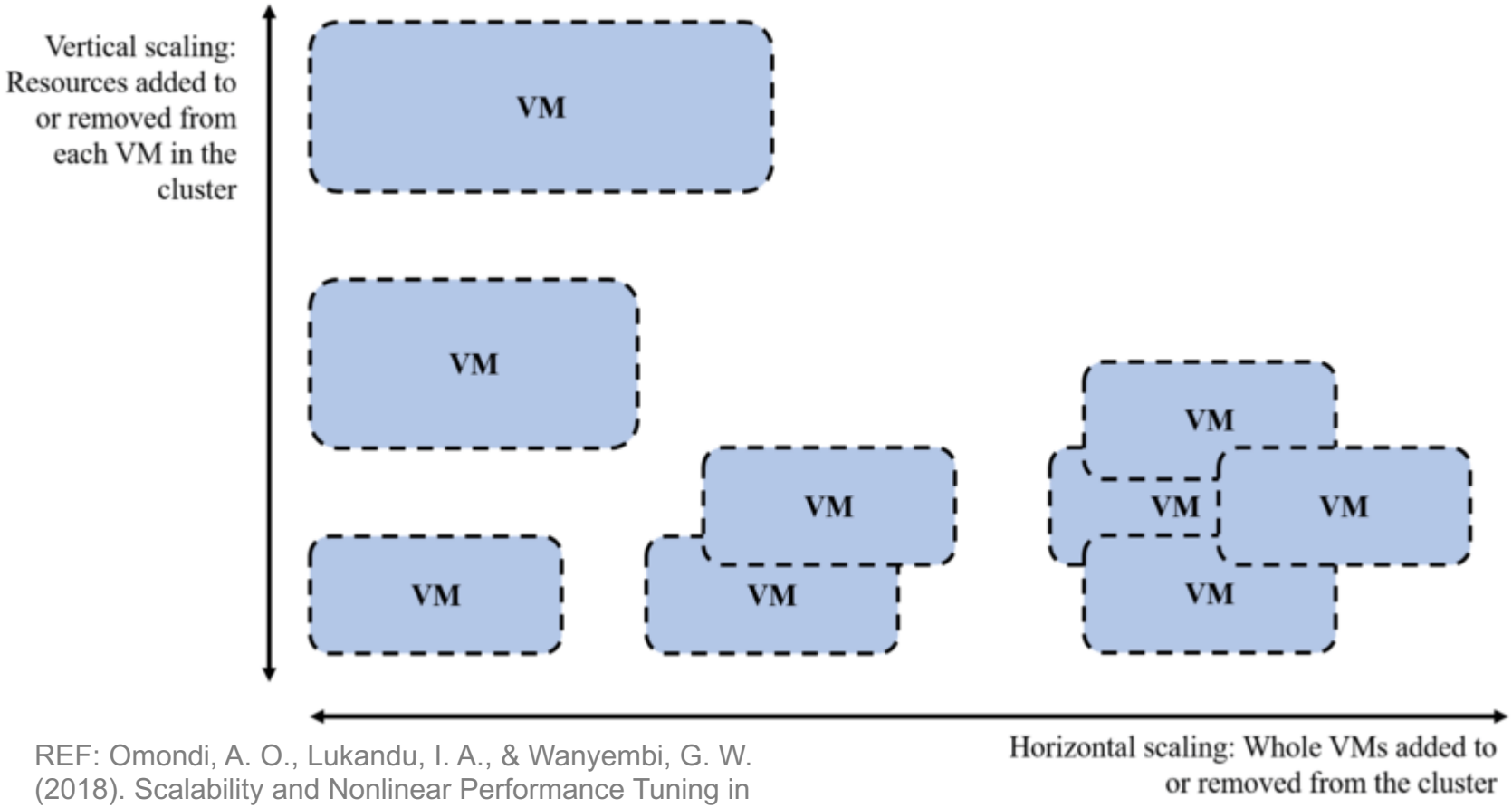
Vertical vs Horizontal Scaling



REF: Omondi, A. O., Lukandu, I. A., & Wanyembi, G. W. (2018). Scalability and Nonlinear Performance Tuning in Storage Servers. *International Journal of Research Studies in Science, Engineering and Technology*, *5*(9), 7-18.

