



Research Highlight

e-Asthma Tracker

by Eun Hea Unsicker, Kelsee Stromberg, Flory Nkoy, Namita Mahtta, and Bryan Stone, Department of Pediatrics



Asthma is one of the most common chronic diseases among children. National guidelines recommend providing ongoing education for parents and children to learn about their own asthma so they can manage it better. In addition, it is recommended that parents and children learn their symptoms and triggers, and communicate with their primary care providers to catch deterioration of their asthma early enough to avoid serious exacerbations. Despite these recommendations, asthma control among pediatric population remains suboptimal. In 2016, more than 8 million children under the age of 18 had asthma, and over 50% of these children experienced an asthma exacerbation due to poorly controlled asthma.

Development of Asthma Tracker

With funding from the Agency for Healthcare Research and Quality (AHRQ), our team developed a paper-based Asthma Tracker to help families manage their asthma better at home. The Asthma Tracker utilizes the Asthma Control Test (ACT), a widely used asthma questionnaire, (below left figure), that was previously validated to assess a patients’ asthma control levels on a monthly basis. Our team received permission to modify the ACT, to add graphs (below right figure) with recommendations and to validate it to be used weekly (in children 12-18 years and for proxy use by parents of younger children 2-12 years of age) as part of Asthma Tracker. The Asthma Tracker graphs provide weekly asthma control scores (ranging from 5-25) displayed longitudinally, separated into three zones: Red (poorly controlled), Yellow (not well controlled), and Green (well controlled). Asthma Tracker provides recommendations depending on which zone they fall in each week to help patients avoid an exacerbation. Our studies suggest Asthma Tracker, its graphical output, and recommendations are relatively sensitive at detecting deterioration in asthma control well before it becomes serious, allowing the child’s healthcare provider to intervene early and prevent worsening loss of control.

To test, we enrolled patients and families admitted to Primary Children’s Hospital for asthma exacerbation to use Asthma Tracker for six months. The 210 participants were provided with a poster to graph their weekly asthma control scores, and a set of pre-paid postcards to complete and mail back to the research team weekly, allowing us to follow-up and evaluate their asthma control over time. When we received a postcard with a score falling in the Red or Yellow zones, our team contacted the families and encouraged them to follow-up with their primary care provider to get their asthma back in better control. The results were positive, with asthma admission rates lowered among patients using Asthma Tracker compared to patients who were not. However, the use of mail resulted in a delay in response, and in many instances by the time our team became aware of a Red or Yellow zone score and contacted that patient it was too late to intervene. In some cases, the patient was back in the hospital

During the PAST WEEK:

How much of the time did your asthma keep you from getting as much done at home, school, or work?

1. All of the time 2. Most of the time 3. Some of the time 4. A little of the time 5. None of the time

How often have you had shortness of breath?

1. More than once a day 2. Once a day 3. Three to six times 4. Once or twice 5. Not at all

How often did your asthma symptoms (wheezing, coughing, shortness of breath, chest tightness or pain) wake you up at night or earlier than usual in the morning?

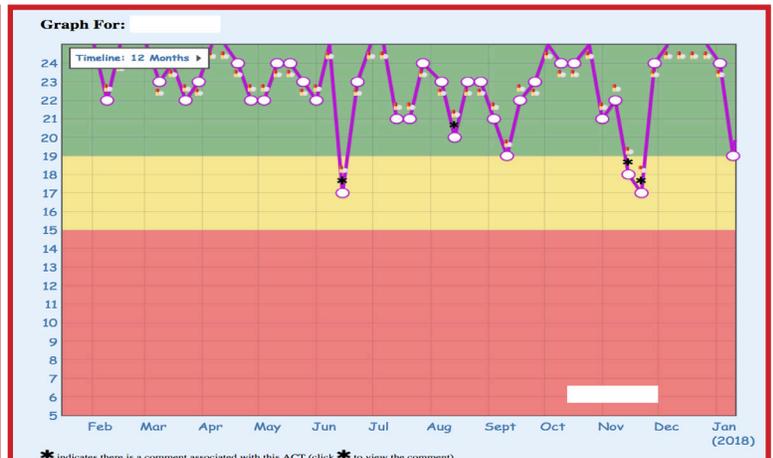
1. Four or more nights 2. Three nights 3. Two nights 4. One night 5. Not at all

How often have you used your quick-relief inhaler or nebulizer medication (such as albuterol, Ventolin®, Proventil®, or Maxair®)?

