When STAR meets the Clouds – Virtualization & Cloud Computing Experiences

J Lauret\textsuperscript{1}, M Walker\textsuperscript{2}, S Goasguen\textsuperscript{3}
\textsuperscript{1}Brookhaven National Laboratory, Upton, NY, USA
\textsuperscript{2}Massachusetts Institute of Technology (MIT), Cambridge, MA, USA
\textsuperscript{3}School of Computing, Clemson University, Clemson, SC, USA
E-mail: jlauret@bnl.gov

L Stout\textsuperscript{1}, M Fenn\textsuperscript{3}, J Balewski\textsuperscript{2}, L Hajdu\textsuperscript{1}, K Keahey\textsuperscript{4}
\textsuperscript{1}Brookhaven National Laboratory, Upton, NY, USA
\textsuperscript{2}Massachusetts Institute of Technology (MIT), Cambridge, MA, USA
\textsuperscript{3}School of Computing, Clemson University, Clemson, SC, USA
\textsuperscript{4}Argonne National Laboratory, IL, USA

Abstract. In recent years, Cloud computing has become a very attractive paradigm and popular model for accessing distributed resources. The Cloud has emerged as the next big trend. The burst of platform and projects providing Cloud resources and interfaces at the very same time that Grid projects are entering a production phase in their life cycle has however raised the question of the best approach to handling distributed resources. Especially, are Cloud resources scaling at the levels shown by Grids? Are they performing at the same level? What is the overhead on the IT teams and infrastructure? Rather than seeing the two as orthogonal, the STAR experiment has viewed them as complimentary and has studied merging the best of the two worlds with Grid middleware providing the aggregation of both Cloud and traditional resources. Since its first use of Cloud resources on Amazon EC2 in 2008/2009 using a Nimbus/EC2 interface, the STAR software team has tested and experimented with many novel approaches: from a traditional, native EC2 approach to the Virtual Organization Cluster (VOC) at Clemson University and Condor/VM on the GLOW resources at the University of Wisconsin. The STAR team is also planning to run as part of the DOE/Magellan project. In this paper, we will present an overview of our findings from using truly opportunistic resources and scaling-out two orders of magnitude in both tests and practical usage.

1. Introduction
High Energy and Nuclear Physics experiments, facing daunting data and processing challenges, have envisioned at an early stage the use of widely distributed resources spread over a global cyber-infrastructure. Focused on providing a platform allowing aggregation of resources into a seamless pool, the era of Grid computing [1] was born, and since its inception it has shown its success in data processing and handling over vast distances and across many continents. Grid projects such as the Open Science Grid [2] matured and hardened the technology to provide national Grids. Whilst their successes are certain, limitations of Grids were obvious from the start: with heterogeneity at the heart of its philosophy and the plethora of environments, the exploitation of such global resources became rather challenging as the number of platforms, operating systems, sets of libraries and compiler versions exploded in diversity and flavors. The arrival of Cloud computing, however, has introduced a new reality. The infrastructure layer of Clouds, known as Infrastructure as a Service (IaaS), relies on virtualization to provide on-demand, elastic and pay per use access to resources. A virtual machine instantiated on a cloud can contain all the required components from a tailored and specific operating system, a specific middleware, and an experimental software stack pre-verified, controlled, and
approved by the Virtual Organization (VO) authoring the VM. Not only is the software provisioning complexity suddenly tremendously reduced, but the software validation is greatly simplified. In other words, the dream of harvesting unused resources is made possible by making all resources homogeneous via virtualization. From private commercial clouds to public national laboratory or university community clouds, the explosion of Cloud models raise some new interesting challenges, among them the lack of standards and the lack of understanding which mode best fits the VO’s already exploiting Grid resources. It is also noteworthy to state that while virtualization is a very attractive feature at the infrastructure layer of cloud computing, Clouds are not “silver bullets”: the problems of data movement, monitoring, accounting, security, workflow management and optimization -addressed by the major Grid projects- remain. This paper will present a few models tested by the RHIC/STAR experiment and describe experiences with using these models.

