



BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

Architecting Storage Applications for the Public Cloud Economy

Josh Salomon & Orit Wasserman
Red Hat

jsalomon@redhat.com

owasserm@redhat.com



- Why use Storage Applications in the public cloud?
- Public cloud structure and its economy
- Tips for developing economic Storage Applications for the public cloud.

Once Upon a Time in Windows 3.x

What is the difference?

```
while (GetMessage (&uMsg, NULL, 0, 0) > 0) {  
    TranslateMessage (&uMsg);  
    DispatchMessage (&uMsg);  
}
```

```
while (WM_QUIT != uMsg.message) {  
    if (PeekMessage (&uMsg, NULL, 0, 0, PM_REMOVE) > 0) {  
        TranslateMessage (&uMsg);  
        DispatchMessage (&uMsg);  
    } else if (IsBGWork()) {  
        DoBGWork();  
    }  
}
```

GetMessage is battery manager friendly while PeekMessage is not

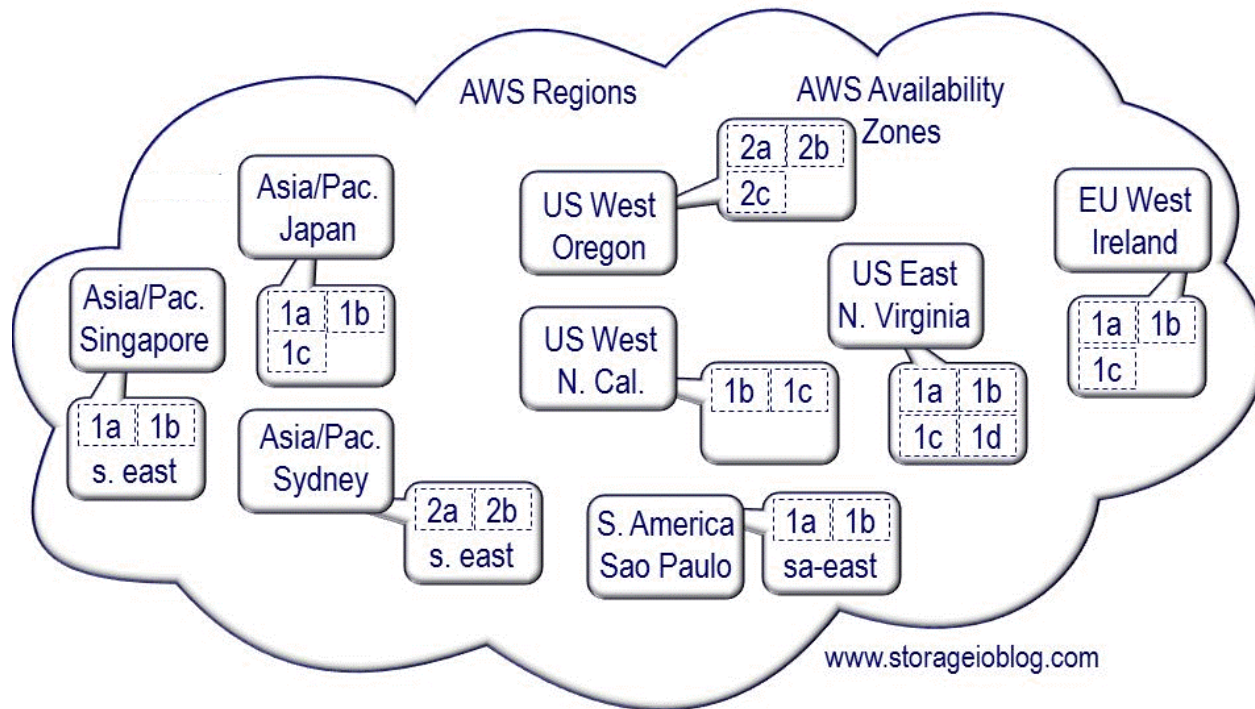
Why Do We Need Storage Applications in the Public Cloud?