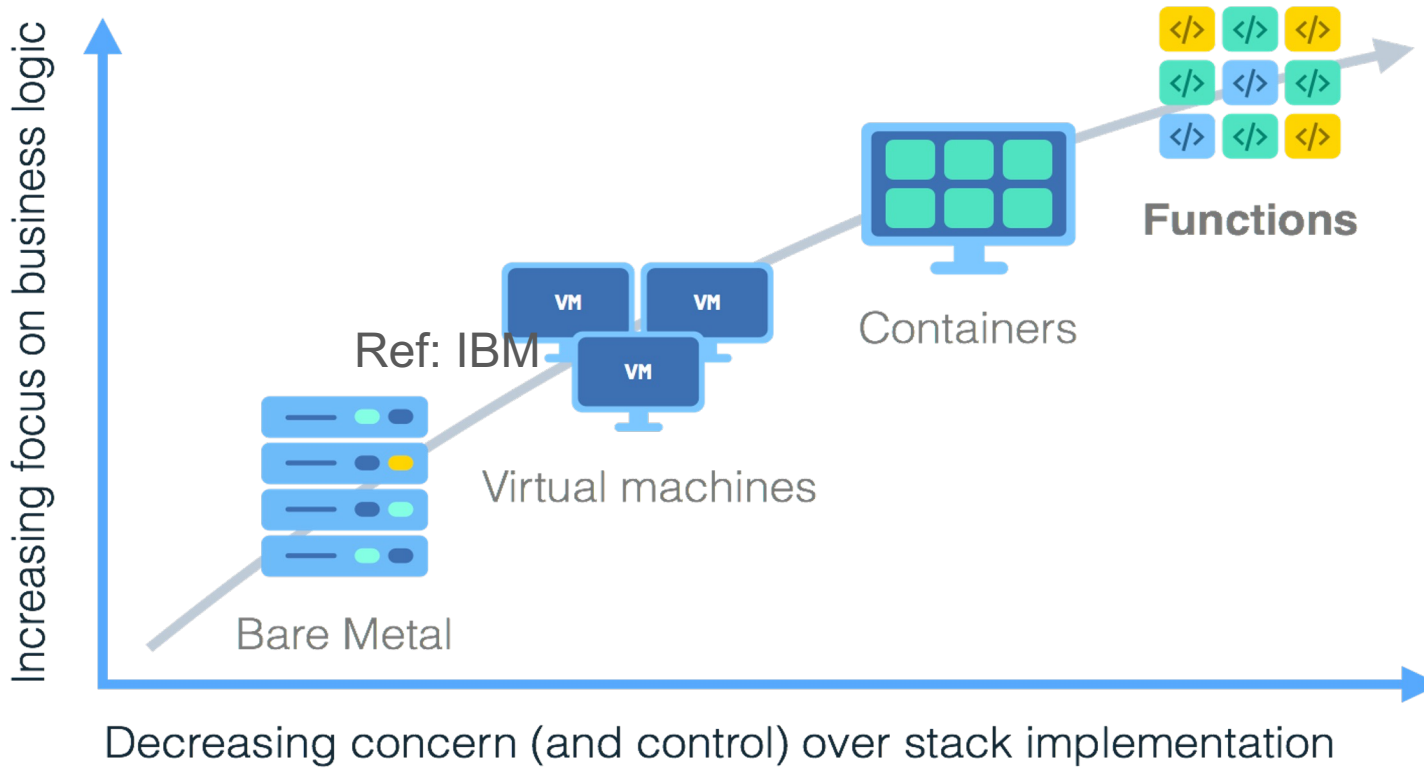
Comparison of Scaling Methods

- Cost
- Down time due to scaling
- Availability (Outage, H/W problems)
- Licensing fees
- Scale-up limit
- Impact on the data center

Serverless Computing

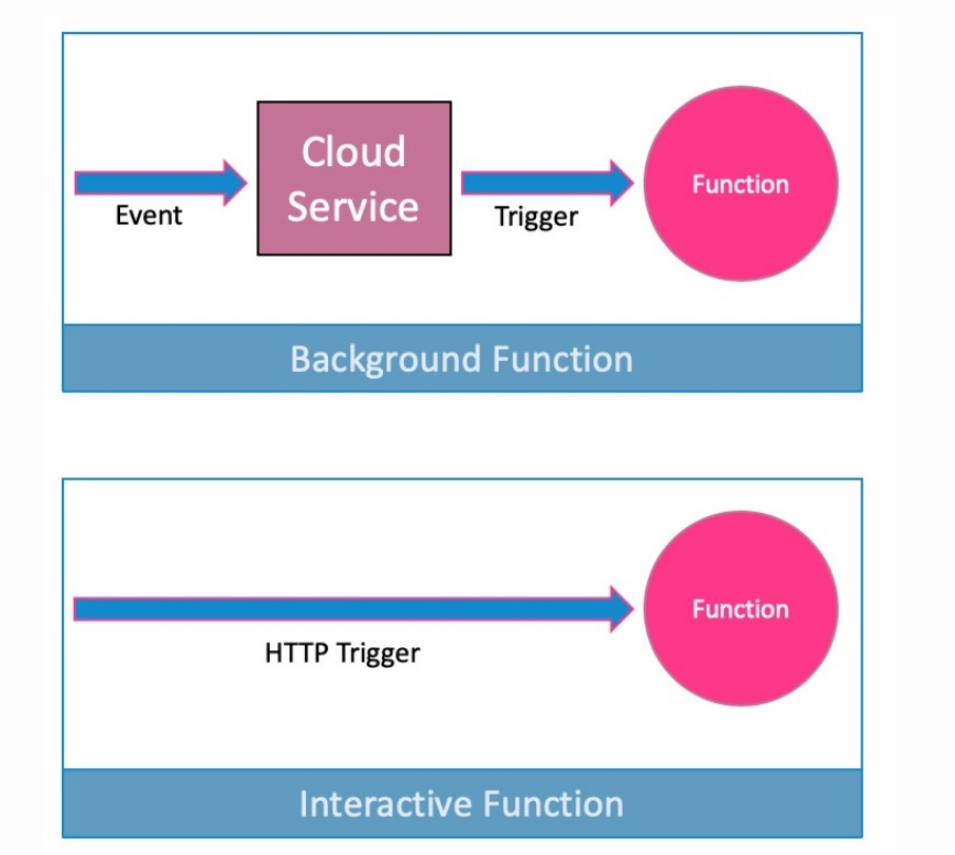
- Yet another step in Cloud Computing
- How to do it with “no servers”??
- FaaS: Function as a Service
- Functions: Small code snippets
- Main idea: Functions are executed on the cloud with no control over the computational resources.
- “No server” from the developer perspective.
- Resource Provisioning, monitoring, maintenance, scalability, fault-tolerance → Cloud Provider