The Solenoidal Tracker At RHIC (STAR) experiment [3] is a Nuclear Physics experiment located at the Brookhaven National Laboratory (BNL) and part of the RHIC [4] program. At STAR, physicists and skilled specialists from across the world are working hard to understand the nature of the early universe and the tiniest building blocks of matter through the study of nuclear collisions at the highest energies achieved in the laboratory. To achieve this goal, the STAR experiment has developed an ambitious physics program now entering the Petascale data challenge. In its resource planning [5], the experiment showed that half of its user analysis as well as all of its simulation needs relied on outsourcing off the main BNL computing center. With the experiment now moving to analysis of large data sample, STAR enters an era where systematic uncertainties are predominant and this may raise needs for additional simulation for understanding the effect of the detector response, fine grain efficiency, and momentum corrections.

![Figure 1](STAR participation in Grid and Cloud projects. Testing of Cloud resources was as early as 2006 but real use started in 2008/2009.)

To face this challenge, STAR has had a long-standing participation in Grid co-laboratories programs. The timeline is represented on Figure 1. Since its early phase, STAR has been part of the Particle Physics Data Grid [6] consortium, the Trillium project and its next incarnation, the Open Science Grid. In parallel the STAR experiment Software and Computing team has also carried and maintained active participation in alternative distributed computing approaches ([7],[8]) with its first truly last minute opportunistic use of Cloud computing based resources in early 2009 ([9]). Certainly, such a timeline suggests that considering the inclusion of virtualization technologies in Grid projects was possible at a very early stage, even before Clouds became mature. In the next sections, we will show diverse models tested in recent years.

2. The Cloud models
Figure 2: STAR testing scale as a function of time. From the initial Nimbus test in 2006 to the Kestrel model in summer 2010, the scale of sustainable operation has grown by two orders of magnitude, indicating a maturation of the technology and approaches.

2.1. Tested Cloud & “virtualized Grids” models.
Cloud computing anatomy is treated in [10]. In this paper, we focus on how to start and manage jobs efficiently and scale to levels allowing supporting data production operations. In other words we concentrate on how to perform tasks usually in need of many individual jobs slots on a Cloud platform and how the models approached this problem.

Our five test models were the Amazon Web service model [10], the Nimbus/EC2 approach [11], the Virtual Organization Cluster (VOC) from Clemson University [12-13], the Condor/VM model [14] and finally, the Clemson/Kestrel job management model [15]. Some of those are not clouds in the sense of Amazon web services but rather appear as “virtualized Grids” (i.e. Grid sites provisioned via virtualization). Reference [18] reports in details the diverse models our findings are summarized in Table 1 and Figure 4. All models were tested at a particular point in time and hence the measured efficiency (defined as number of successful jobs over the total number of jobs submitted) reflects the level of maturity at the time. However, lessons are extracted from each exercise and the progression has led to an even more robust Cloud or virtualized Grid approach.

Table 1: Summary of the tested model, efficiencies and type of processing

| Model          | Efficiency | Scope and notes                                                                 |
|----------------|------------|---------------------------------------------------------------------------------|
| Amazon/EC2     | > 99%      | Simulation workflow, event generator: no input, some output. Medium instances used, a 5 MB/sec transfer was then adequate to transfer our results back to our Tier0 center. Model has a very competitive price – for simple workflows, a year long CPU @ 100 jobs saturation would be 79k$. |
| Nimbus/EC2     | ~ 85% on first try, 97% on 2nd try | Simulation workflow, event generator. The initial drop of efficiency was due to the RMS “inside” and its configuration. This approach does not remove scalability issues inherent to standard local or Grid job submission. Model was simple to use (appeared as another Grid site with OSG software stack inside). First REAL usage for Physics (see [8]) |
### Simulation workflow