1. Three or more times per day 2. One or two times per day 3. Two or three times during the week 4. Once this week 5. Not at all

How would you rate your asthma control?

1. Not controlled at all 2. Poorly controlled 3. Somewhat controlled 4. Well controlled 5. Completely controlled



because their asthma had gotten worse after a Red score. We needed a system with real-time feedback for both families and their providers.

Development of electronic Asthma Tracker

To be able to provide real-time feedback to patients, we redesigned Asthma Tracker as an electronic web application: electronic-Asthma Tracker (e-AT). The e-AT is a web-based, mobile-web-friendly, self-management tool for patients and families. The e-AT system sends reminder messages when it is time to complete the e-AT surveys. If the score is low (in the Red zone or in the Yellow zone for two consecutive weeks), it sends alerts to patients, parents, and their primary care provider. For this project, we needed a database to store patient information, weekly asthma control scores, medication list, and other health-related information that was entered by the patients through the e-AT. Because our research and clinical care data contains sensitive information regarding patients and families, it was necessary to find an appropriate environment to store this data. The Center for High Performance Computing (CHPC) Protected Environment, with its security, met our needs and provided the necessary environment for this project. We collaborated with CHPC to get the e-AT project housed within their servers and they continue to assist us in maintaining the database where all the project data is stored.

With funding from AHRQ, we tested the e-AT by enrolling 219 asthma patients and families admitted to inpatient units at Primary Children's Hospital to use the e-AT for six months. The research team received real-time email alerts when a patients' scores were in the Red or Yellow zones. With this, we were able to call the families, find out what was happening, and help them to seek early medical care to prevent their asthma from getting worse.

e-AT project

After successful use of e-AT within inpatient populations, we received funding from the Patient Centered Outcomes Research Institute (PCORI) to implement the use of e-AT in eleven outpatient clinics. For this project, we collaborated with five parent stakeholders (parents of children with asthma), a Primary Care Provider Stakeholder, Utah Department of Health (UDOH), and other community stakeholders who assisted with research design and participated in clinic recruitment and the e-AT implementation processes. In these participating clinics, 325 participants were enrolled to use e-AT for one year. In addition, the clinic staff and primary care providers also used the e-AT to keep track of the patients' asthma control. Clinic users received real-time email alerts whenever a patients' scores dipped into the red or twice in the yellow zone. This provided an opportunity for clinicians to intervene early to prevent their patient's asthma from deteriorating further. After one year of e-AT use, the e-AT users had significant improvement in their asthma outcomes, including increased Quality of Life and asthma control, reduced missed school/work days, reduced

ED/hospital admission, and reduced oral corticosteroid use (a proxy for asthma exacerbations). All reductions were statistically significant and clinically important.

Other Projects

We have two additional projects on asthma and air pollution involving the use of the e-AT. We are collaborating with University of Utah's PRISMS project (<https://nursing.utah.edu/prisms>) to test indoor and wearable air pollution sensors. In a second project, we are working with the College of Pharmacy to determine how differences in specific asthma related genotypes affect the way asthma patients react to air pollutants and other environmental factors, as well as how the genotypes influence the way the body reacts to medications used to treat asthma. We use the e-AT to collect longitudinal symptom data from asthma patients as we compare their genotype and their exposure to air pollutants and other environmental triggers. With this information, we hope to understand how to provide personalized treatments that better fit the needs of the individual.

Future Direction

Recently, we submitted a proposal to PCORI to disseminate and implement the e-AT in more than 50 outpatient clinics throughout Utah. We will be working with University of Utah Health Clinics, Intermountain Healthcare Clinics, and rural community health centers through collaboration with UPIQ (Utah Pediatric Partnership to Improve Healthcare Quality). We anticipate 2000 patients to be enrolled to use the e-AT as a result of this; perhaps testing our partnership with CHPC with regard to data volume.

