OPEN DAYLIGHT SUMMIT

OPEN DAYLIGHT SUMMIT

End to End SLA for Enterprise Multi-Tenant Applications

Girish Moodalbail, Principal Engineer, Oracle Inc.
Venugopal Iyer, Principal Engineer, Oracle Inc.

#ODSummit

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Introduction
- Application centric networking
- Multi-tenant application
- End-to-End SLA enforcement using Oracle Solaris and OpenDaylight

Oracle Solaris and OpenDaylight

- We are part of the Oracle Solaris Networking team
- Oracle Database 12c is the latest generation of database from Oracle. It introduces a new multi-tenant architecture
- We are working on providing end-to-end service for Oracle Database 12c tenants leveraging
  - OpenDaylight for scalability and Flow management and
  - Oracle Solaris Network Virtualization features on the edge
- Oracle is a silver member of ODL
- Oracle Solaris plans to support ODL in a future release on both x86 and SPARC platforms

Core Tenet of Cloud/Data Center - It's All About Applications

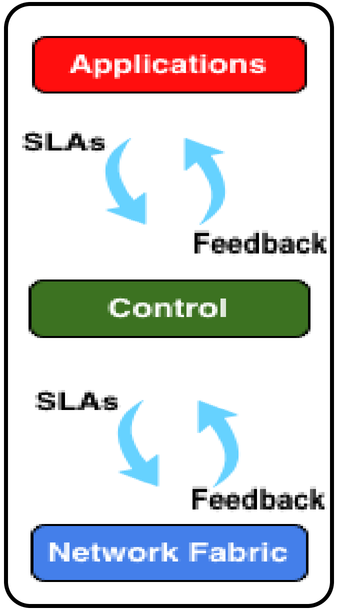
- Running Applications is the main purpose of the Cloud/Data Center
- Application developers are increasingly deciding the value of a Platform/Cloud
- Cloud needs to be Application centric and provide services oriented towards
  - Application monitoring
  - Application performance
  - Application security/isolation
  - Application availability
- Typically an application represents set of flows that must be treated end-to-end
  - Point of origin to Point of termination
  - Across virtual and physical switches

Existing Gaps between Applications and Networking

- Traditional Network Infrastructure challenges
  - Little awareness of application flows
  - Performance focused on “Speeds and feeds” instead of performance metrics like application flows completion times
  - Network unpredictability (configuration errors, unplanned outages, ..)
  - Networks are over provisioned and often underutilized
- Applications are becoming multi-tenant
  - Layer 2-4 support no longer is good enough
- Large number of different types of flows have to co-exist
  - Flows with different characteristics have to be treated differently in the network
  - Flow distribution and placement becomes important

Bridging Gaps between Applications and Networking

- Using Open Source SDN platform such as OpenDaylight its support for diverse SouthBound plugins
- Using OpenFlow and Service Function Chaining to program flows and services in the cloud
- Offloading services to the edge, when possible, to distribute processing
  - Further offloading services on the edge to underlying NICs for better performance
- An enterprise can get the most value when SDN is Application Driven
  - When clients of infrastructure are in control: Applications, Data Warehouses, Databases, …
  - … and not when SDN is Network Infrastructure Driven or Platform-specific Hypervisor Driven



