, I know that campus efforts or maybe the efforts that OSG and XSEDE have in that area. We may also partner with education outreach that's being conducted at instruments or experimental facilities. And then, finally, if we look ahead at what's going on in distributed systems or in systems in general that users had access to for growing cyber infrastructure, I think there's a growing demand for automation. So, the systems have become more complex. So, if we look at the HPC systems, they have become much more heterogenous. They have very specialized data storage that's available on them. They are also increasingly faulty. Distributed systems also have new capabilities. They have, for example, software defined networks that you can program the way you want to. You have various specialized data storage such as DPNs that you may want to use during your computations. Cloud's also a new platform that scientists have been using, although they've been around, and we explored their use back in 2008, I think they're becoming more important. However, they are very heterogenous and also, they can be very costly. So, really, resource management is key to harnessing the power of these type of infrastructures, and you may want to do it under different constraints of time to solution, budget. You need to deal with the faulty environment where you need to do cellular detection and attribution. You also need to deal with this large and heterogenous storage hierarchy. So, for example, you have the memory, the [inaudible] that I'm not coming online in different high throughput computing systems and can reduce the time to solution when you're running workflows in such environments. You have the file systems, data transfer nodes. All the stuff needs to be orchestrated and managed in an efficient way. And, obviously, in cloud forms you need to make sure that you manage the resources, and when they're no longer used, they get, they're torn down and so forth. And, the other thing that's kind of the big elephant in the room is that industry has been moving at the very rapid pace in the area of big data technologies and machine learning solutions. And so, I think we need to really keep track and try to harness these technologies for the benefit of science and to incorporate them into our software stacks. For example, I think a lot of work that's done in machine learning can be applied to fault detection and attribution in distributed environment. So, I think really leveraging this technology can help us solve problems that before seemed very daunting. And then, finally, I just wanted to leave you with some other examples of use of Pegasus in science, and I would like to thank my team and the collaborators and the funders that have supported us over the years. We also, as you can see, we have a very collaborative group, and we like working with a number of different scientific domains and other colleagues in computer science. So, I look forward to working with some of you in the future. Thank you.