- Efficient resource usage
 - EBS volumes come with fixed IOPS/size ratio
 - Other options (io1) can cost much more under heavy load
 - Cost structure: price per GB-mon + per provisioned IOPS-month
- Native cloud storage limitations
 - Multi Availability Zone (AZ) storage
 - EBS volumes are limited to 16TB.
- Thin slicing of the storage
 - Some workloads require very small capacity from the storage, using native cloud storage can cost a lot.

Public Cloud Structure

- Public Clouds
 - AWS, Azure, GCP, IBM, others
- Regions
 - Separate geographic areas for the infrastructure
 - Examples us-east (N. Virginia, Ohio), APAC (Mumbai, Seoul and 4 more), 5 European regions and more
- Availability Zones
 - Multiple data centers within the same region
 - Typically 3, but could be 2 up to 6 (today)
 - For 2 AZ region, there is no real protection from AZ failure, unless we have arbiter outside the region.

Public Cloud Structure



Examples use AWS terminology - https://storageio.com/images/SIO_AWS_Regions.gif

Public Cloud Economy

- Replace Capex with Opex
- Pay for use of *everything*
 - Compute
 - Node / VM vs. Serverless
 - Storage
 - Different storage types with various cost structures
 - Network
 - Intra-AZ network is cheaper than Inter-AZ network
 - Applications
 - Databases, Dev tool, AI tools, Replications all provided in XaaS model

Efficient public cloud applications minimize resource usage even when the resources exist.

Reducing Cloud Storage Cost

TIP 1 - Device Selection 1

Instance storage vs Cloud block storage (EBS):

- **Cheaper:**
 - For instance storage, storage price is included in the instance price
 - EBS (US East, N Virginia):
 - gp2 cost \$0.1/GB per month
 - io1/io2 cost \$0.125/GB per month + \$0.065 per provisioned IOPS-month
- **More performant:**
 - EBS gp2:
 - max 16K IOPS, 3 IOPS/GB
 - i3en*:
 - up to 2M IOPS, up to 30 IOPS/GB
- **Network BW is slow relative to device performance**
 - An opportunity and a challenge



TIP 1 - Device Selection 2

- Instance storage is less reliable
 - Not replicated
 - Not persistent through instance stops, terminations, or hardware failures but persistent across node reboots.
 - SDS systems know how to handle unreliable media !
 - This should be reflected in the reliability model



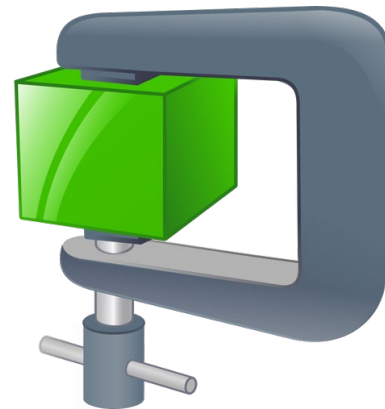
TIP 1 - Device Selection 3

- When using EBS, use it sparingly and consider using:
 - Less replicas, reduce cross AZ protection
 - Erasure Coding schemes
 - Cheaper compute nodes



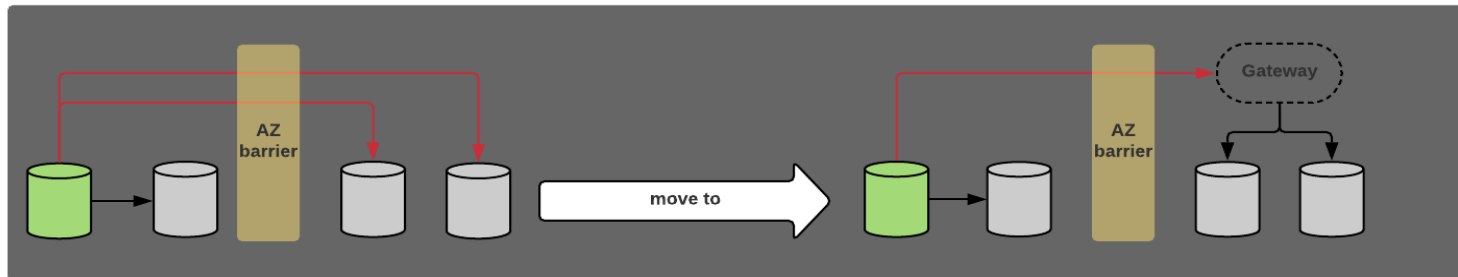
TIP 2 - Network Optimization 1

- Cross AZ network is:
 - Expensive
 - Slower
- Minimize cross AZ communications:
 - Single availability zone when possible
 - Implement AZ read affinity
 - Use compression where appropriate