In addition, we are currently working to develop a tracking tool similar to e-AT for children with complex medical conditions (CMC) and their caregivers. The CMC Tracker will be a mobile-based tool, with the data housed within the CHPC servers. Similar to e-AT, the CMC Tracker will help children and parents track their health and will notify a healthcare provider when their status deteriorates for early intervention. We are still in the development phase of designing the mobile interface for patients and providers.

There is potential to develop other tracking tools for many chronic childhood health conditions, such as diabetes and congenital heart disease. These tracking tools can be developed not only for the pediatric population, but for all ages to promote self-management of chronic diseases, and health and wellness on a large scale. With our experience in the development and utilization of e-AT and CMC, we have been approached by researchers with plans to build tools similar to the e-AT for other chronic disease management. With our expanding list of research projects, our ongoing collaboration with CHPC will certainly be crucial in helping us to determine how to store and maintain the massive data generated from e-AT, CMC Tracker, and other potential tracker tools.

by *Martin Cuma, CHPC; Saveez Saffarian, Ipsita Saha, and Gleb Shtengel, Department of Physics*

Research computing support sometimes requires different twists and changes in direction. This was the case with enabling University of Utah Physics Professor Saveez Saffarian and his group to process iPALM (Interferometric photoactivation and localization microscopy) data on the University of Utah's Center for High Performance Computing (CHPC) resources. Previous to their use of iPALM, the Saffarian lab data processing and imaging was carried out at a smaller scale on individual PC analysis machines in their lab. This was not practical for the iPALM experimental technique due to both the amount of data and the processing time needed, leading the Saffarian group to look to CHPC for assistance. Working with CHPC was critical to allow a full transition of all of the Saffarian group's data storage and processing onto CHPC resources.

The Objective

The Saffarian group develops microscopy techniques that focus on live high-resolution imaging of essential molecular mechanisms critical for the function of retroviruses, such as HIV. As a part of this effort the group has been using the iPALM technique, which is capable of 3D imaging at a resolution of 10-20 nm. Specifically, iPALM has a greatly improved (about 2.5 times better) azimuthal resolution over the original PALM techniques that have been used by the group. At this resolution, protein molecules which constitute the virus can be observed in vivo, providing new insights into viral evolution and life cycle. However, with the increased resolution come larger data sets (10s of GB per image), which need to be processed by a custom software, developed at Howard Hughes Medical Institute (HHMI), to identify the objects of interest. As this analysis is time consuming, HHMI developed a cluster processing offload (i.e., a process to distribute the analysis workload to a high performance computing cluster instead of running it sequentially on one computer) via a Graphical User Interface (GUI) frontend.

Both the increased amount of data generated with the use of iPALM as well as the increased processing time associated with the new analysis software made the group realize that they needed to look beyond using their local resources to both store and process the data, and this prompted Professor Saffarian to contact CHPC to discuss the use of CHPC resources. After an initial meeting with Dr. Anita Orendt, the Professor met with CHPC Scientific Consultant and ACI-REF Dr. Martin Cuma, who, among other things, maintains Interactive Data Language (IDL), which is the language used for the iPALM data processing software. Getting the iPALM processing software to work was not a simple task, as described further below.

The Solution

As mentioned above, the iPALM analysis program, called

PeakSelector, is written in IDL, a programming language used for data analysis. Martin thought that he would install the program, interface it with the cluster scheduler and be done. However, using the IDL runtime that was already installed on our Linux clusters resulted in a segmentation fault, no matter what IDL version was used. Doing the same steps on a Windows system worked fine. As CHPC had been exploring containers at the same time, Martin tried to create a Singularity Windows container using the WINE emulator on Linux. This is quite atypical but it worked, although it ended up not being very stable and user friendly. Therefore, this lead Martin back to exploring the Linux error. As this error was not known to the HHMI developer, who was using the program in a Linux environment, Martin reached out to the IDL support which promptly suspected a known bug in a function that generates some of the GUI widgets and suggested that the GUI be rewritten. However, as the GUI is quite complex and the HHMI developer was focusing on other projects, rewriting the GUI would have resulted in a significant delay in the iPALM deployment.

