What a z/OS Guy Learned About AWS Over 10 Years

Scott Chapman

Enterprise Performance Strategies, Inc.

Scott.Chapman@EPStrategies.com

Contact, Copyright, and Trademarks

Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at https://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

Abstract (why you're here!)

At Enterprise Performance Strategies, we're some of the best in the world at helping people understand, improve, and manage the performance of their z/OS systems. When Scott Chapman joined EPS in 2014, he brought with him over 20 years of managing IT in a z/OS environment but then had to shift to AWS. Over the past 10 years as both EPS and the AWS offering catalog have grown, Scott has formed a few opinions and observations of AWS that may be of interest to those who have a mandate for migrating part or all their mainframe workload to the cloud. Spoiler alert: not all grass is greener on the other side of the fence.

Agenda

- Brief intro to myself, EPS, and Pivotor
- What is “the cloud”
- A few AWS services and terms
- AWS Strengths and Challenges
- Mainframe Strengths and Challenges
- Conclusions

EPS: We do z/OS performance…

- Pivotor - Reporting and analysis software and services
  - Not just reporting, but analysis-based reporting based on our expertise
- Education and instruction
  - We have taught our z/OS performance workshops all over the world
- Consulting
  - Performance war rooms: concentrated, highly productive group discussions and analysis
- Information
  - We present around the world and participate in online forums https://www.pivotor.com/content.html

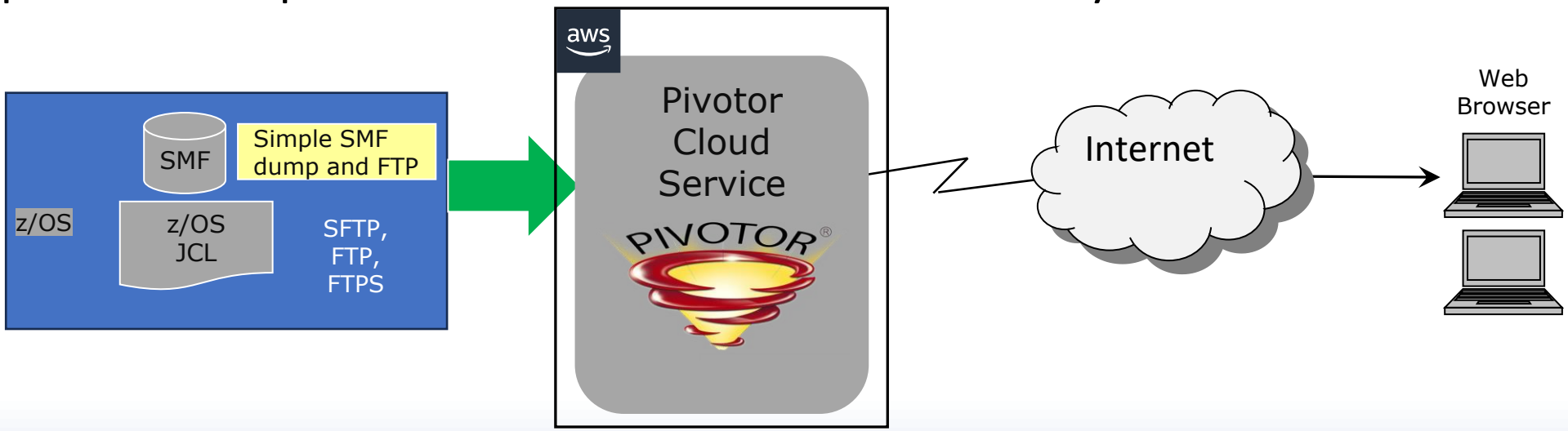
z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating Goals
  - February 19-23, 2024
- Parallel Sysplex and z/OS Performance Tuning
  - August 20-21, 2024
- Essential z/OS Performance Tuning
  - September 16-20, 2024
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Pivotor: z/OS Performance in the Cloud

- Pivotor is our data reporting tool & service designed specifically for z/OS performance reporting
  - Designed and used by z/OS performance experts
  - Processes data from SMF, DCOLLECT, and customer sources
  - Contains hundreds of z/OS performance reports “out of the box”
  - Reports are pre-generated and organized into logical and searchable report sets
  - Report sets are presented in a calendar format for easy access



Scott Chapman: Why can I talk about this?