| Model       | Effect | Details |
|-------------|--------|---------|
| Clemson/VOC | ~ 100% | We did not lose a single job, very transparent and entirely dynamic with no effort on VO side. As the previous model, contextualization remains a (time) overhead but the subscribe mechanism to an external batch system seem have brought enhanced scalability and consistency of resource use between virtualized and non-virtualized resources. Performance highly improved by caching the image on the host. |
| Condor/VM   | [80%, 85%] | Main finding was that the lifetime of the VM (reboot every 24 hours) would cause at least a 5% job loss (Monte Carlo simulation have some uncertainty on job’s runtime). No real job inside the VM (need cron or similar mechanism). Nearly no contextualization however. Local IP space was “local”. Data transfer mechanism through a common storage had to be handled separately. **First successful scale at 500 jobs. Analysis proof of principle.** |
| Clemson/Kestrel | ~ 90% | Full simulation workflow (event generation, detector response, track reconstruction). Clemson and CERN nodes (only a few) mixed run smoothly. Full Grid stack inside – use of myproxy and handled transfer back to Tier0 using Grid tools. NO batch system (Kestrel manages start/stop jobs via a jabber-like client – see section 2.3 for details). **First successful scale at 1000 jobs, results used for an international conference.** |

#### 2.2. Virtualization boundaries

![Virtualization Boundary](image)

**Figure 3:** Virtualization boundary may differ depending on the model used. (a) A virtual cluster model (b) virtualized worker nodes subscribes to a resource management pool and (c) virtual machines are raised by an external batch system (job slots are filled by VM instances).

In the tested models, we have encountered three approaches differentiated by where the boundary of virtualization resides as summarized in Figure 3. Most of our observations on limitations directly follow from the choice of models.

In (a), an entire virtual cluster is provisioned; one of the VM serves as a head node running a Globus gatekeeper (Globus toolkit 2.4) and a condor manager while the other VMs act as worker nodes.
Within this “virtual cluster” model and from an external user stand-point, the cluster appears just like another Grid resource on the Open Science Grid. Worker node VMs may have a Grid software stack (e.g. OSG Software stack) installed as part of the customization of the image during instantiation. A process called contextualization. Therefore, workflow integration is immediate with the virtualization components ensuring easy and validated software provisioning. This approach is used by the Nimbus/EC2 and to some extent, the Kestrel models. Note that all proxy and credentials are self-consistent in this case.

In (b), an external mechanism (i.e. modified Globus job manager) triggers the provisioning of VMs as demand for particular resources grow. The VMs run a local resource manager (e.g Condor, PBS/Maui As VMs appear in the RMS, the batch slots accept jobs and execute them within the VMs. This “on-demand” VM structure forms a secondary resource layer which may allocate and partition dynamically a physical cluster depending on the share of each VO. Unlike (a), the dynamic nature of the model is convenient as several images may start even within the same VO. This was the path taken by the Clemson VOC model. Also, in the VOC model, all VMs had their own IP, allowing transfer of files out of the resource. This largely eased integrations (i.e. we could access Web services, transfer files in and out using Grid services with no problems), but the case is not always scalable: to maintain both native and hosted OS, one needs twice the address space needed for a public IP worker node alone.

Finally, the approach taken in (c) is to use a standard batch approach and manage the instantiation of a virtual machine as a regular batch jobs which needs data staging (i.e the VM image) and executes a binary (i.e. call to the local hypervisor to start the VM). The management of resources is again consistent between the local and virtualized world, the same batch system keeps track of the resource allocation and the rules for scheduling may be unified. This approach is used by the Condor/VM model. Since this mechanism is a full match-making mechanism [17], a large number of VOs and VMs may be supported offering maximal flexibility. The downside however was that the workflow needed to be managed separately since in our case Condor does not manage it. Here, we also faced the use of a local IP space not allowing transferring out of the data. In such a case, a cumbersome mechanism involving an intermediate storage element at the site is used and yet another workflow element pulls the data out of the site as it is produced. Delegation of data transfer using Grid tools cannot be safely done as each VM is confined to a local network and cannot contact the remote myproxy server.