The mystery remained why Peakselector works at HHMI; multiple calls with the developer did not provide any additional information as to the source of the differences. A few months later, Professor Saffarian and his graduate student, Ipsita Saha, were planning a trip to HHMI in Maryland to work with the iPALM developers on learning the technique, and invited Martin to join them for a few days. This turned out to be very productive on several fronts, as described below.

First, we saw with our own eyes that the program worked at HHMI, and by analyzing the software and hardware being used it was found that the Linux crash was video related. After trying many different systems it was determined that the program works on a handful of Nvidia video cards. Once back home, we tried a number of existing machines with the Nvidia video cards that we knew worked at HHMI, with mixed success. The crashing mystery continued. To extend the pool of machines that we had available for troubleshooting, we contacted the University Bookstore and spent a couple of hours with their extremely accommodating staff to try a wide range of computers, and finally found the magic solution – an Nvidia video card paired with ultra-high-definition (higher than 1080p) resolution display.

In addition to determining the source of the crash, the remainder of the stay at HHMI was spent getting to know the HHMI cluster setup, and implementing the HHMI Peakselector cluster offload at CHPC. Peakselector divides that analysis into hundreds of independent serial calculations each of which is submitted to a cluster. HHMI allows node sharing, but CHPC

does not, so, apart from modifying the submission system from LSF scheduler to SLURM, we also had to come up with a strategy to aggregate the individual calculations into groups to fill up all the cores on CHPC cluster nodes. Compared to the GUI crash troubleshooting, this was the easy part.

Once we had a stable platform on which to run the Peakselector, as well as the strategy to run the calculations on CHPC resources, the rest of the onboarding was quite standard. Martin helped Professor Saffarian and Ipsita set up the FastX remote desktop program on their local computers. He also installed and created the module file for Peakselector on CHPC's Linux systems and taught the Saffarian group how to run Peakselector and how to monitor batch jobs. Without the transition to CHPC resources, the processing of the iPALM data on the existing Saffarian group desktops would have been slow if not impossible. In addition, without the technical assistance of ACI-REF Martin Cuma, the group would not have been able to make this transition.

With the move to using CHPC resources for the analysis also came the need to purchase storage (CHPC's group space offering) to address the larger amounts of data that the group was now generating. In addition, the transfer of the data from the microscope to this storage needed to be explored. Therefore, along with engaging Martin, we also worked with the the CHPC networking team to explore the network connection options in the new lab space that will be hosting the iPALM microscope, and worked with campus networking to arrange for a 10Gb/s connection (originally set for a 1GB/s connection) from the instrument to the campus backbone. This connection allowed for the timely transfer of the images from the microscope to the newly purchased CHPC group space for storage and processing.

The Result

The newly installed iPALM software was used to finish the analysis of preliminary iPALM data collected in summer of 2017 and resulted in manuscript on the development of iPALM microscopy, "Correlative iPALM and SEM resolves virus cavity and Gag lattice defects in HIV virions," which is under review. Figure 1 shows the clear resolution advantage of iPALM versus the microscopy technique previous used to study these type of systems.

The Saffarian Lab was awarded an NIH RO1 award in summer of 2018, and their initial work with iPALM microscopy on purified HIV virions and cellular pathways was an essential component in obtaining this award.

From CHPC's standpoint the main take home message of this exercise was that things are not always clear cut and that troubleshooting and finding alternatives is a part of our support job. In addition, the project was a learning experience for Martin as he got the chance to explore different alternatives for IDL runtime deployment and got to see how research is supported and done at one of nation's leading private research institutes (the HHMI).