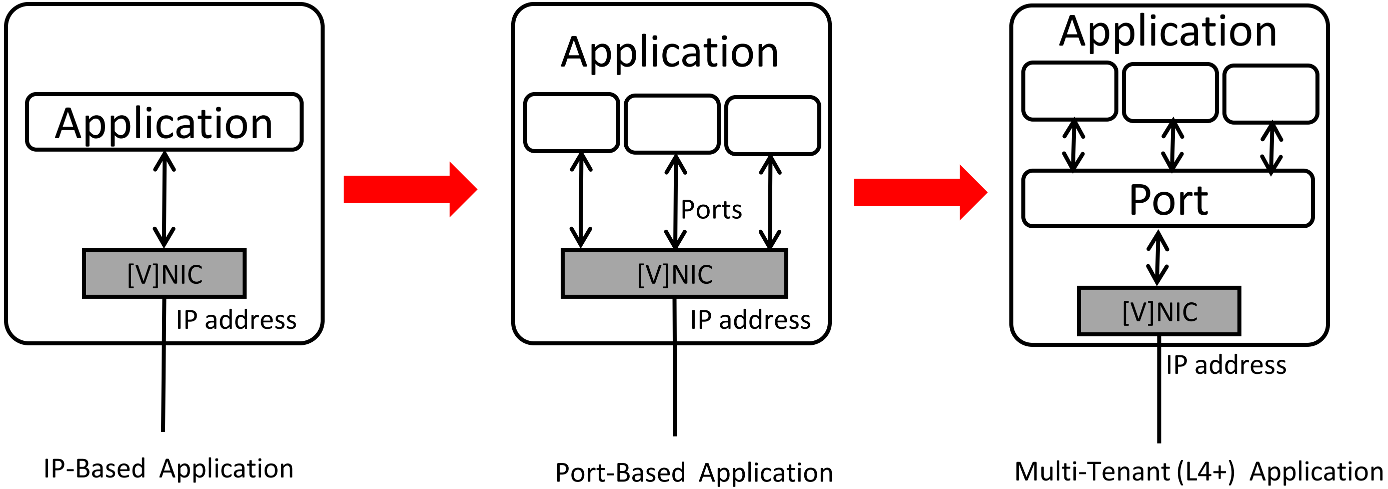
OPEN DAYLIGHT SUMMIT

Hypervisor Overlay in the Cloud/Data Center

- Network virtualization build on top of overlays
  - Needs just IP connectivity from the underlay
  - Decouples virtual network from underlying network fabric
  - Simple and rapid to deploy
- However physical network looses capability to provide differentiate services due to encapsulation
  - Geneve might improve this situation
- Virtual and Physical networks are separate entities with different management software, provisioning policies, and SLA
- Oracle Database 12c application requires support from both underlay and overlay

