Computation Institute

Facility Virtualization Experience at MWT2

Lincoln Bryant

Computation and Enrico Fermi Institutes

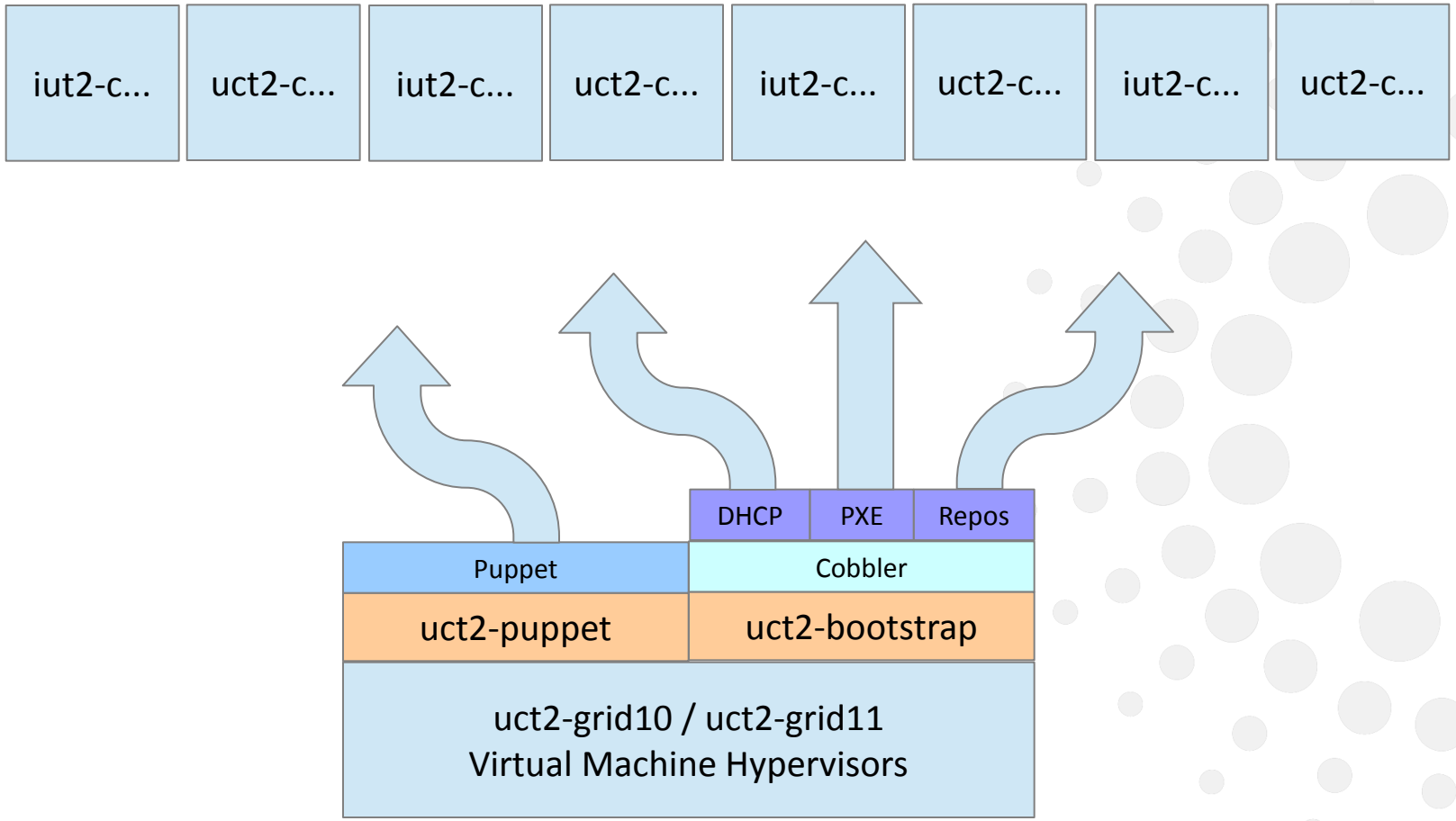
Introduction

- We were asked to set up a virtual Panda queue at Midwest Tier 2
- I'll speak about provisioning a virtual cluster and setting up the panda queue
- I'll show off some performance assessments of the queue.
- And finally I'll look at some future technologies, and how we can evolve this project into a cloud platform for MWT2.

What is the current infrastructure at MWT2?