Cloud Computing Evolution



Ref: IBM

Triggering FaaS Functions



REF: https://thenewstack.io/serverless-everything-you-need-to-know-about-google-cloud-functions/

Use Case (AWS Lambda)

Some details

- FaaS requires specific programming model and software architecture
- Function: Deployment unit
- Short running, stateless computation
- Streamlined for event-driven applications
- Pricing at the millisecond resolution
- Scale-down to Zero !

Comparison with Previous Approaches

	On-prem	VMs	Containers	Serverless
Time to provision	Weeks-months	Minutes	Seconds-Minutes	Milliseconds
Utilization	Low	High	Higher	Highest
Charging granularity	CapEx	Hours	Minutes	Blocks of milliseconds

Ref: Jason McGee, IBM; Serverless Conference 2017.

When to go Serverless

Serverless is good for

short-running
stateless
event-driven

- Microservices
- Mobile Backends
- Bots, ML Inferencing
- IoT
- Modest Stream Processing
- Service integration

Serverless is not good for

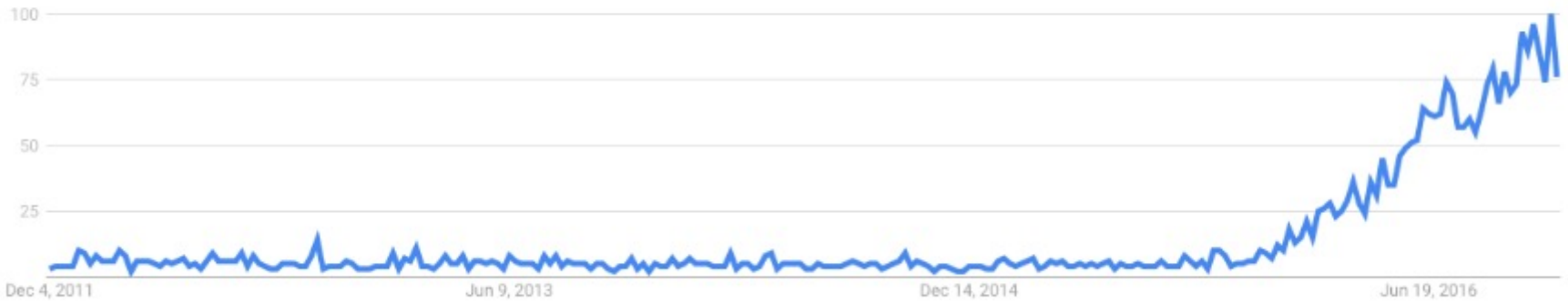
long-running
stateful
number crunching

- Databases
- Deep Learning Training
- Heavy-Duty Stream Analytics
- Numerical Simulation
- Video Streaming

Caveat: Things are changing fast. Research in 2021 uses FaaS for HPC and DL

Source: Opening of Third International Workshop on Serverless Computing (WoSC) 2018

Google Trends for “Serverless”



