Limelight NETWORKS

OTT Video @ the Edge

Steve Miller-Jones | VP, Product Strategy

Mile High Video 2019

ITS ALL ABOUT THE EDGE

"The combination of **billions of IoT devices and 5G networks** is set to drive a **profound change** in the way computing **workloads are deployed**." (ZDNet)

"The era of the **cloud's total dominance** is drawing to a close" (The Economist)

"Edge Computing Can Make Industries '**Massively More Efficient**' (Microsoft)"

"Real-Time Video Analytics: The **'Killer App' for Edge Computing**" (IEEE Computer Society)

IN THE NEXT @30 MINS

TOPICS

- How are service orientated architectures being used in OTT workflows?
- What parts of the video workflow will benefit most from the availability of compute resources at the edge?
- How is Limelight helping?

LETS HAVE SOME DEFINITION

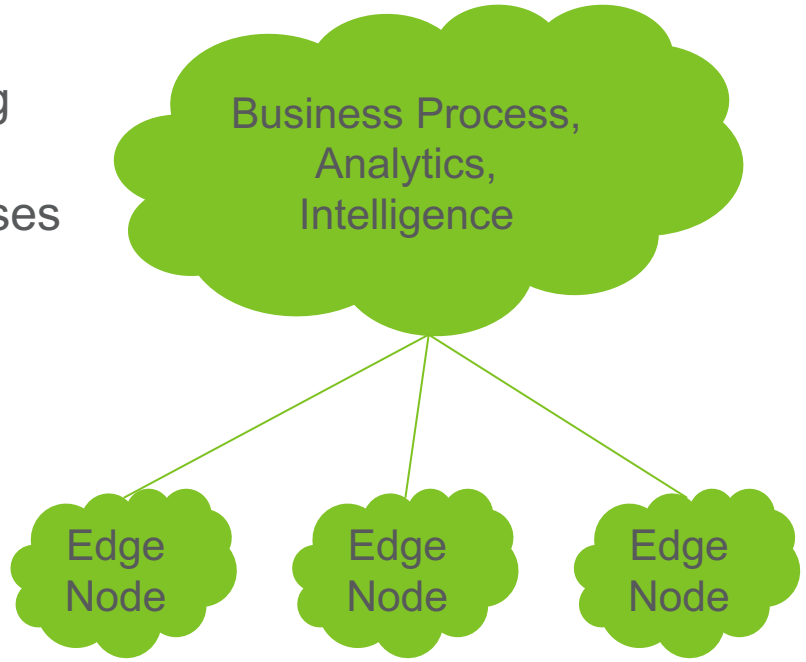
WHAT IS EDGE COMPUTING?

Edge Computing: A distributed computing paradigm which brings computation and data storage closer to the location where it is needed, to improve response times and save bandwidth.

Service Orientated Architecture: A software architecture concept based on a request/reply design paradigm for synchronous and asynchronous applications. Services are provided to application components by other components of varying granularity, through a communication protocol over a network.

OTT VIDEO PROCESS STACK

The Cloud
Big data processing & warehousing
Business logic processes
Coarse grained application processes



OTT Video examples
Encoding
Packaging
Content Origin

The Edge
Local / global network
Data analysis & reduction
Control response
Fine grained application processes

OTT Video examples
Packaging
Security
Beacon collection
Ad insertion

Higher
Processing speed / response time
Lower

DEMANDS OF THE DIGITAL PRODUCT

CUSTOMER, PLATFORM, PROVIDER

THE CHALLENGE

DEPLOY CUSTOM VIDEO WORKLOADS

1. Provide a scalable environment that can be used with the CDN, that enables customers to deploy custom OTT workloads to the edge of the delivery network.

2. Scale this on demand.

3. Make it suitable for low-latency startup and execution of workloads.

STATEFUL OR STATELESS?

SOME QUESTIONS TO DRIVE DIRECTION

- How are Service Orientated Architectures and Microservices are being used in OTT workflows?
- What are the challenges in moving from monolithic vendor led software applications to SOAs and edge services?
- What parts of the video workflow will benefit most?

Stateful – Needs access to more resources, requires availability of a range of additional services, imposes operational overheads outside scope of concern

Stateless – Simple enablement of customer provided functions that are latency sensitive parts of the video and end user workflow.