Multi-Tenant Application

Evolution of Application

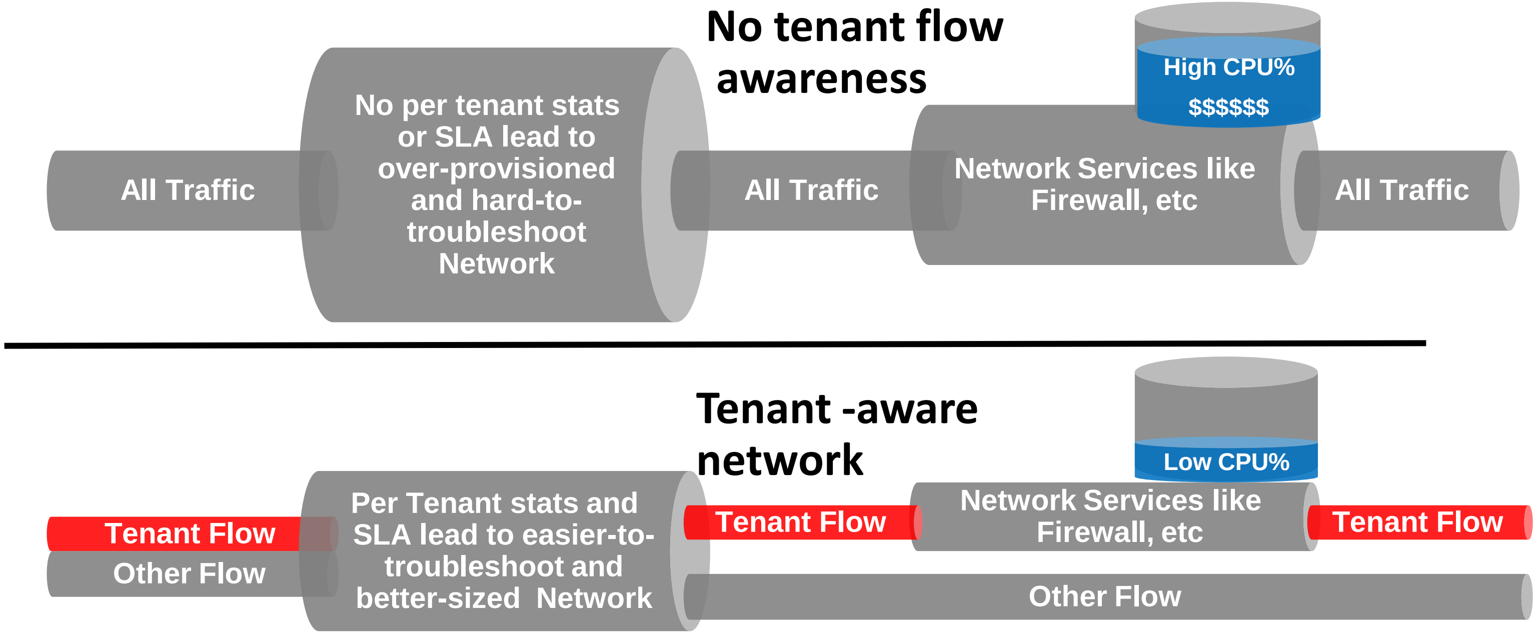


Multi-tenant Aware Enterprise Applications

- Pros:
  - An application instance shared among several tenants (economies of scale)
  - Fewer application instances to be deployed and maintained (economies of simplicity)
  - Highest consolidation density (economies of cost)
- Cons:
  - Need to ensure isolation between the tenants (Security)
  - Need to ensue one tenants activities does not affect other tenants (Noisy Neighbor)
  - Failure of application means all the tenants are affected (Availability)
  - Enforcement of SLA across various tenants becomes a challenge
- SDN is a very good fit for addressing these challenges for Cloud providers

Why Support Multi-tenant Application Flows?

- Better utilization of bandwidth and other resources



No tenant flow awareness

All Traffic

No per tenant stats or SLA lead to over-provisioned and hard-to-troubleshoot Network