Platforms for Serverless Computing

AWS
Lambda

Azure Functions

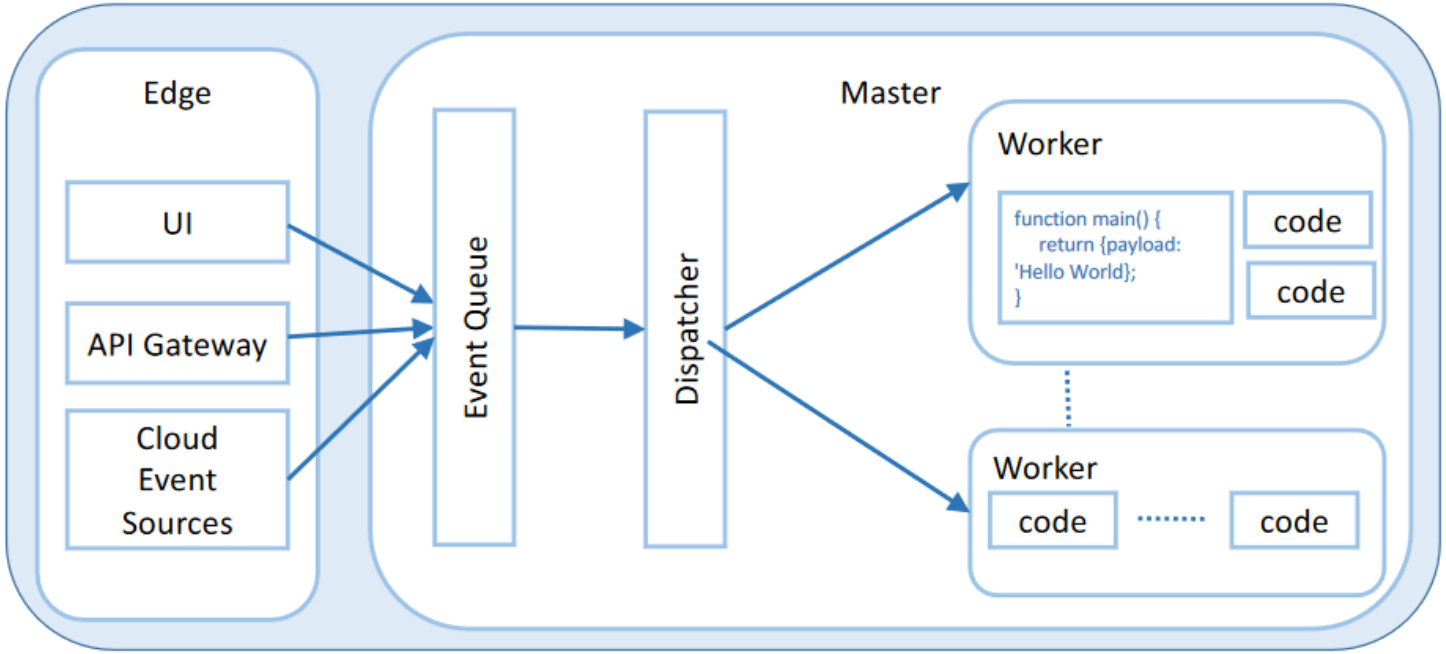
Kubernetes

IBM Cloud Functions

Red-Hat

Google Functions

Basic Architecture for FaaS



Ref: Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research advances in cloud computing* (pp. 1-20). Springer, Singapore.

Characteristics

- Cost
- Performance & Limits
- Programming Language
- Programming Model
- Composability
- Deployment
- Security & Accounting
- Monitoring & Debugging

Ref: Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research advances in cloud computing* (pp. 1-20). Springer, Singapore.

Programming Environments

AWS

Language and framework support policies

- **Node.js** – github.com
- **Python** – devguide.python.org
- **Ruby** – www.ruby-lang.org
- **Java** – www.oracle.com and Corretto FAQs
- **Go** – golang.org
- **.NET Core** – dotnet.microsoft.com

GCP

- Node.js Runtime
- Python Runtime
- Go Runtime
- Java Runtime
- .NET Runtime
- Ruby Runtime
- PHP Runtime

Programming Environments

AZURE

Languages by runtime version

Several versions of the Azure Functions runtime are available. The following table shows which languages are supported in each runtime version.

Language	1.x	2.x	3.x	4.x
C#	GA (.NET Framework 4.8)	GA (.NET Core 2.1[1])	GA (.NET Core 3.1) GA (.NET 5.0)	GA (.NET 6.0)
JavaScript	GA (Node.js 6)	GA (Node.js 10 & 8)	GA (Node.js 14, 12, & 10)	GA (Node.js 14) Preview (Node.js 16)
F#	GA (.NET Framework 4.8)	GA (.NET Core 2.1[1])	GA (.NET Core 3.1)	GA (.NET 6.0)
Java	N/A	GA (Java 8)	GA (Java 11 & 8)	GA (Java 11 & 8)
PowerShell	N/A	GA (PowerShell Core 6)	GA (PowerShell 7.0 & Core 6)	GA (PowerShell 7.0)
Python	N/A	GA (Python 3.7 & 3.6)	GA (Python 3.9, 3.8, 3.7, & 3.6)	GA (Python 3.9, 3.8)
TypeScript[2]	N/A	GA	GA	GA

Timeout limits - AWS

AWS Lambda enables functions that can run up to 15 minutes

Posted On: Oct 10, 2018

You can now configure your AWS Lambda functions to run up to 15 minutes per execution. Previously, the maximum execution time (timeout) for a Lambda function was 5 minutes. Now, it is easier than ever to perform big data analysis, bulk data transformation, batch event processing, and statistical computations using longer running functions.