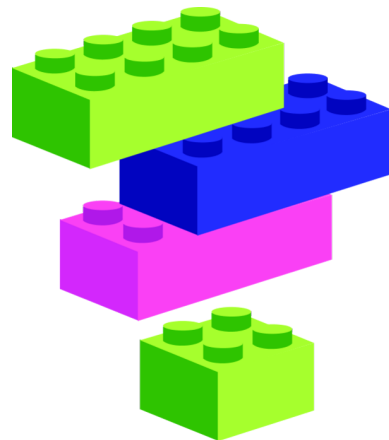
TIP 2 - Network Optimization 2

- Consider alternative topologies, for example using gateways instead of point to point communications
- Minimize the use of large inter-AZ communications.



Tip 3 - Elasticity / Best Fit Size

- Always try to find the smallest system size
 - Empty or idle system is waste of money!
- Use small building blocks:
 - Tighter scaling options
 - Optimize cost/size for your use-case and workload



Tip 4 - Use the Right Pricing Tier

- Study your cloud provider pricing model!
- Plan your capacity in advance
- Use provider savings plans
- Choose the right pricing tiers



Pricing Tier - On Demand Instance

- Pay for compute usage, by the hour or second
- No long-term commitments
- No upfront payments
- Increase/Decrease your compute capacity by demand



Pricing Tier - Reserved Instance

- From 1 to 3 years reservation
- Discount between 40% to 75%
- Example m5.4xl:
 - On demand: \$0.768 per hour
 - Reserved 1 year (upfront): \$0.452, 41% discount
 - Reserved 3 years (no upfront): \$0.332, 57% discount
 - Reserved 3 years (upfront): \$0.289, 62% discount



Pricing Tier - Spot Instance

- Uses the cloud provider spare capacity
- Cheaper but not guaranteed
 - Might be taken in case of a load with 2 minutes notice.
- Queues are maintained per instance type and availability zone
- Up to 90% discount, for example m5.4xlarge:
 - On demand: \$0.768 per hour
 - Spot instance: \$0.1862, 76% discount



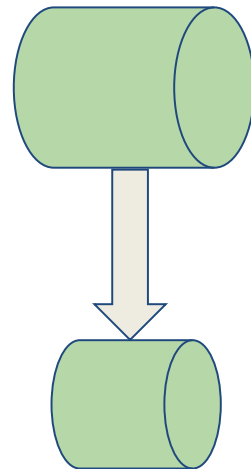
Using Spot Instances

- Ideal for stateless (or near stateless) workloads
- Perfect for driving EBS volumes
 - If the data is on EBS, the driving application is stateless relative to the hosting instance
- Your application should be able to move to a new instance almost immediately (up to 2 min)
- Flexibility in instance types
- Reserve spare instances
- Switch to “On Demand” during load or after eviction



Tip 5 - Enable Downsizing

- Lacking support from public cloud providers
 - Cannot reduce EBS volumes size
 - The sizing units are EBS volumes, or storage nodes - plan the size of these units carefully.
- Enable downsizing for your customers
 - If the customers pay per storage capacity, they will thank you!



Summary - Key Points for Cloud Systems

- Elasticity
 - While this is the obvious it is also the key for successful cloud implementation.
- Multiple Deployment Options
 - Pricing model can change without notice, system should be flexible enough to adapt quickly
- Think OPEX
 - Need to optimize on more dimensions



**Please take a moment
to rate this session.**

Your feedback matters to us.

Thank you.

jsalomon@redhat.com

owasserm@redhat.com



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHatLabs

