Making Openstack more Energy Efficient...

...a little story from your friends at ICCLab...

The challenge

- Energy continues to be a primary concern in very large clouds and data centres
- It will impact smaller deployments eventually...
  - ...through policy/regulation or making IT manager responsible for energy budget...
- Cloud stacks need to be energy aware...
  - ...to deliver energy efficient IT
- This is not a new idea...
  - Eucalyptus, Open Nebula
  - ...but has not been realized in Openstack as yet

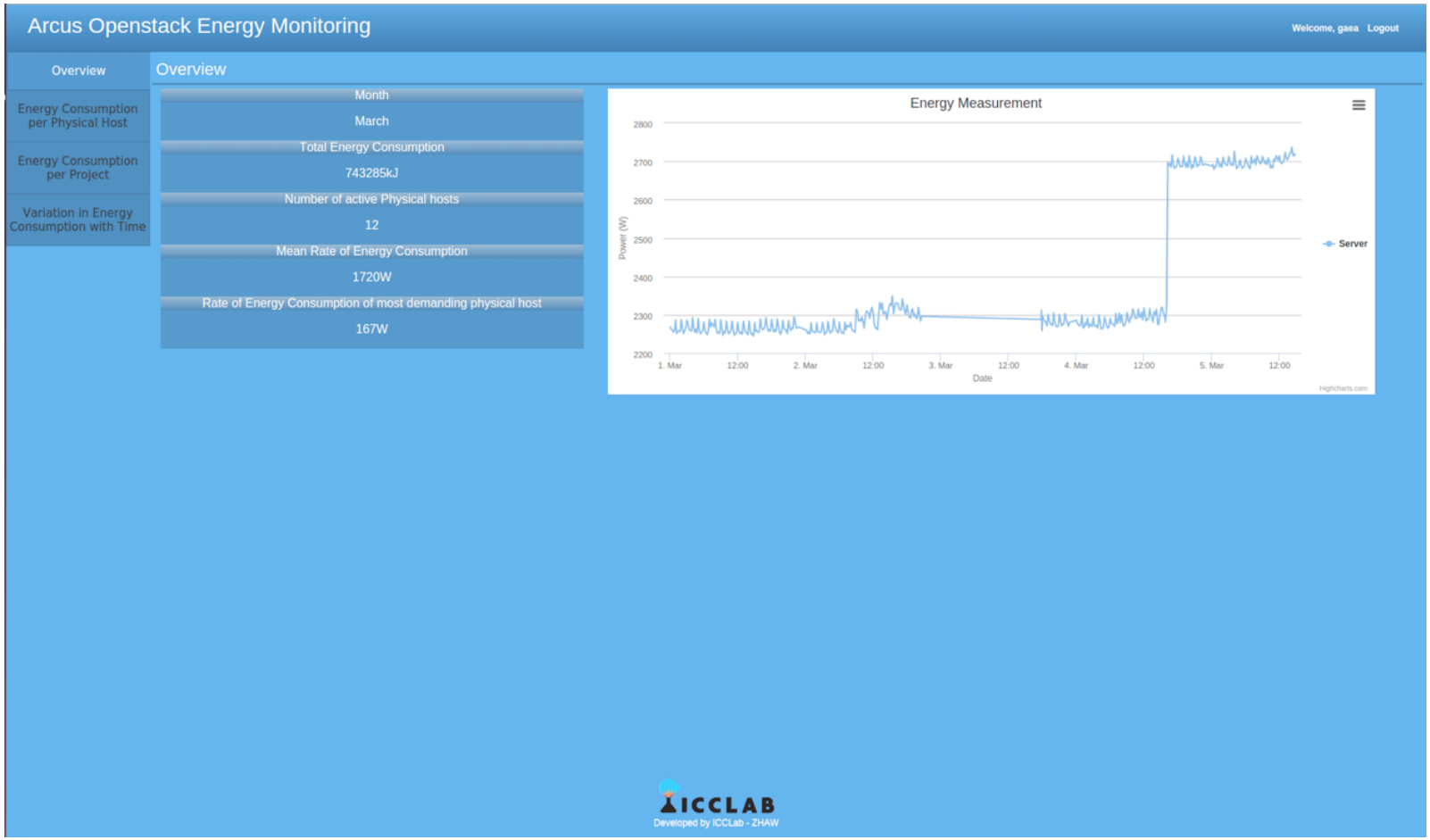
So what did we do?