You can now set the timeout value for a function to any value up to 15 minutes. When the specified timeout is reached, AWS Lambda terminates execution of your Lambda function. As a best practice, you should set the timeout value based on your expected execution time to prevent your function from running longer than intended.

This feature is available in all regions where AWS Lambda is available. Please visit our product page for more information about AWS Lambda or log in to the AWS Lambda console to get started.

Timeout limits - GCP

Timeout

Function execution time is limited by the timeout duration, which you can specify at function deployment time. By default, a function times out after 1 minute, but you can extend this period up to 9 minutes.

When function execution exceeds the timeout, an error status is immediately returned to the caller. CPU resources used by the timed-out function instance are throttled and request processing may be immediately paused. Paused work may or may not proceed on subsequent requests, which can cause unexpected side effects.

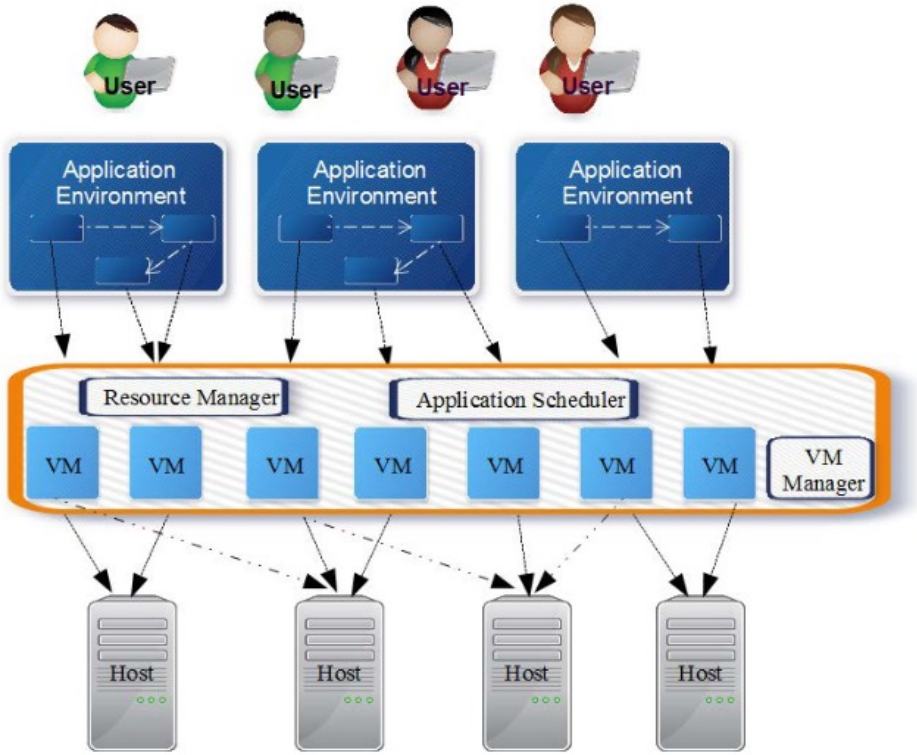
Timeout limits - Azure

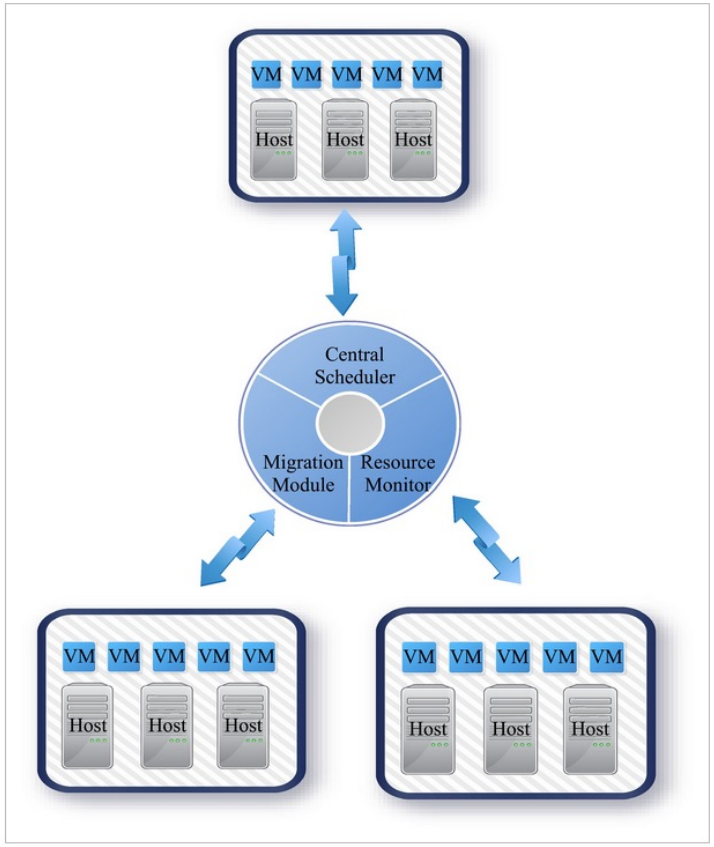
Function app timeout duration

The timeout duration of a function app is defined by the `functionTimeout` property in the host.json project file. The following table shows the default and maximum values in minutes for both plans and the different runtime versions:

Plan	Runtime Version	Default	Maximum
Consumption	1.x	5	10
Consumption	2.x	5	10
Consumption	3.x	5	10
Premium	1.x	Unlimited	Unlimited
Premium	2.x	30	Unlimited
Premium	3.x	30	Unlimited
App Service	1.x	Unlimited	Unlimited
App Service	2.x	30	Unlimited
App Service	3.x	30	Unlimited

App ←→ VM ←→Host Mapping





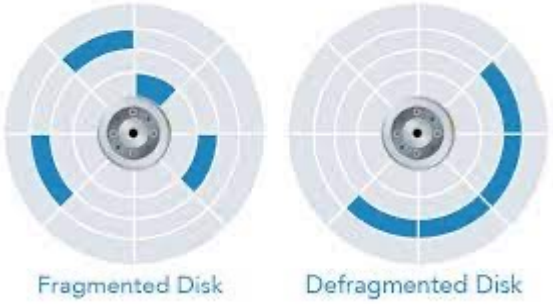
Xu, M., Tian, W., & Buyya, R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice and Experience*, *29*(12), e4123.

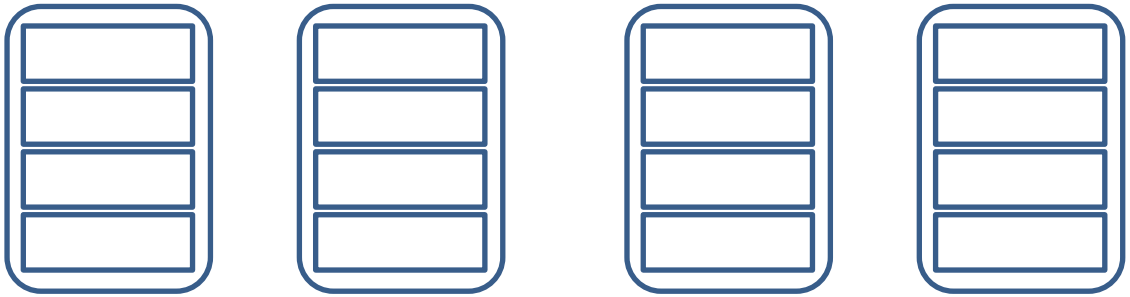
VM Migration

- Transfer of VM from one physical host to another.
- Resembles file transfer but there is much more !
- Why a CSP (Cloud Service Provider) need it?
  - Zero-down time HW Maintenance
  - Load Balancing
  - Server Consolidation
  - Across-site management: “Follow the sun”, “Follow the moon”
  - Hybrid Cloud Performance: Offload to public cloud when necessary
  - Break vendor lock-in
  - Dealing with user mobility: Edge scenarios

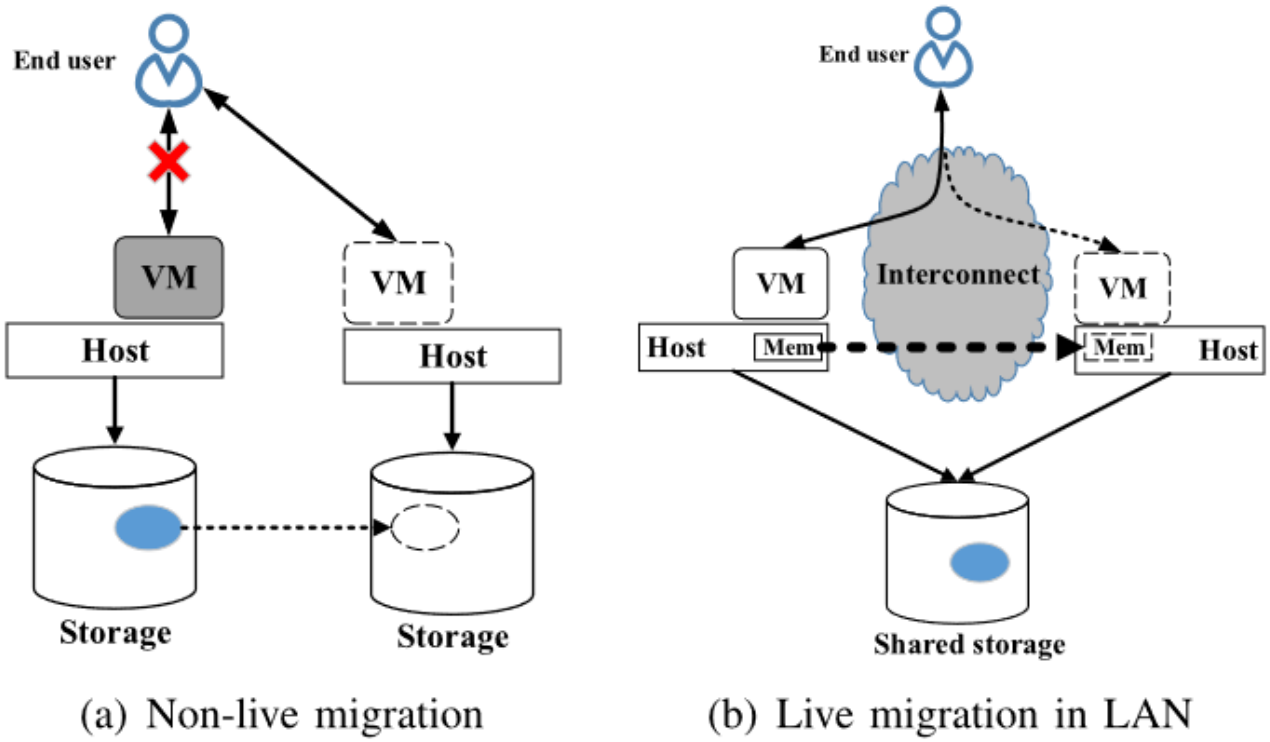
REF: Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, *20*(2), 1206-1243.