All Traffic

High CPU%

$$$$$$

Network Services like Firewall, etc

All Traffic

Tenant -aware network

Tenant Flow

Other Flow

Per Tenant stats and SLA lead to easier-to-troubleshoot and better-sized Network

Tenant Flow

Low CPU%

Network Services like Firewall, etc

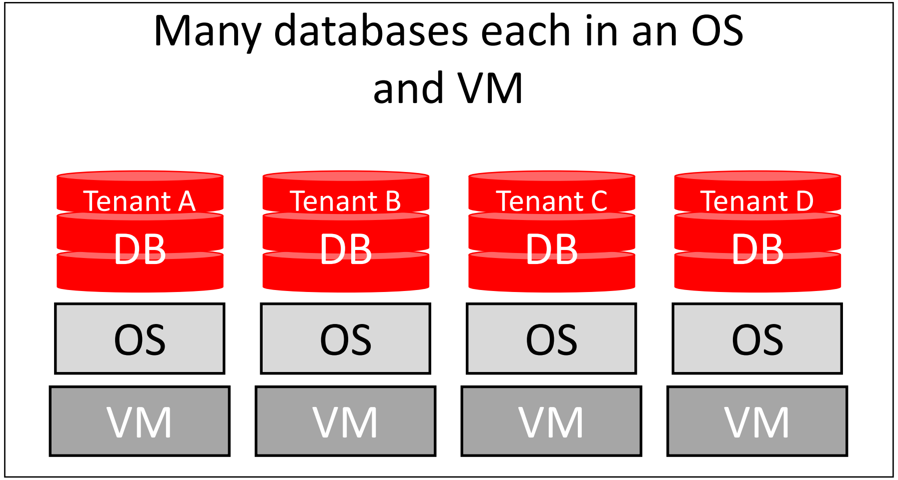
Tenant Flow

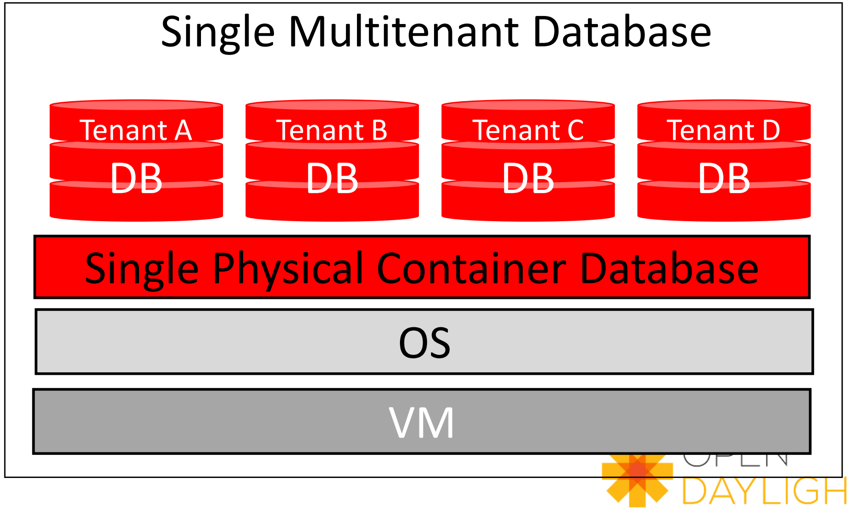
Other Flow

Application Tenant Flow Profile
SLA (Priority, Bandwidth, Latency)
Networking Services (Firewall, etc)

Multi-tenant Aware Oracle DB 12c

- A multitenant container database (CDB) that includes zero, one, or many pluggable databases (PDBs) created for tenants



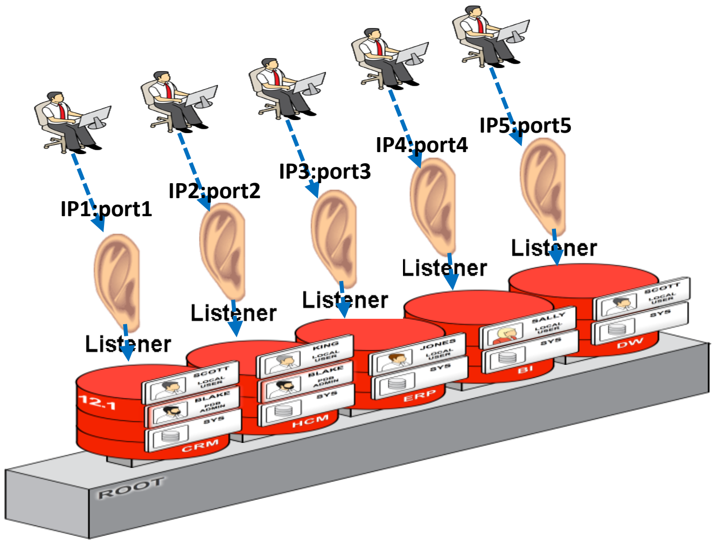

Single Multitenant Database
Tenant A
DB
Tenant B
DB
Tenant C
DB
Tenant D
DB
Single Physical Container Database
OS
VM

Enforcing End-to-End SLA for Multi-Tenant Aware Enterprise Applications

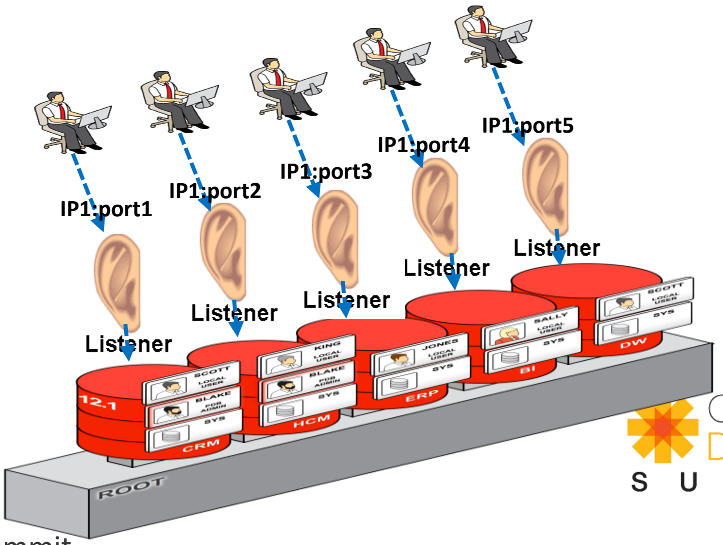
Multi-Tenant Aware Oracle DB 12c – Identifying Tenant Flows