### 2.3. The Kestrel model

Since the Clemson/Kestrel model was extensively used and its results integrated in a full fledge scientific deliverable of the STAR collaboration, let us describe its main principles. The general architecture diagram is given on Figure 4.

---

**Figure 4:** Based on the eXtensible Message Passing Protocol (XMPP, see [8]), a broker communicates with the worker nodes (i.e Kestrel worker agent) possibly located at multiple sites. A jabber [8] client allows communicating with the system. The communication is asynchronous, allowing communication scalability to a large number of clients. As VM start, they register to the Kestrel resource pool.
The hypervisor used was KVM [24] and a full simulation workflow was planned for this study, requiring the full STAR software to be installed (over 2.5 Million lines of codes). To ease the workflow and allow for independence of each worker from the network, we made use of an embedded database snapshot (i.e. a compressed copy of the necessary database parameters were embedded in the VM, de-compressed on the fly, a database service started and the program execution would use the local DB). The total size impact was 0.5 GB additional space over a 5 GB deflated VM instance. Since KVM snapshotting was used (leveraging the lesson learned from the VOC model in ref. [13]), the operation did not add any significant overhead. The effort targeted a large simulation of proton-proton collision in support of the spin program. A total of 12 Billion PYTHIA [23] events were generated using this model, the largest sample of this kind ever produced. The accumulated usage totalled 400,000 CPU hours over 1,000 CPUs over the course of a month length. The final sample represented, at a typical 50 CPUs an individual user may claim at the Tier0 as usage, a year of science running for a student in STAR and a 25% resource expansion for STAR as a whole over the course of the running period. Science-wise, the resulting dataset represented four order of magnitude increases in number of events comparing to any analysis ever presented in STAR and allowed the STAR collaboration to un-ambiguously demonstrate good agreement between the data and the simulation [22]. Virtual machines were mostly located at Clemson University, to demonstrate the possibility of Kestrel to have far remote resources joining the pool we added a dozen virtual machines from CERN and Amazon EC2. Additional VMs could be started at Grid sites that have enabled virtualization or on other Cloud providers. Figure 6 shows the behaviour of the system under constant job feeding. As we can see, the good tracking of available resources and running jobs over an average of ~ 1,000 jobs (and peaks at 1,200) indicates a scalable and robust system of communication.

3. Conclusions
Cloud computing approaches have emerged and offer a very attractive alternative to Grids by implementing the value added component that is virtualization. However, many models are appearing all considering different approaches and use of virtualization. From pure Cloud models equipped with their own API and interfaces (e.g. Amazon/EC2) to the appearance of virtual clusters models (e.g. VOC) or resources managed by external batch systems, the convergence toward a unique vision is not yet crystalizing while each approach try to address particular scalability, instantiation, burden of...
contextualization or ease of use from the VO perspective. In this paper, we pointed out a few differences and limitations but also presented a more recent model, the Clemson/Kestrel model introducing an approach and vision not relying on standard batch system and using proven scalable technology (XMPP). The STAR experiment has made good use of this model and will continue to utilize resources from other Cloud models (especially the one from the Magellan project [19]) and help converge toward a unified vision.