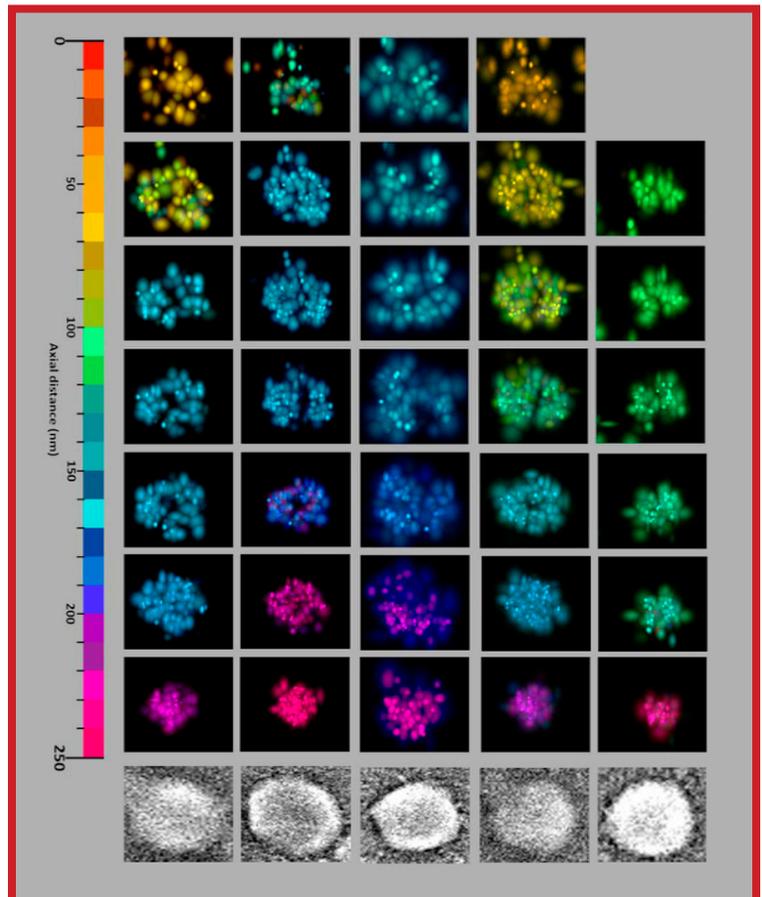


Figure 1. 3D visualization of HIV group-specific antigen (gag) protein molecules inside single virions

This figure shows five virus like particles (VLPs), one in each column, imaged with Scanning Electron Microscopy (SEM) – bottom, and iPALM. The six iPALM image rows represent different view depths providing 3D resolution of the virions' gag protein dendra. The diameter of the virions based on the SEM images is ~ 200 nm. The iPALM image processing was performed with the Peakselector program. Note the higher resolution of iPALM as compared to the SEM.

Funding Sources

This study was supported by NIH R01GM125444 to Saffarian. Martin Cuma was supported in part by a grant from the National Science Foundation, Award #1341935, Advanced Cyberinfrastructure - Research and Educational Facilitation: Campus-Based Computational Research Support.

Growing Notchpeak Compute Cluster

CHPC recently added eleven additional 32-core general nodes to notchpeak. Nine of these nodes, have 192GB of memory while the final two have 768G (notch044-045). This takes us to a total of 26 general nodes. In addition, there are three general GPU nodes on Notchpeak, each with three V100 GPU cards.

by Martin Cuma, CHPC

Software is becoming increasingly complex and more difficult to build and keep up to date. Many software publishers have chosen to package their programs into containers and distribute them via container repositories. Docker, the most popular container framework, has its own repository Docker Hub, which has become a *de facto* standard repository for application containers. Running a program through a container obtained from Docker Hub is becoming common as it gives a controlled way for the developers to provide the application, and an easy way for the users to get it.

Using the Docker program to launch a Docker Hub container is not allowed at CHPC due to security concerns. However, there is a fully user space based program called udocker which offers a functionality similar to Docker, which CHPC recently installed. The advantage of udocker over other tools with similar capability (Singularity, Charliecloud) is that it uses very similar commands to those from a Docker and it is a relatively simple implementation.

To use udocker, one first needs to load the module with `module load udocker`. To run a command from Docker Hub container, as a user inside the container, execute:

```
udocker run --user=$USER --bindhome
--volume=/scratch ubuntu /bin/bash
```

where `ubuntu` is the name of the Docker Hub container. `--bindhome --volume=/scratch` make the home and scratch file systems visible inside the container. This will download the `ubuntu` container from the Docker Hub, save it to user's home directory, and start the bash shell inside the container.