- Cobbler (PXE images, repos, DHCP) and Puppet (post-install configuration).
- We have four hypervisor machines between IU and UC.
- Each hypervisor is a Dell R710 with 24 logical CPUs and 96 GB of RAM
- Our hypervisors run virtual head node services (gums, srm, mysql, condor, etc..)

MWT2 Node Provisioning Infrastructure



(* UIUC nodes provisioned a bit differently at the moment)

What sort of infrastructure do we need for a virtual queue?

- Two paradigms:
  - Go through the entire installation process for each VM and do post install configuration
  - “Bake in” everything we need into a VM image and deploy
- We could have chosen do the former, but the latter makes more sense for a cloud deployment. (Plus we get the image for free from BNL)
- Instead of potentially swamping our hypervisors providing head node services, we chose to refit our standard R410 worker node.
  - Given 24 logical cores and 48 GB of RAM, each hypervisor should be able to host 4 or 5 virtual workers with some room left over for the hypervisor itself
- We'll also need to set up a Panda queue, AutoPyFactory, and a Condor head node.

What tools do we need to build a virtual queue?

- Standard stuff like DHCP, DNS, etc.
  - Cobbler does a reasonable job at this and is already well integrated into our infrastructure.
- Some sort of virtualization layer.
  - BNL uses OpenStack. That seems like a good starting point.
    - On the other hand, OpenStack requires a little R&D.
  - We've been using libvirt + KVM for a while. Why not try that?
- Some sort of tool to configure the workers.
  - Puppet is useful for configuring nodes post-install, but it would be much nicer to just start out with our workers prefabbed.
  - Enter BoxGrinder.

Pros and Cons of BoxGrinder

- Pro:
  - Very nice tool for building virtual machines that work out of the box
  - Appliance definition supplied by BNL creates functional worker nodes with minimal configuration on our part.
- Con:
  - Can't easily leverage our existing Puppet infrastructure
  - Supports libvirt, but the plugin is still immature
  - RHEL-variants aren't supported. Need to use Fedora.
  - Or do we?