DEVELOPING A SERVERLESS ENVIRONMENT

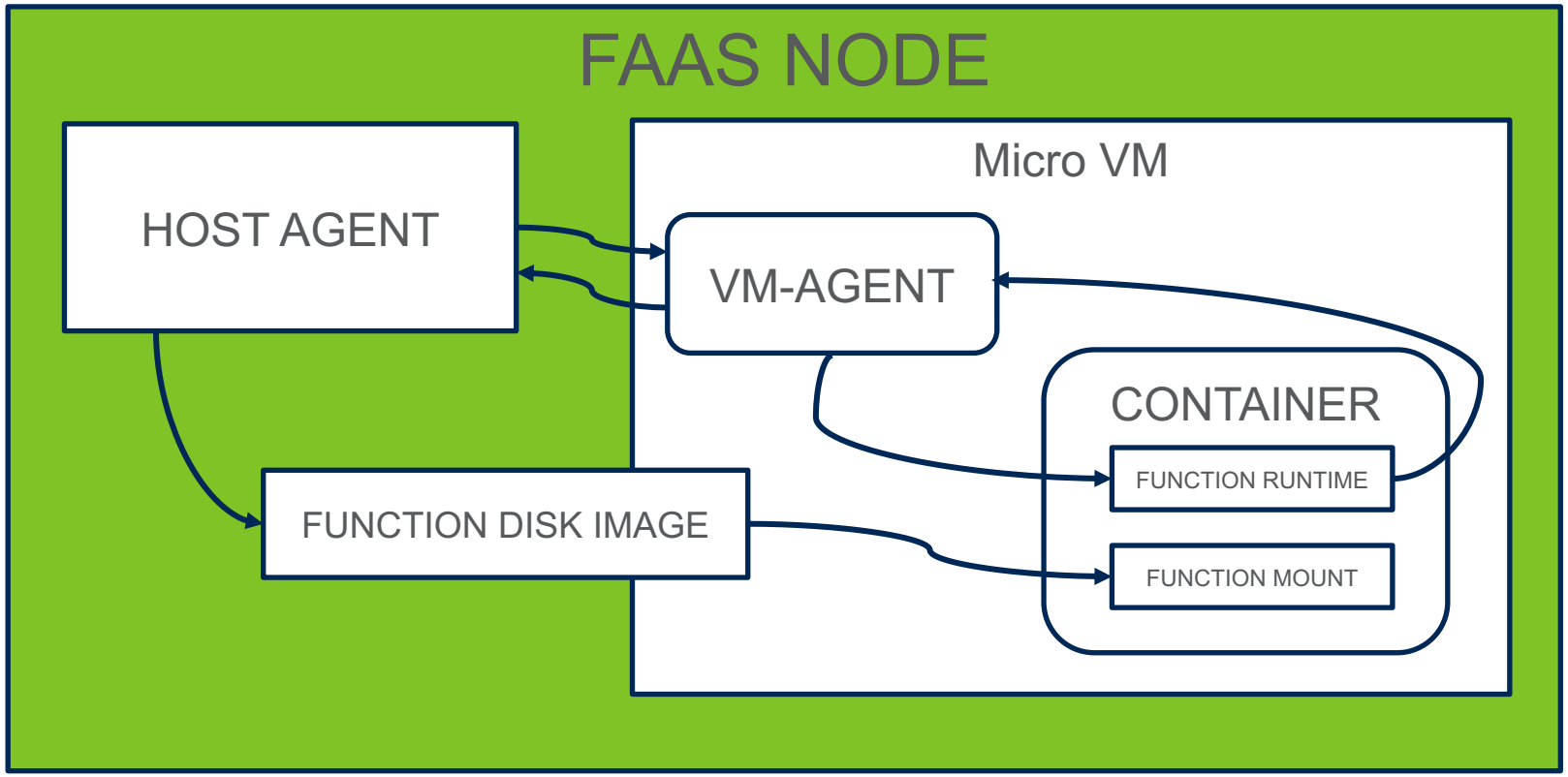
CORE PRINCIPALS FOR ARCHITECTURE

- Distributed function metadata and archives
- Customer uploads code and libraries to persistent storage
- FaaS nodes perform caching of function archives in local filesystem when accessed
- Load shedding functionality used/required to prevent individual nodes from being overwhelmed / overloaded
- Using CDN as a load-balancer.

WHAT'S IN A FAAS NODE?

OVERVIEW

- FaaS node provides an on-demand function execution environment.
- Using a combination of open source and custom services.
- Function runtime is hosted inside a container, in a VM.
- Functions are aliased, routable and fronted by the CDN: distribution, scale, load-balancing, caching.