- Spent over 20 year doing z/OS system and application performance and capacity at one of the largest power companies in the US
  - Started in applications, moved to systems
  - Experience with Parallel Sysplex and DB2 data sharing in starting in the mid-90s
  - Involved in lots of hardware upgrades and migrations over the years
  - Responsible for understanding and monitoring the IBM MLC bill
  - In short, lots of mainframe experience
- In 2014, joined Peter at EPS to help more customers with z/OS performance
  - Also, work on Pivotor development, and run the Pivotor SaaS
  - Basically took over IT when Peter's prior IT guy left
  - IT = Linux servers on AWS
    - I had some Linux experience, but had never touched AWS
  - Since then, our data processing has probably grown by ~20x
    - Learned a lot along the way!

Why this presentation

- We hear about Mainframe sites having a mandate to “move to the cloud”
  - Actually: seems to be common regardless of whether the org has a mainframe
- Many mainframers have limited practical experience with “the cloud”
- The cloud people often have limited mainframe experience
- Both groups have a lot of assumptions
- This presentation is meant to be a brief introduction to AWS to give mainframers some talking points when discussing a cloud mandate
  - And if you’re not a mainframe person maybe you’ll learn something about how mainframers think!

What is “the cloud”?

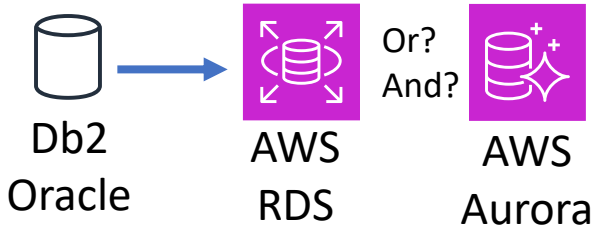
Cloud Computing Definitions

- "Someone else's computer" – not inaccurate
- Basically Outsourcing: pay someone else to run/manage part of your IT
  - Someone else's datacenter = Colocation
  - Someone else's server hardware = Infrastructure as a Service (IaaS)
  - Someone else's OS, DB, app architecture = Platform as a Service (PaaS)
  - Someone else's software = Software as a Service (SaaS)
- Primary advantage is "pay as you go" model without having to invest in the necessary components (including people) to manage each level
  - Similar to how your org probably doesn't generate its own power
  - May have larger operational expense but lower capital expense
- Critical point: if you're paying somebody else to manage something for you, they have to make a profit too
  - Does their scale let them do it cheaper than you could?

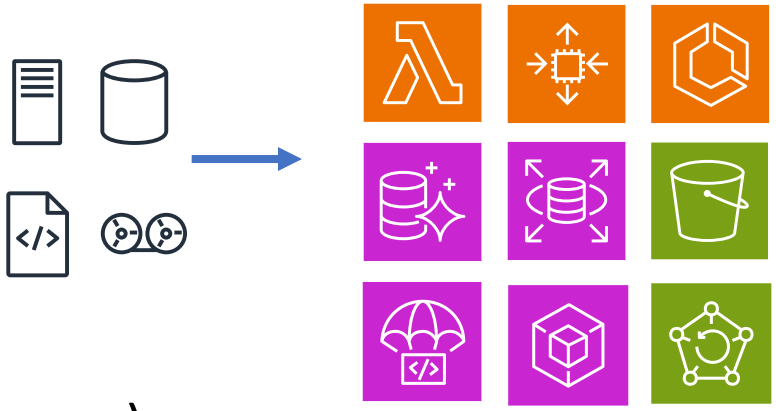
Cloud migration strategies

Re-host
Move your on-prem servers to servers in the cloud

Re-platform
Move some functions to new cloud-native service



Re-factor
Change your applications to be cloud-native



- "More Cloudy" can mean:
  - More savings (popular, but questionable, theory)
  - More vendor lock-in
  - Less control
  - More potential for hidden/unexpected dependencies