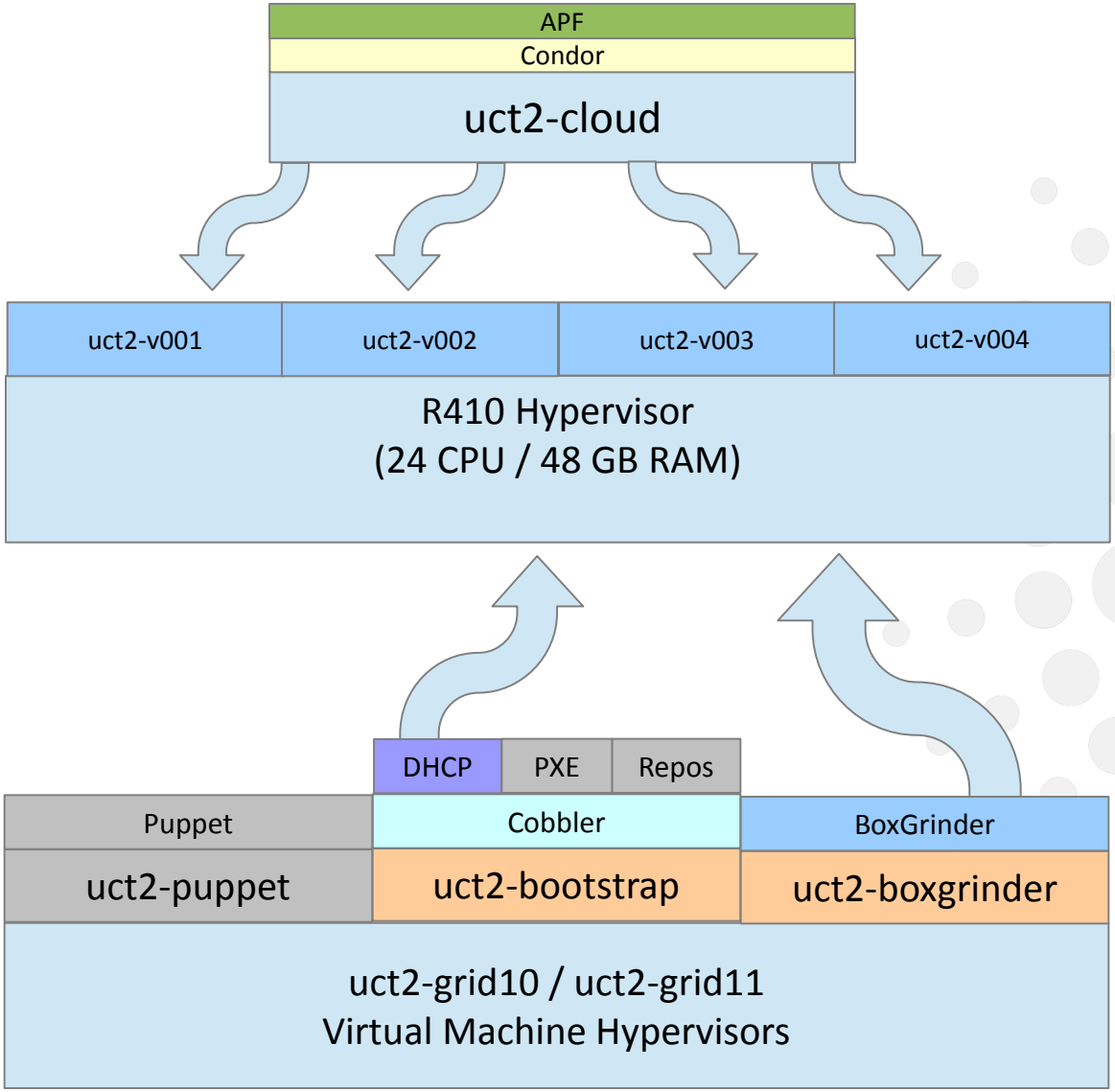
Boxgrinder: Trials & Tribulations

- We're an SL shop – why not try to build BoxGrinder for SL 6?
  - Surprisingly easy! Only needed to rebuild the appliance-tools RPM from Fedora 17
  - The rest of the dependencies were installed via Ruby Gems and YUM
- BoxGrinder's libvirt support is a bit rough around the edges
  - BoxGrinder makes incorrect assumptions about our network configuration.
  - Need to use a sed script to inject kernel console parameters.
  - VMs need to be re-imported by hand with virt-install and have their MAC addresses added to Cobbler's DHCP server
  - Clearly a hurdle if we want to scale up.

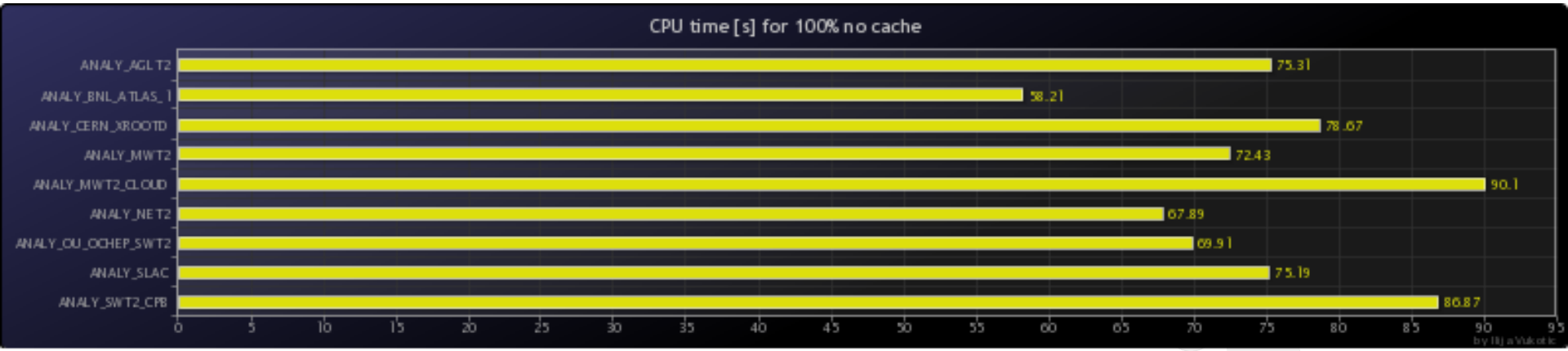
ANALY_MWT2_CLOUD Queue and Head Node Configuration