4. Acknowledgements
This work was supported by the Nuclear Physics Divisions of the Office of Science of the U.S. DOE.

References
1. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. I. Foster, C. Kesselman, S. Tuecke. *International J. Supercomputer Applications*, 15(3), 2001
2. The Open Science Grid consortium, [http://www.opensciencegrid.org/](http://www.opensciencegrid.org/)
3. The STAR collaboration: J. Adams et al., Experimental and theoretical challenges in the search for the quark gluon plasma: The STAR collaboration’s critical assessment of the evidence from RHIC collisions. Nuclear Physics A, 757:102, 2005
4. The Relativistic heavy ion Collider. [http://www.bnl.gov/rhic/](http://www.bnl.gov/rhic/)
5. The STAR computing resource plan 2009, STAR Notes [CSN0474](http://www.sun.com/service/sungrid/brookhaven.pdf)
6. The Particle Data Grid (PDDG) Collaboratory Pilot, [http://www.ppdg.net/](http://www.ppdg.net/)
7. SunGrid and the STAR Experiment, [http://www.sun.com/service/sungrid/brookhaven.pdf](http://www.sun.com/service/sungrid/brookhaven.pdf) + Experience with on-demand physics simulations on the Sun Microsystems computing facility (SunGrid) at network.com, J. Lauret et al 2008 *J. Phys.: Conf. Ser.* 119 052024
8. Integrating X-Grid into the HENP distributed computing model. L Hajdu, A Kocoloski, J. Lauret and M. Miller, 2008 *J. Phys.: Conf. Ser.* 119 072018
9. Nimbus and Cloud Computing Meet STAR Production Demands, *HPC wire April 02 2009* + Clouds make way for STAR to shine, *iSGTW April 8th 2009* + Number Crunching Made Easy, Cloud computing is making high-end computing readily available to researchers in rich and poor nations alike, *Newsweek May 2nd 2009* + Computing for the RHIC experiments, *CHEP 2009 invited contribution.*
10. Cloud versus Cloud: The Blessing and Challenges of Cloud Computing for Science, Kate Keahey, *CHEP 2010 plenary contribution*Amazon Elastic computing Cloud, Amazon EC2, [http://aws.amazon.com/ec2/](http://aws.amazon.com/ec2/)
11. Nimbus, an OpenSource toolkit for enabling IaaS on clusters, [http://www.nimbusproject.org/](http://www.nimbusproject.org/)
12. M. A. Murphy and S. Goasguen “Virtual Organization Clusters: Self-Provisioned Clouds on the Grid submitted to Elsevier Journal of Future Generation Computer Systems special issue on Cloud Computing, (In Press)
13. Contextualization in Practice: The Clemson Experience, M. fenn, S. Goasguen, J. lauret, proceedings of 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, PoS(ACAT2010)027
14. The Condor project, a high throughput computing scheduler system and its Virtual Machine Applications.
15. Kestrel: an XMPP-based framework for many task computing applications, L Stour, M. Murphy, S. Goasgen, In Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (Portland, Oregon, November 16 - 16, 2009). MTAGS ’09. ACM, New York, NY, 1-6. DOI= [http://doi.acm.org/10.1145/1646468.1646479](http://doi.acm.org/10.1145/1646468.1646479)
16. MyProxy, an OpenSource software for managing X509 Public Key management Infrastructure (PKI)
17. Condor Matchmaking with ClasAds, [http://www.cs.wisc.edu/condor/manual/v6.1/2_3Condor_Matchmaking.html](http://www.cs.wisc.edu/condor/manual/v6.1/2_3Condor_Matchmaking.html)
18. *From Grid to Cloud, the STAR experience* – J. Lauret et al., SciDAC 2010 conference proceedings (to be published) – also available as *STAR Notes SN0524.*
19. Magellan Project, *Exploring CLOUD Computing for DOE’s Scientific Mission.*
20. Jabber Technology, http://www.cisco.com/web/about/ac49/ac0/ac1/ac258/JabberInc.html
21. The eXtensible Message Passing Prototol, http://xmpp.org/
22. Spin 2010 conference, Dijet in Polarized Proton-Poron collision at AGeV=200, Matthew Walker for the STAR collaboration (proceeding to be published).
23. The PYTHIA event generator, http://home.thep.lu.se/~torbjorn/Pythia.html
24. The Kernel based Virtual Machine (KVM) system, http://www.linux-kvm.org/page/Main_Page