Cloud Defragmentation→ Server Consolidation



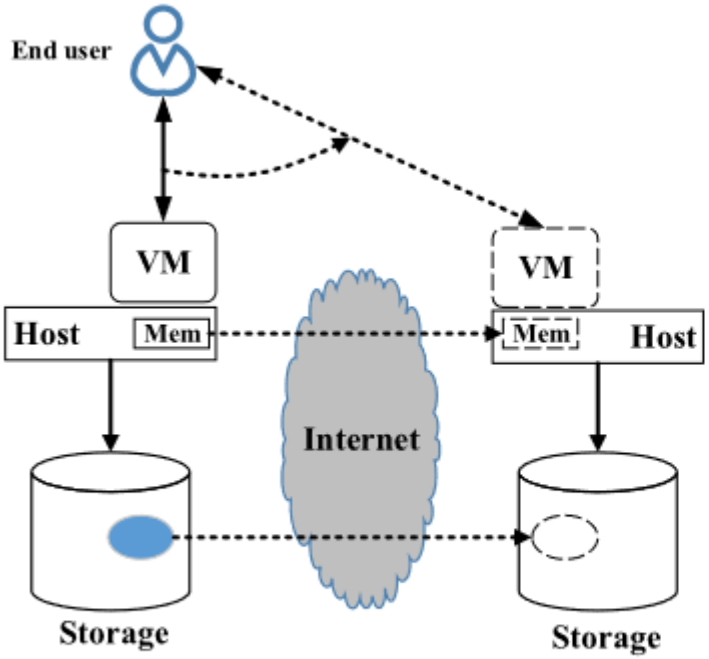


VM Migration - I

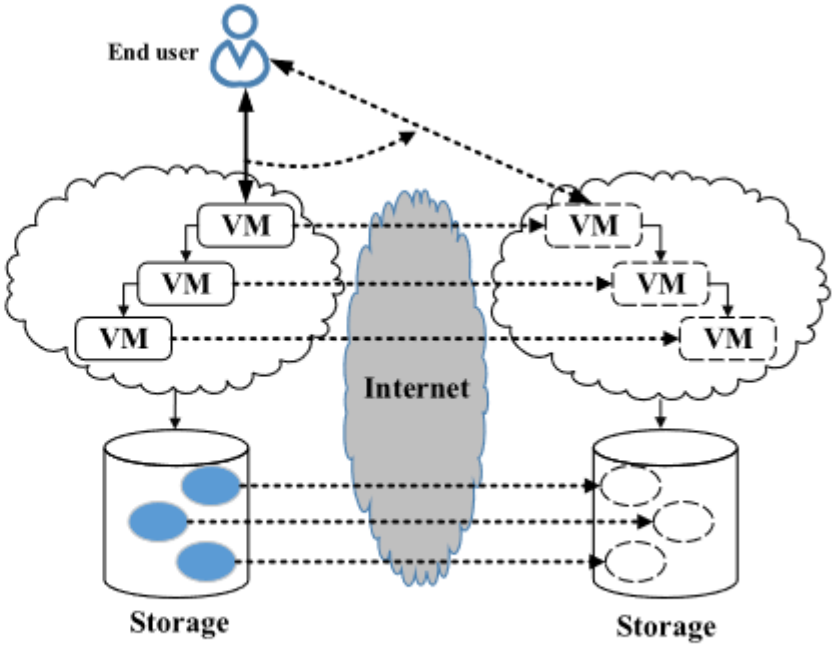


REF: Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, *20*(2), 1206-1243.