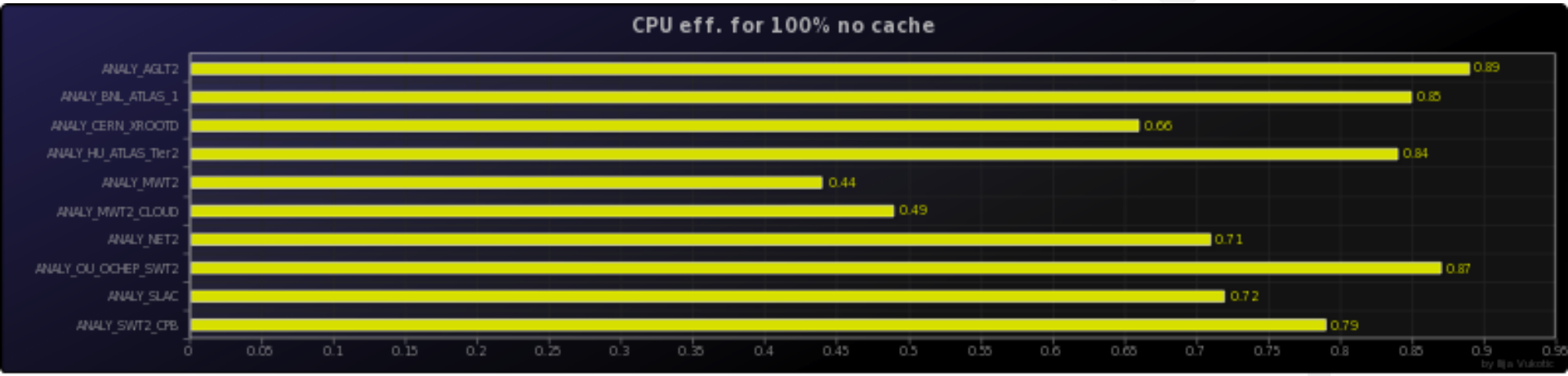
- Standard Condor head node with minor changes to BNL's configuration
- We were able to copy BNL_CLOUD as a template for ANALY_MWT2_CLOUD
- Small problem with AutoPyFactory installation.
  - As it turns out, a required RPM (python-simplejson) went missing in EPEL for SL5.
  - John and Jose found the required RPM and put it in the APF repo for us