As most of us don't just want to run bash in a stock Ubuntu OS, we will now discuss a more practical container usage case. For this example we will work with Biocontainers, a repository of bioinformatics containers within Docker Hub that has hundreds of bioinformatics tools available. To run the blastp tool execute:

```
udocker run --user=$USER --novol /data
--novol /config biocontainers/blast blastp
-help
```

In this command, we use the `--novol` flags as a workaround to ignore two directories which are present as mount points in the container but are not on our systems. We just ran the help for the blastp tool, but, this demonstrates the functionality.

For more details on udocker see <https://www.chpc.utah.edu/documentation/software/udocker.php>; for additional information on other container tools at CHPC see <https://www.chpc.utah.edu/documentation/software/containers.php>.

CHPC Fall 2018 Presentation Schedule

See <https://www.chpc.utah.edu/presentations/> for details

August 23	Overview of HPC & CHPC	October 18	Introduction to Python, Part 2
August 28	Modules Basics	October 23	Numpy, Part 1 (Python, Part 3)
August 30	Slurm Basics	October 25	Numpy, Part 2 (Python, Part 4)
September 5-6	XSEDE HPC Monthly Workshop – Big Data	October 30	Introduction to Programming with OpenMP
September 11	Introduction to Linux, Part 1	November 1	Hybrid MPI-OpenMP Programming
September 13	Introduction to Linux, Part 2	November 6	XSEDE HPC Monthly Workshop – OpenACC
September 18	Introduction to Linux, Part 3	November 8	Introduction to Debugging
September 20	Introduction to Parallel Computing	November 13	Introduction to R
September 25	Introduction to Linux, Part 4	November 20	Introduction to Profiling
September 27	Introduction to Programming with MPI	November 27	National Compute Resources
October 2-3	XSEDE HPC Monthly Workshop – MPI	November 29	Introduction to Containers
October 4	GPU Programming	December 4	Introduction to I/O at CHPC
October 16	Introduction to Python, Part 1	December 6	Overview of the Protected Environment

High performance computing (HPC) has historically required the use of a terminal for UNIX commands input, which can be a hurdle for new users. Among the recent attempts to make access to HPC easier is the use of web portals. Many CHPC users are familiar with the FastX remote desktop application which can be run from a web browser. This approach frees one from having a terminal client program on his/her desktop, but, still necessitates some knowledge of the underlying infrastructure, in our case the Linux operation system commands and SLURM scheduler.

Open OnDemand (<http://openondemand.org/>) is a complete web portal which abstracts the Linux and scheduler commands through a web browser interface. CHPC runs a pilot instance at <http://ondemand.chpc.utah.edu>, which can be accessed with CHPC user name and password. The portal's main menu allows one to browse files (in Files menu tab), monitor and submit jobs (in Jobs menu, with Active Jobs and Job Composer items), launch a terminal on cluster interactive nodes (with Clusters menu item), and run interactive jobs on cluster compute nodes (with Interactive Apps menu item).

While the Files browsing and terminal access replicate what one could achieve with FastX using Linux desktop file managers and terminals, there are unique options in the Jobs and Interactive Apps, which don't require knowledge of the appropriate terminal commands. The Active Jobs list cluster jobs and their details. The Job Composer allows one to submit a job on a cluster, either from an existing job script, or from a template. We have populated the job templates with ones for a number of popular applications. One can choose the template, edit the job script by pressing the Edit Files button, add your input files and then submit the job by pressing the green Play button.

The latest feature, and the one that we are the most excited about, is the ability to launch interactive applications on the clusters. These are listed in the Interactive Apps menu tab. For our pilot, we have set up interactive desktop jobs on all CHPC clusters, and for four explicit applications on the Lonepeak cluster: Ansys, Matlab, Jupyter Notebook and RStudio Server. We will be adding support for other clusters and applications based on user demand.