VM Migration - II



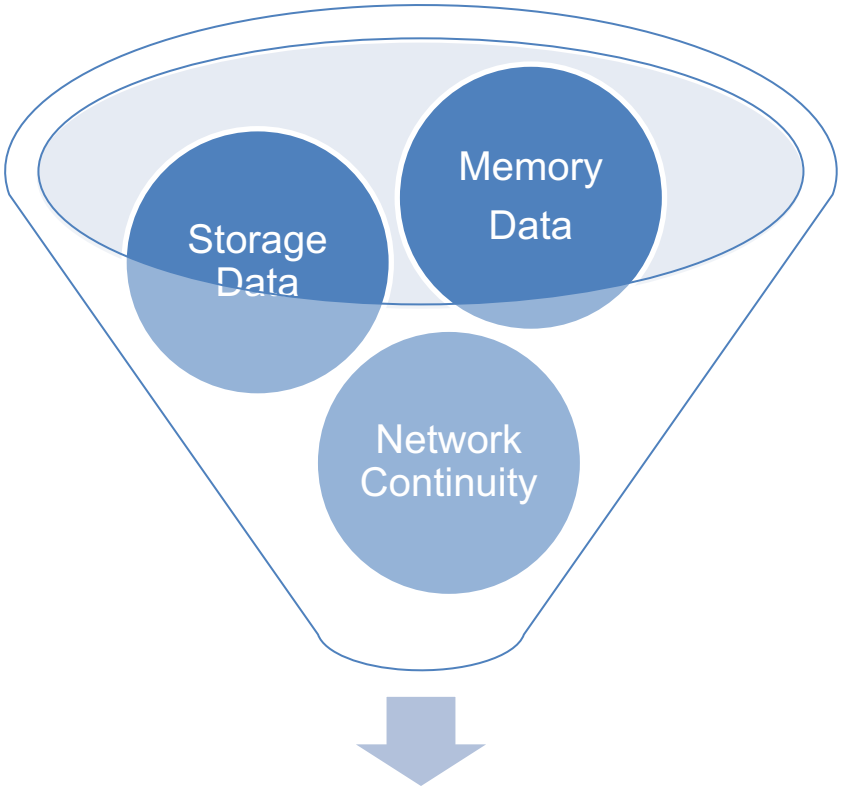
(c) Live migration over WAN



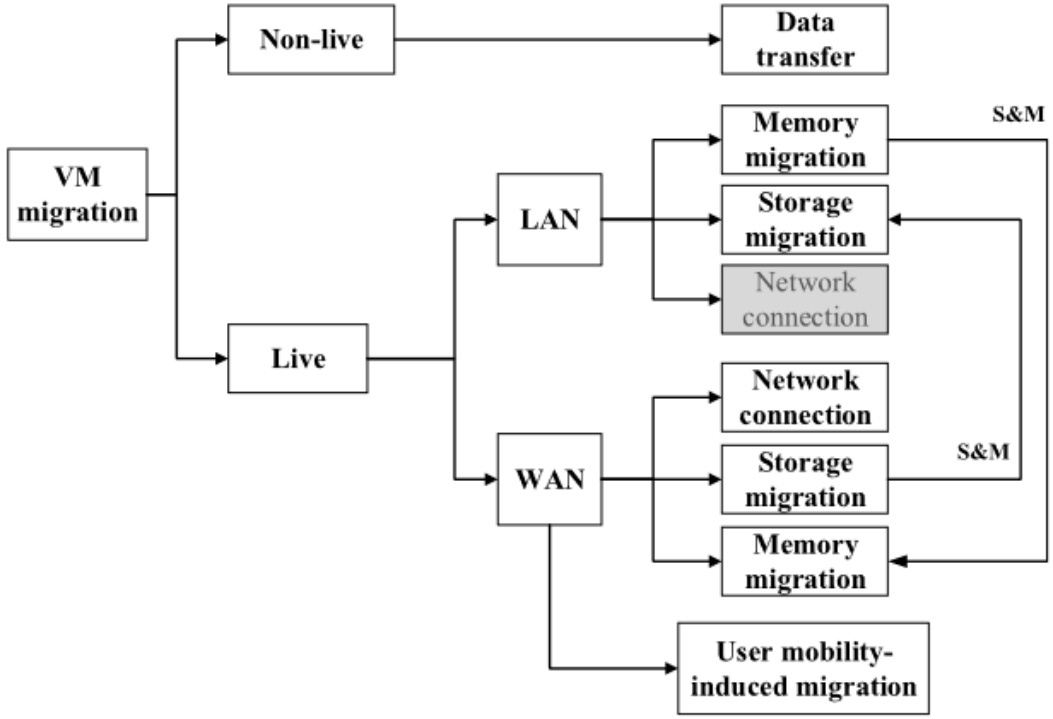
(d) Multiple migration

REF: Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, *20*(2), 1206-1243.

VM Migration Challenges



VM Migration Approaches



REF: Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, *20*(2), 1206-1243.