- In the two configurations below, it is easy to identify per-tenant DB flow and push the flow across the network

Dedicated Listener per-tenant with unique VIP:PORT#

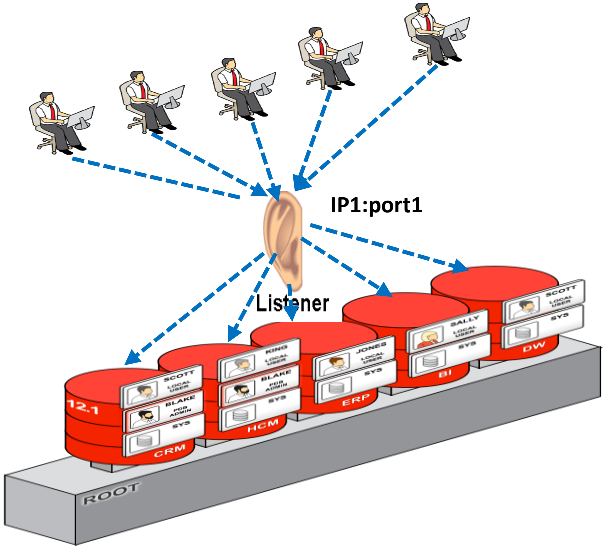


Dedicated Listener per-tenant with same VIP but different PORT#



Multi-Tenant Aware Oracle DB 12c – Identifying Tenant Flows

Shared Listener for all tenants with same VIP and same PORT#

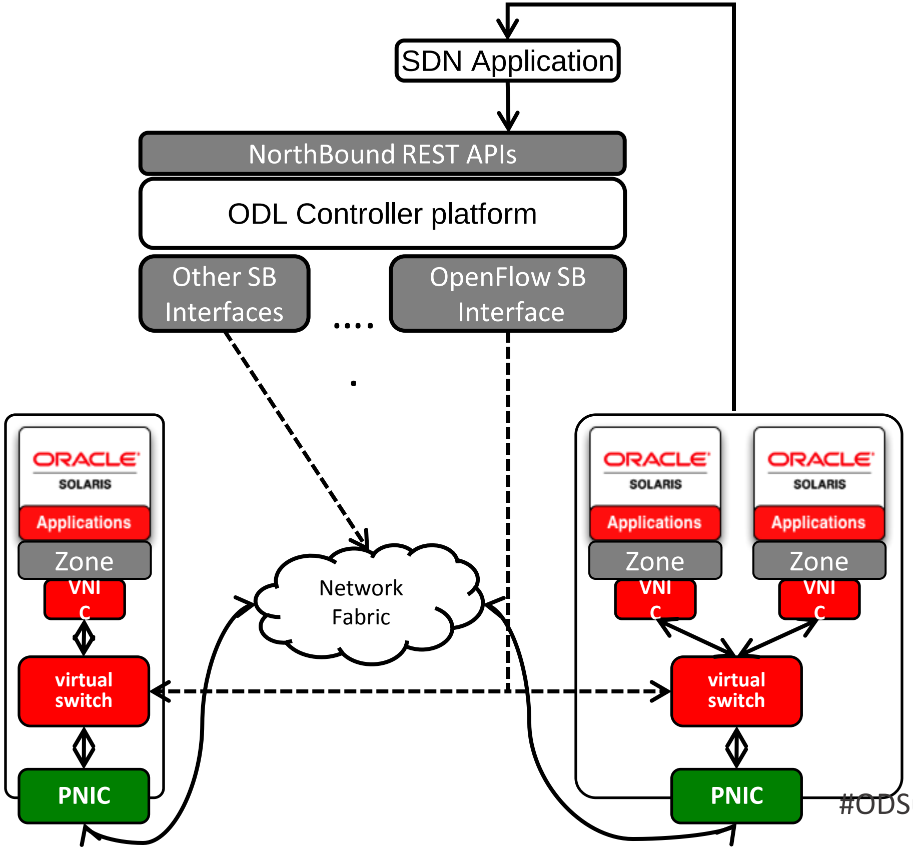


- Per-tenant DB flows can’t be identified using L4 information since they are all same
- Network Isolation is done using connection string
- Necessitates the need for L5-L7 classification on the host either
  - Using an efficient from of Deep packet inspection or
  - Providing APIs for application or its framework to identify flows using L4 information

Multi-Tenant End to End Flows

- L3-L4 flow classification is sufficient if the application is not multi-tenant
- For a multi-tenant application, we provide a mechanism for L5-L7 classification on the host to identify tenant flows, either