RESOURCE MANAGEMENT

THE PROBLEM OF LOAD BALANCING

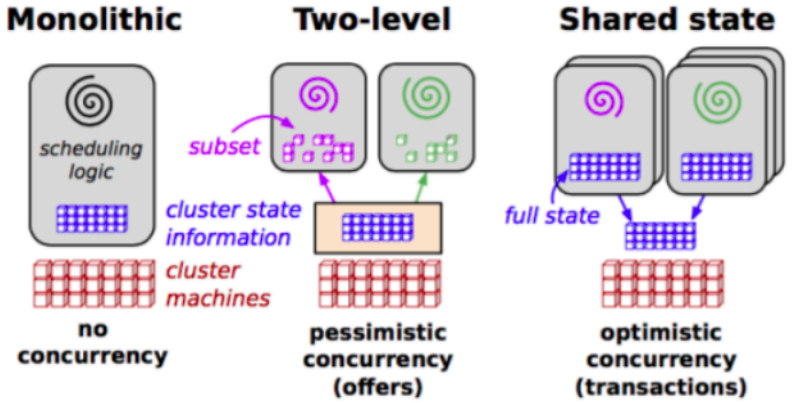
Managing resources (load balancing) in a computing cluster to achieve optimal utilization is one of the fundamental problems of distributed computing which can be reduced to the "Bin packing problem":

In the bin packing problem, objects of different volumes must be packed into a finite number of bins or containers each of volume V in a way that minimizes the number of bins used.

RESOURCE MANAGEMENT

RESEARCH

The problem has received considerable amount of attention from both academia and industry researches, proposing solutions ranging from centralized monolithic to decentralized opportunistic schedulers.



- *"Large-scale cluster management at Google with Borg" (Google)*
- *"Omega: flexible, scalable schedulers for large compute clusters" (Google)*
- *"Sparrow: Distributed, Low Latency Scheduling" (UC Berkeley)*
- *"The Power of Two Choices in Randomized Load Balancing" (UC Berkeley)*
- *"Apollo: Scalable and Coordinated Scheduling for Cloud-Scale Computing" (Microsoft Research)*

HOW TO SOLVING FOR LOAD?

RESOURCE SCHEDULING

Resource schedulers place new instances on the least loaded servers, based on resource allocation of a cluster in an availability zone, and the algorithm we have implemented.

Premise:

Attempt to achieve fair resource allocation at the customer level.

To improve scheduling decisions, allocation is facilitated by dynamic feedback on queue wait times, provided by function agents.

INVOKING FUNCTIONS

WHERE TO INVOKE, THAT IS THE QUESTION

Running in clusters throughout the network, we have to determine where to place a function instance so that it can be invoked.

Achieved with a scheduling service

To start a new instance of a function:

- Pick some participating servers
- Read their resource availability,
- Decide if a new instance can be created.
- If yes, a server is picked and this data distributed back to the scheduler.

INVOKING FUNCTIONS

SCALE UP, SCALE DOWN

- Host agent starts new instances based on requests received from load balancing.
- Host agents share the state of an "availability zone" and are responsible for adding / removing instances based on client wait time feedback and configured concurrency limit.
- Functions automatically scaled down based on resource demand and defined max idle time

CUSTOMER WORKFLOW

INTERACTION WITH A FUNCTION

1. CREATE A FUNCTION
2. WRITE CODE, TEST UNTIL READY TO PUBLISH
3. PUBLISH FOR PUBLIC ACCESS, USING AN ALIAS
4. CONTINUE DEVELOPMENT ON “LATEST” UNTIL READY TO UPDATE
5. CUT NEW VERSION, UPDATE PRODUCTION ALIAS TO POINT TO IT
6. ROLL-BACK THE UPDATE BY POINTING ALIAS TO PREVIOUS VERSION
7. REPEAT STEPS 1-6 AS NEEDED
8. DELETE FUCNTION

Possible example use cases

Video

- Ad Insertion
- Watermarking
- Performance Beacon collection

Web Content

- Personalization
- Content targeting
- A/B testing
- Image resizing

Apps & Others

- Edge Authentication
- API gateway
- Service data collection

DEMANDS OF THE DIGITAL PRODUCT

CUSTOMER, PLATFORM, PROVIDER

Limelight NETWORKS

info@llnw.com

twitter.com/llnw

linkedin.com/company/limelight-networks

facebook.com/LimelightNetworks