Cloud Controversies

- Cloud repatriation is a hot topic
  - 37Signals saves millions per year leaving AWS:
    - https://world.hey.com/dhh/the-big-cloud-exit-faq-20274010
  - Probably most applicable / easiest for IaaS situations
- Should you be multi-cloud?
  - Probably a bad thing for multiple reasons: more management overhead, less purchase scale/discounts, cross-cloud data transfer costs
  - Some value for specific uses: e.g. for deep backup to completely different provider
    - Google Cloud deleted a large customer's entire account across 2 geographies
    - Recovery possible because they had partial backups with another provider (maybe?)
    - https://www.networkworld.com/article/2099457/google-cloud-issue-blamed-for-unisuper-week-long-service-disruption.html
  - Expectation is that multi-account and multi-region with 1 provider is sufficient
    - Google follow-up to the above indicated backups were actually in another GCP account

AWS Services and Terms

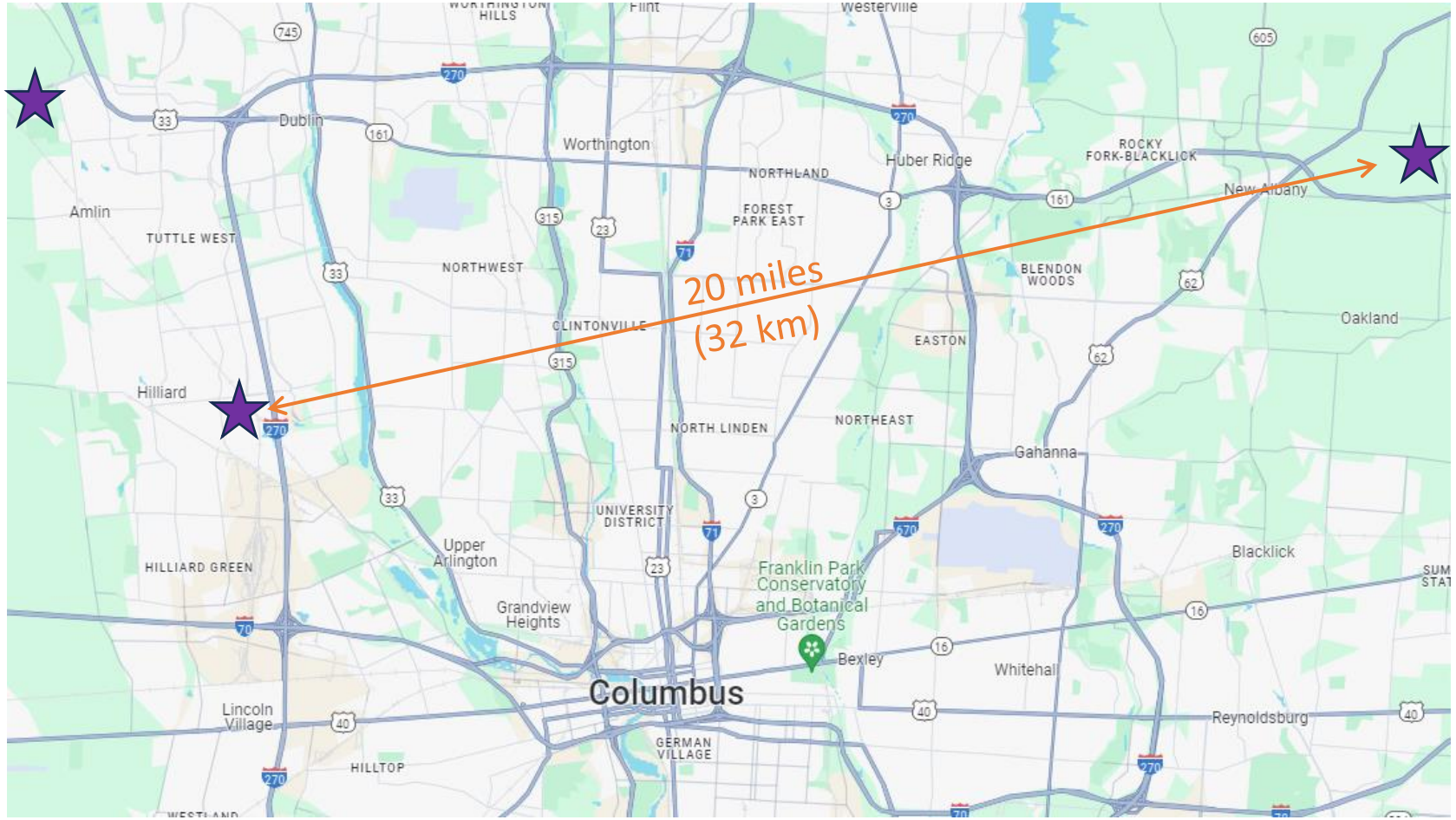
AWS Services

- There are >200 AWS services these are just some highlights
- EC2 = Elastic Cloud Compute = virtual server instances
- EBS = Elastic Block Store = virtual disk volumes (connect to EC2 instances)
- S3 = Simple Storage Service = “object” storage
  - Can looks like files in directories, but it’s really not
- VPC = Virtual Private Cloud = virtual network
- SNS = Simple Notification Service = send email/text/etc. notifications
  - Related: SES = Simple Email Service
- CloudWatch = Cloud monitoring
- CloudTrail = Cloud logging
- Lambda = “Serverless” code execution environment

AWS Terms

- Account = generally AWS “things” can’t be shared between accounts
- Organization = group of accounts bundled together for billing and control
- Instance = server
  - Will also need at least 1 EBS volume to boot from!