  - Using an efficient form of Deep packet inspection or
  - Providing API for application or its framework to identify tenant flows and mapping them to L4 based flows
- Once we identify a tenant flow, we
  - Translate the tenant flow to a L3-L4 flow and use OpenFlow rules to enforce SLA end to end
  - Alternatively, we could tag the packet (in an overlay network) and use the tag to enforce SLA end to end using OpenFlow
- We rely on an external mechanism to configure policies for the application flows

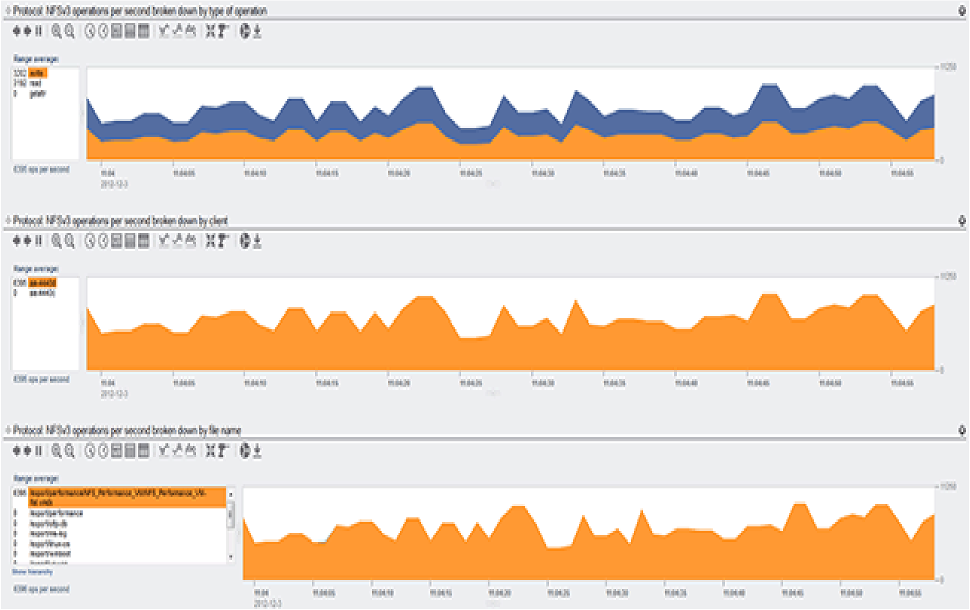
Multi-Tenant Flow using ODL and OpenFlow



- Identify an application flow based on L5-L7
- Translate that flow into multiple L3-L4 rule OpenFlows
- Install OpenFlow rules on the nodes in the network for that Application
- Install corresponding SLA (out-of-band) for that flow
- Packet is processed in the network based on OpenFlow rules
- SLA is enforced on all the nodes for the flow
- SLA is enforced in the network using DSCP