- Develop Openstack-based Energy Monitoring solution
- Initial work on developing control mechanism to increase energy efficiency...
  - ...tied into some advanced, more robust live migration mechanisms...
  - ...essentially focused on powering-down servers when possible

The (Arcus) Energy Monitoring Tool

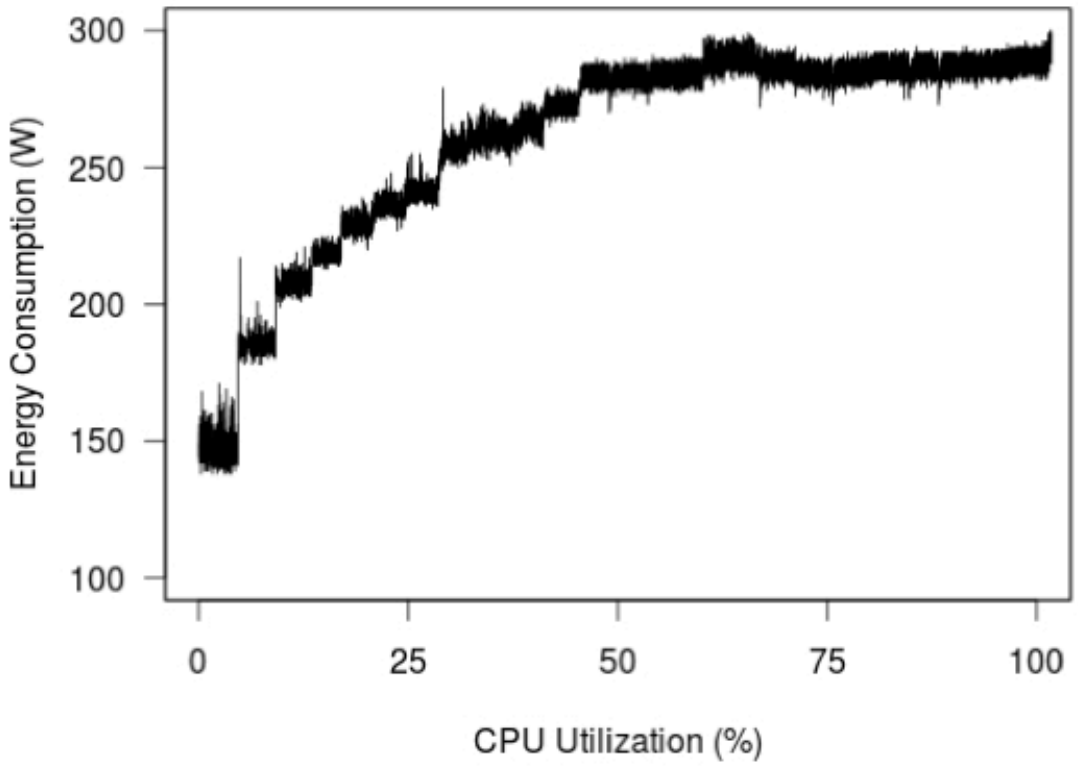
- Openstack focused Energy Monitoring Tool
  - Primarily designed to understand our own energy consumption
- Leverages Kwapi
  - Energy Monitoring subsystem within Openstack ecosystem
    - Supports data collection from disparate energy monitoring devices
    - Stores in Ceilometer
  - Collects information from libaem (IBM servers) and Supermicro IPMI tool (Supermicro servers)

Screenshot...



...and more to follow...

CPU utilization and Energy Consumption



...IBM x3550 M4, Dual Xeon E5-2640 processors...

Energy Aware Load Management

- Basic approach is to perform load consolidation
  - 50-75% utilization is better operating point from energy perspective
- Avoid servers with small amounts of work
  - Baseline energy consumption high when server powered up
- Power down servers when possible...
  - ...and WakeOnLan to revive them...