- AMI = Amazon Machine Image = Image of a server
- vCPU = Virtual CPU, usually just a thread on a core
  - 2 vCPU server usually means 1 core with HyperThreading
- Bucket = *Globally unique* name grouping S3 objects
- Availability Zone = Group of datacenters in close proximity of each other
- Region = Group of 2 or more AZs separated by several kms, but within 100 km of each other

US-East-2 (Ohio)



Availability Zones in Region US-East-2

AZ in US-East-2 near Hilliard

AWS Strengths

(With some caveats)

Starts at $0

- AWS services are typically priced in pennies per hour / GB / something
- Generally (not always) only pay for what you "use"
  - "In the cloud, you only pay for what you forgot to turn off"
- Many services have a free tier that allows for easy startup
  - Some services free tier is limited time (e.g. x months after starting)
- But…
- Pennies can add up quickly!
  - Remember there are ~720 hours / month so 10¢/hour = $72/month = $864/year
  - On-demand Linux Pricing, that's a 2 vCPU Intel server with 8GB of RAM
    - Does not include the EBS storage that you'll need (maybe 8¢/GB-Month)
- Services sometimes have multiple pricing dimensions
- Services sometimes have requirements for other services

Dynamic capacity

- Standing up new server or new storage can be *very* fast and easy
  - Can potentially start servers in seconds to minutes
  - EBS volumes can be grown dynamically with no downtime
- Can be very easy and cost-effective to test things
- But…
- How fast you can stand up new servers depends in part on the work you've done ahead of time
- Don't forget to turn off test environments when you're done!
- Clouds are not infinite
  - Very large requests may not be able to be fulfilled right now
  - Not all services or instance types available in all regions

