WORKLOAD NOTES: contains email exchange with experts

Notes from Marco Guazzone:

Unfortunately, there is a very lack of real workload traces in the cloud  
computing research community.  
As far as I know, much of existing works on cloud systems use traces  
coming from grid systems (e.g., see [http://gwa.ewi.tudelft.nl/](http://gwa.ewi.tudelft.nl/" \t "_blank)) or  
legacy parallel systems (e.g.,  
[http://www.cs.huji.ac.il/labs/parallel/workload/](http://www.cs.huji.ac.il/labs/parallel/workload/" \t "_blank)).  
  
As you know, grid and parallel systems are a completely different story.  
So these traces  are not representative of actual cloud systems.  
Indeed, in my research I usually rely on workload generated by real  
applications.  
To do so, I use the following applications and load generators.  
  
Applications:  
- The Rice University Bidding System (RUBiS): [http://rubis.ow2.org/](http://rubis.ow2.org/" \t "_blank)  
- The Rice University Bulletin Board System (RUBBoS):  
[http://jmob.ow2.org/rubbos.html](http://jmob.ow2.org/rubbos.html" \t "_blank)  
- The Apache Olio project: [https://incubator.apache.org/olio/](https://incubator.apache.org/olio/" \t "_blank)  
- The Apache Cassandra project: [https://cassandra.apache.org/](https://cassandra.apache.org/" \t "_blank)  
  
Workload Generators:  
- The RAIN Workload Toolkit from University of Berkeley:  
[https://github.com/yungsters/rain-workload-toolkit](https://github.com/yungsters/rain-workload-toolkit" \t "_blank)  
- The Yahoo! Cloud Serving Benchmark:  
[https://github.com/brianfrankcooper/YCSB](https://github.com/brianfrankcooper/YCSB" \t "_blank)  
  
You can also find useful the CloudSuite benchmark:  
[http://parsa.epfl.ch/cloudsuite/cloudsuite.html](http://parsa.epfl.ch/cloudsuite/cloudsuite.html" \t "_blank)

========================================================

> Thanks a lot for your detailed reply. I have couple more questions.  
> Is there a way to only generate traces using RAIN toolkit without  
> actually running the workload? Looking at the RAIN paper I felt they  
> are trying to run the workload as well.  
>  
Nope :(  
  
But I've just remembered one paper about the workload characterization  
for private clouds:  
   [http://www.computer.org/csdl/trans/sc/preprint/06642026-abs.html](http://www.computer.org/csdl/trans/sc/preprint/06642026-abs.html" \t "_blank)  
(You can also download it at:  
[https://p2p.cs.ucsb.edu/research/tech\_reports/reports/2013-05.pdf](https://p2p.cs.ucsb.edu/research/tech_reports/reports/2013-05.pdf" \t "_blank))  
  
In addition to the model for synthetic generation described in the  
paper, you can access to the traces at:  
   [https://www.cs.ucsb.edu/~rich/workload/](https://www.cs.ucsb.edu/~rich/workload/" \t "_blank)  
  
If you program in C++, you can take a look the Boost library here:  
[http://www.boost.org/doc/libs/1\_57\_0/libs/math/doc/html/math\_toolkit/dist\_ref/dists/hyperexponential\_dist.html](http://www.boost.org/doc/libs/1_57_0/libs/math/doc/html/math_toolkit/dist_ref/dists/hyperexponential_dist.html" \t "_blank)  
for working with hyper-exponential distributions, and here:  
[https://github.com/sguazt/random/blob/feature/hyperexponential\_dist/include/boost/random/hyperexponential\_distribution.hpp](https://github.com/sguazt/random/blob/feature/hyperexponential_dist/include/boost/random/hyperexponential_distribution.hpp" \t "_blank)  
for randomly generating numbers according to the hyper-exponential (this  
is a work-in-progress; I need to fix some issue before it can be  
accepted in the Boost library; unfortunately, currently I have no spare  
time)  
  
Otherwise, it is rather straightforward implementation a random variate  
generator for the hyper-exponential distribution (e.g., see  
[http://www.columbia.edu/~mh2078/MCS04/MCS\_generate\_rv.pdf](http://www.columbia.edu/~mh2078/MCS04/MCS_generate_rv.pdf" \t "_blank)).  
  
So, if you like private clouds, you can try to play with such  
traces/model :)

========================================================