Multi-Tenant Monitoring and Tracing

- Network services are per tenant
- Monitor each tenant statistics
- Co-relate networking application performance and network state
- Provide hooks to fault management framework
- Provide flow tracing


Protocol: NFSv3 operations per second broken down by type of operation
Range average:
write
read
getattr
Protocol: NFSv3 operations per second broken down by client
Range average:
Protocol: NFSv3 operations per second broken down by file name
Range average:

Service Function Chaining

- Application processing involves subjecting packets to several potential services
  - Load balancer
  - Firewall
  - NFV functions
- We plan to rely on ODL framework to stitch services together for applications
- For multi-tenant applications, pushing SFC to the edge allows us to optimize service processing

SFC Classification on the Edge

- SFC classification based on L5-L7 is more practical on the edge
- SFC chains can be optimized by collapsing the functions (assuming all the services are on the same host)
- Further functions could be offloaded to the hardware, where supported, to improve performance
- Optimized edge services for enterprise applications can be chained with additional services, e.g. if Tenants are exceeding their SLA, then they could be diverted to additional services

Oracle Solaris and OpenDaylight

Oracle Solaris – Security. Speed. Simplicity.

- Full VM lockdown with immutable file systems
- Automated compliance monitoring and reporting
- Services run with minimal necessary privileges

- Automatic cryptographic offload
- Zero overhead virtualization
- Highest performance with Software in Silicon

- Application driven SDN
- Highly integrated OpenStack
- Agile self-service environments

- Co-engineered with Oracle hardware and software stack
- Engineered for mission critical workloads

Cloud-scale networking with Oracle Solaris

Network Virtualization	Virtual NICs (VNICs), VNIC migration, Virtual switching, Hardware-assisted virtualization, Automatic VNICs for zones, SR-IOV Integration, VLAN isolation, Anti-spoofing protection, Converged Storage/Networking with DCB, Edge Virtual Bridging, Elastic Virtual Switch, L2 HA (DLMP)
Resource Control	Integrated QOS, Bandwidth limits, Mapping to CPUs or CPU pools for isolation
Performance	Parallel stack, NUMA I/O Framework, SR-IOV Integration, Dynamic Polling, Buffer Management, Pre-mapped buffers, Kernel Socket API, 4x Lower latency vs KVM, Converged Ethernet, Large Receive Offload
Built-in Network	Routing, Firewall, Load Balancing, VRRP, Bridging
Management	IPMP re-architecture, Vanity naming, Automatic IP configuration, Centralized IP administration, Centralized data link administration, Consolidated data link properties, Integrated data store, GLDv3 unification for legacy drivers,
Observability	Real-time data link, hardware, and flow statistics. History integrated with extended accounting. Capture local traffic through through virtual switch and IP loopback path.
APIs	Committed GLDv3 APIs, pluggable TCP congestion algorithms, IP Filter Hooks, Kernel socket API, Socket level Flow APIs
OpenStack Neutron	OpenStack Neutron integration with Elastic Virtual Switch

OPEN DAYLIGHT SUMMIT

Oracle Solaris and ODL

- We plan to support Open vSwitch on Solaris
- We have compiled and run ODL on Solaris
  - leveldb from community does not work on Solaris. We have ported it to Solaris and are working with the community to make it available upstream
  - Currently Node.js is not supported on SPARC
- Some of the issues we have come across
  - Determining the license for all the components in ODL. We have received useful information from the community. Thanks!
  - Testing frameworks in the community uses tools such as mininet that are not available on all the platforms. We'd encourage the community to think about how to generalize this, to the extent possible

Oracle Solaris and ODL

- Reach out to us about participating in the Platinum Customer Beta to get early access to Oracle Solaris support for ODL
- We'd also like to encourage the community to make sure ODL works cross-platform (x86, SPARC etc.) and testing (build and functionality) is designed with that in mind
- We plan to work with the community by providing build environments for Oracle Solaris

Thank You

OPEN DAYLIGHT SUMMIT