S3 is pretty great

- No known instances of S3 standard storage losing data (in 18 years!)
  - 11 nines durability: data stored across multiple devices and AZs
- *Great* for backups (active use requires more care)
- Can be quite inexpensive
  - 2.3¢/GiB/Month at its most expensive storage cost ($23/TiB/month)
    - Excluding new "express" option that you probably don't want
  - .099¢/GiB/Month at its cheapest (12 hour access time, though)
- So good that there are S3 API-compatible competitors such as B2 and R2
  - And there are even ways to write your mainframe backups to S3
- But…
- Pricing complicated by per-request costs and multi-block objects
- Performance can be somewhat variable (although generally pretty good)

Unlike disk vendors, S3 priced based on binary instead of decimal gigs!
1 TiB = 1.0995 TB

Choices

- Over 200 AWS services to choose from
  - Plus myriad features/functions within a service
- Lots of ways to solve a problem
  - Multiple relational DBs in RDS (including Db2)
  - Multiple NoSQL options
  - Multiple ways to run containers
  - Lots of instance types
- But…
- Deciding between the choices can take time and effort
  - Testing is recommended!

AWS Challenges

Server scaling not granular

- AWS has *lots* of EC2 instance size options, but generally…
  - Cannot independently select vCPUs and memory
  - Three ratios of GB RAM to vCPU: 2 to 1, 4 to 1, 8 to 1
  - Within an instance family, next size up is initially 2x, then 50% larger at larger sizes
  - Price scales with capacity
- Changing instance type requires shutting down the instance
  - Probably at least partially an OS limitation
- Expectation of scale-out not scale-up
  - Cattle not pets
  - Scale out can require more management
    - Charged for what you forget to turn off!

Name	GiB RAM	vCPUs	$/hour
M7I Large	8	2	$0.10
M7I Extra Large	16	4	$0.20
M7I Double Extra Large	32	8	$0.40
M7I Quadruple Extra Large	64	16	$0.81
M7I Eight Extra Large	128	32	$1.61
M7I 12xlarge	192	48	$2.42
M7I 16xlarge	256	64	$3.23
M7I 24xlarge	384	96	$4.84
M7I 48xlarge	768	192	$9.68