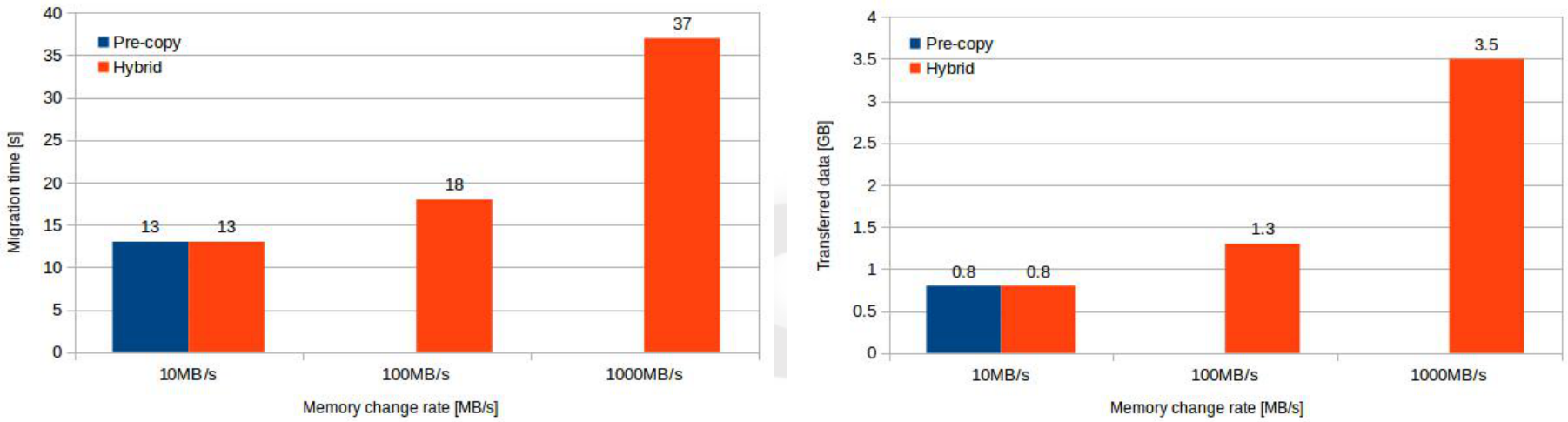
Live Migration...a little detour...

- Standard VM live migration operates using so-called pre-copy approach
  - Source remains active while memory copied to destination
- Has some issues with robustness...
  - ...may not converge, depending on the memory activity of the VM
- Post-copy live migration offers alternative
  - ...does not suffer from convergence problem
- Hybrid solution best, offering performance and robustness

Deployed hybrid live migration in Openstack

- Requires the use of
  - patched kernel
    - with userfaultfd(), remap_anon_range()
  - patched qemu
    - with hybrid, postcopy supports
  - patched post-copy aware libvirt
- Quite stable, but not a straightforward deployment process

Performance of hybrid LM



Varying Memory Change Rate with AppMembenchTool: 10 MB/s, 100MB/s, 1000MB/s
Post-copy succeeds in every single scenario (downtime ~0.5s)
Pre-copy convergence very unpredictable ~100MB/s MCR

And?

- With robust migration mechanisms, load consolidation more reliable
- Currently have basic load management mechanism
  - classifies servers by utilization - critical underload (<10%), lowly loaded, medium load, highly loaded, critical overload (>90%)
  - move critically overloaded load to other server
    - based on classification
    - power up new server if not possible
  - more critically underloaded load to other servers
    - based on classification
  - if load migrated off critically underloaded server, shut down server
- Have basic mechanism which works on our minimal lab resources
  - Need to test it on larger deployment
- Also implementing more sophisticated approach with basic simulation tool

Initial results

- Simulation results show significant savings (40%)
  - for synthetic workloads
- Significant savings possible for our own cloud based on workload analysis (40%)
  - Quite underutilized resources

Next steps

- Enhance the load management mechanisms and understand how much savings are possible in different contexts
- Deploy basic mechanisms on pseudo-production systems

Here's Bruno...

Acknowledgement...

- Bruno and Vojtech did all of this work ;-)

http://blog.zhaw.ch/icclab/

@icc_lab

murp@zhaw.ch