As an example we will now show how to run an Ansys CFX calculation. We select Interactive Apps – Ansys Workbench on Lonepeak menu item to obtain the job launch options (Figure 1) and adjust them to our needs (e.g., putting owner-guest/lonepeak-guest account/partition as the default general partition is busy). After pushing the Submit button, we get the job status view; once the job starts we

see the "Launch noVNC in New Tab" button. This signals that the application is ready to be launched. Pushing this button opens a new browser window with Ansys Workbench. There we import a CFX input file, right click on the "Solution" workbench item to bring up the Run menu, and hit "Start Run" to run the simulation. Run progress is then interactively displayed in the Solver Manager window. This provides an user with the same experience as if run on a local workstation, but, instead, using a dedicated many-core cluster compute node, or, in case of the Ansys workbench, even multiple nodes.

We have recently added a new documentation page on the use of the Open OnDemand, <https://www.chpc.utah.edu/documentation/software/ondemand.php>. On this page users can find additional details on the use of the web portal. We encourage users to give it a try and welcome any feedback with respect to usability, as well as cluster and application support.

The screenshot shows the 'ANSYS Workbench on Lonepeak' settings window. On the left is a sidebar menu with categories: Interactive Apps, Desktops, GUIs, and Servers. Under Desktops, 'ANSYS Workbench on Lonepeak' is selected. The main content area contains the following settings:

- ANSYS Workbench version:** 18.2 (dropdown menu)
- Number of tasks:** 20 (input field)
- Number of hours:** 1 (input field)
- Account:** owner-guest (input field)
- Partition:** lonepeak-guest (dropdown menu)
- Node type:** any node (dropdown menu)
- Additional options:**
 - I would like to receive an email when the session starts

A blue 'Launch' button is located at the bottom right of the settings area.

Figure 1. Ansys Workbench interactive app settings window

On the CHPC computer clusters, we use the module command to load and unload software packages in your environment, in order to manage environment settings for those packages. The full module system is documented on our web site, at <https://www.chpc.utah.edu/documentation/software/modules.php>. In April 2018, Martin Cuma updated our modules software and the way our software modules are organized, producing a cleaner software environment with fewer conflicts between modules.

Several more visible outcomes of these changes are:

- Some deprecated modules have been hidden. You can still load these modules, however, and can find them with the "module --show_hidden spider module_name" command.
- Only one compiler module may now be loaded at a time. For example, you cannot load multiple versions of the gcc compiler, or simultaneously load both gcc and intel compilers. With this change, we have removed the compiler tag in the module versions, in particular for MPI distributions and some libraries such as FFTW and NetCDF.

We also about 90 new or updated software packages installed since January 2018

Container Management	Bioinformatics and Genomics	Other Applications	Machine learning
charliecloud, 0.2.4 (Updated)	bamtools, 2.5.1 (New)	ansys, 18.2 (Updated)	caffe, 1.0.0.singularity (New)
singularity, 2.5.0 (Updated)	bioBakery, 1.6 (New)	apbs, 1.5 (New)	caffe2, 0.8.1.singularity (New)
udocker, 1.1.1 (New)	bwa, 040218 (Updated)	augustus, 3.3 (New)	opencv, 3.4.1 (Updated)
Data visualization	dfast, 1.0.5 (New)	ciao, 4.10 (New)	tensorflow, 1.1.0.gpu (Updated)
diamond, 0.9.22.123 (New)	discover-denovo, 52488 (New)	csd, 2018 (Updated)	Compilers and Languages
idl, 8.7 (Updated)	epacts, 3.3.0.singularity (New)	gaussian16, B01 (Updated)	anaconda, 5.1.0 (Updated)
ovito, 2.9.0 (New)	gamola2, 2.0.12 (New)	interconnect, 7.5.1466 (Updated)	anaconda3, 5.1.0 (Updated)
Workflow management	jellyfish, 2.2.9 (New)	lumerical, 8.19.1466 (Updated)	cuda, 9.0.176 (Updated)
snakemake, 4.0.0 (New)	mirdeep-star, 37 (New)	miniconda3, 4.4.10 (New)	gcc, 4.8.5 (Updated)
Statistics and Graphics	multiqc, 1.5 (Updated)	nco, 4.7.4 (Updated)	gcc, 8.1.0 (Updated)
mropen, 3.4.3 (New)	peddy, 0.3.4 (New)	peakselector, jun18 (Updated)	intel, 2018.1.163 (Updated)
qt, 5.11.1 (Updated)	plastid, 0.4.8 (New)	visit, 3.0.0b (Updated)	intelpy, 2018.2.037.py2 (Updated)
R, 3.4.4 (Updated)	plink, 1.90 (Updated)	vmd, 1.9.3 (Updated)	intelpy, 2018.2.037.py3 (Updated)
RStudio, 1.1.442 (Updated)	qiime, 2.2018.4 (Updated)	wdfs, 1.4.2 (New)	jdk, 10.0.1 (Updated)
Simulation	rnafold, 2.4.3 (New)	Utilities	llvm, 5.0.2 (Updated)
freud, 0.6.1 (New)	salmon, 0.9.1 (New)	cfitsio, 3.410 (New)	mathematica, 11.3.0 (Updated)
Software Development	sambamba, 040218 (New)	emacs, 25.3.1 (Updated)	matlab, R2018a (Updated)
cmake, 3.11.2 (Updated)	SEQLinkage, 1.0.0 (New)	fpsync, 1.0.0 (New)	pgi, 18.1 (Updated)
googletest, 05252016 (New)	seqtk, 040218 (New)	gawk, 4.2.1 (Updated)	python, 2.7.11 (Updated)
Performance profiling	shortstack, 3.8.5 (New)	gdc-client, 1.3.0 (New)	python, 3.6.3 (Updated)
advisorxe, 2018.1.1 (Updated)	tassel, 5 (New)	rclone, 1.41 (Updated)	vtune, 2018.1.0 (Updated)
inspectorxe, 2018.1.1 (Updated)	trinity, 2.6.6 (Updated)	root, 6.12.04 (Updated)	Photogrammetric Processing
itac, 2018.1.017 (Updated)	vcftools, 0.1.15-6 (Updated)	slurmjoblogquery, 1.0 (New)	photoscan, 1.4.2 (Updated)
	vsearch, 040218 (New)	vim, 8.1 (New)	

Recent Change -- Retirement of /scratch/lonepeak/serial

At the end of August, we changed /scratch/lonepeak/serial read only as the first step to retiring it from general use. Please watch for an announcement to all users with additional details.

Please make changes in any batch scripts that make use of /scratch/lonepeak/serial. Alternates to /scratch/lonepeak/serial are: /scratch/kingspeak/serial (175 TB capacity) and /scratch/general/lustre (700 TB).

Please remember that CHPC has a script in place that deletes files from scratch file systems if they have not been accessed in 60 days.

The University of Utah
University Information Technology
Center for High Performance Computing
155 South 1452 East, Room 405
SALT LAKE CITY, UT 84112-0190

Thank You for Using CHPC Resources!

Welcome to CHPC News!

If you would like to be added to our mailing list, please provide the following information and send via the CHPC contact methods listed below:

Name:

Phone:

Email:

**Department
or Affiliation:**

**Address:
(U Campus
or U.S. Mail)**

PLEASE ACKNOWLEDGE THE USE OF CHPC RESOURCES

Please help CHPC continue providing access to cutting edge equipment by acknowledging the use of CHPC resources, both CHPC computer time or staff resources, in technical reports, publications, presentations, and dissertations.

An example acknowledgement:

"A grant of computer time from the Center for High Performance Computing is gratefully acknowledged."

If you make use of the CHPC protected environment, please also acknowledge the NIH shared instrumentation grant:

"The computational resources used were partially funded by the NIH Shared Instrumentation Grant 1S10OD021644-01A1."

Please submit copies or citations of dissertations, reports, pre-prints, and reprints in which the CHPC is acknowledged in one of the following ways:

Electronic responses

Email: helpdesk@chpc.utah.edu

Fax: (801)-585-5366

Paper responses

U.S. Mail: 155 South 1452 East, Rm 405
Salt Lake City, UT 84112-0190

U Campus Mail: INSCC 405