ANALY_MWT2_CLOUD Infrastructure

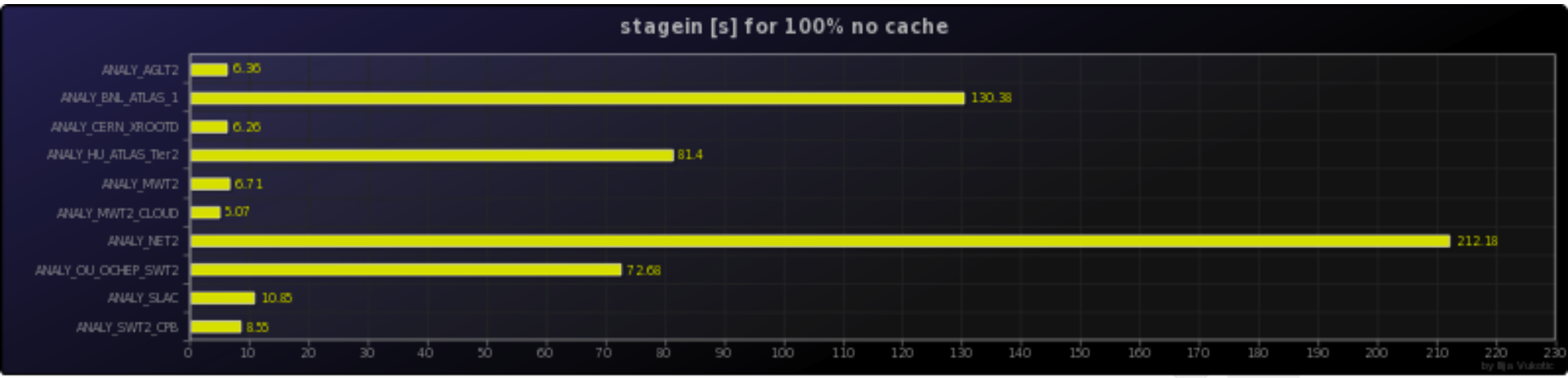


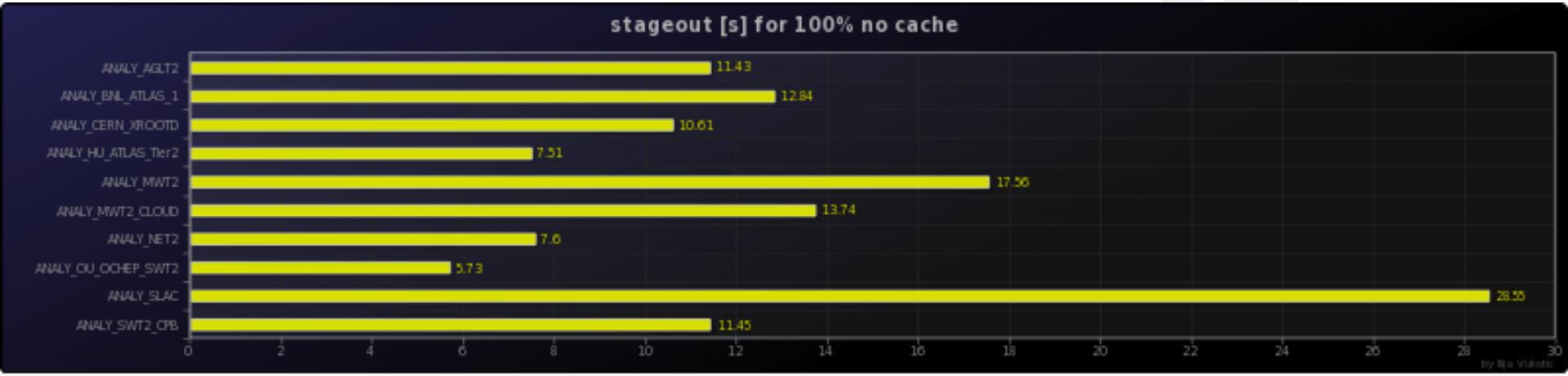
Benchmarking ANALY_MWT2_CLOUD





Benchmarking ANALY_MWT2_CLOUD





Future directions

- Looking forward, a dedicated virtualization platform is necessary to scale up.
  - OpenStack is a proven solution at BNL, so this seems the best place to start.
- What about even farther out? Perhaps Red Hat CloudForms?
  - Successor to Red Hat Satellite
  - Using Katello, integrates Puppet (config), Pulp (repos), Foreman (provisioning), etc.
  - Uses OpenStack for virtualization, should be able to import BoxGrinder images without hassle.
  - Open-source API, no vendor lock-ins!

The Future?


CLOUDFORMS™
by Red Hat® Cloud


KATELLO


openstack™
CLOUD SOFTWARE


FOREMAN


puppet
labs

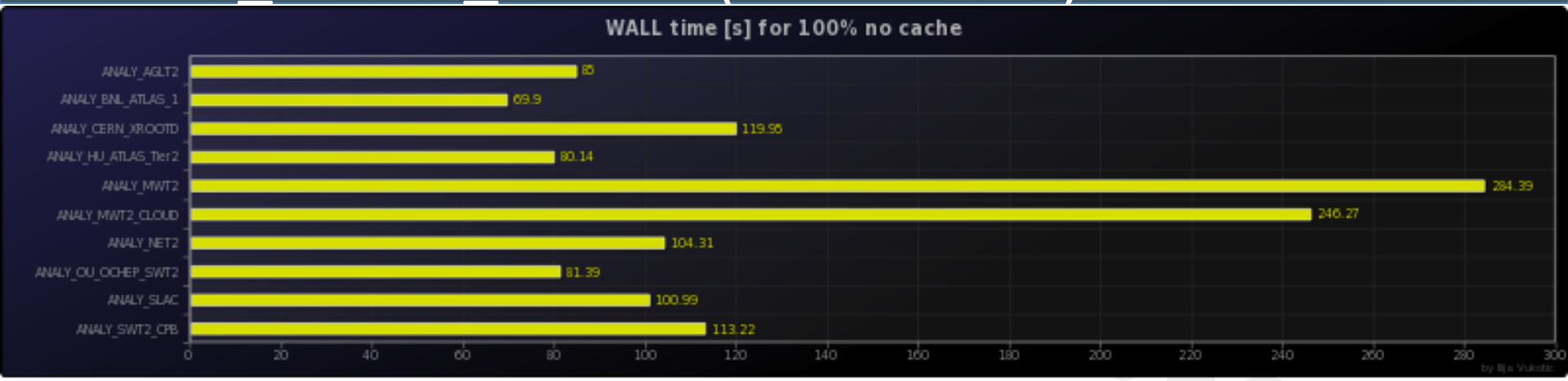

BoxGrinder
Build


pulp

Computation Institute

Thank you!

More Benchmarking for ANALY_MWT2_CLOUD (Bonus Slide)



Even More Benchmarking for ANALY_MWT2_CLOUD (Bonus Slide)