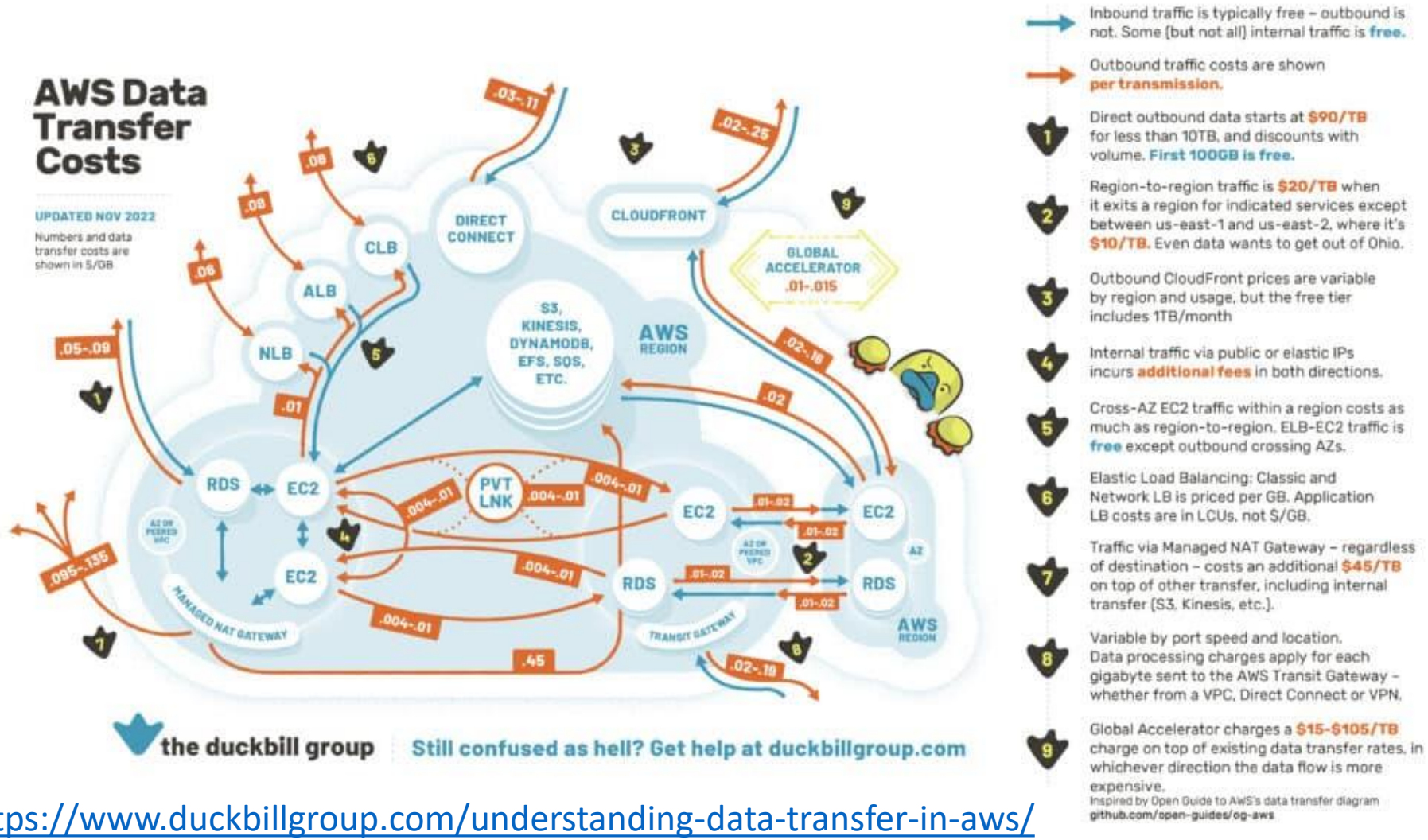
Pricing is *complicated*

- AWS saw IBM's software pricing models and said "Hold my beer"
- Many services have multiple options and dimensions
  - S3 request pricing is a 22x5 matrix with 5 footnotes/caveats
- Some pricing dimensions can't be known until you try
  - How many I/Os is your database going to do?
  - How many requests will be made to your S3 bucket?
  - Some services require other services
- Our monthly detailed invoice (all charges, all hours) is >400,000 lines
  - Yeah, I don't look at that level of detail unless I really need to…
- Use Savings Plans and/or Reserved Instances to reduce instance costs
  - But those also add an additional level of complication!
  - "Laddering" your Savings Plans so you have multiple expiring at multiple times may be a good thing
- Prices vary by region
  - This makes sense, but beware and choose wisely considering location & cost
- Data transfer costs can be significant and are convoluted too…

There are *lots* of configuration decisions (or non-decisions) that can impact your bill. Some seem quite arbitrary.

Infrastructure cost reductions almost always only for new infrastructure types, necessitating migrations.

From Cloud Economist Corey Quinn



Inbound traffic is typically free – outbound is not. Some (but not all) internal traffic is **free.**

Outbound traffic costs are shown **per transmission.**

1. Direct outbound data starts at **$90/TB** for less than 10TB, and discounts with volume. **First 100GB is free.**
2. Region-to-region traffic is **$20/TB** when it exits a region for indicated services except between us-east-1 and us-east-2, where it's **$10/TB.** Even data wants to get out of Ohio.
3. Outbound CloudFront prices are variable by region and usage, but the free tier includes 1TB/month
4. Internal traffic via public or elastic IPs incurs **additional fees** in both directions.
5. Cross-AZ EC2 traffic within a region costs as much as region-to-region. ELB-EC2 traffic is **free** except outbound crossing AZs.
6. Elastic Load Balancing: Classic and Network LB is priced per GB. Application LB costs are in LCUs, not $/GB.
7. Traffic via Managed NAT Gateway – regardless of destination – costs an additional **$45/TB** on top of other transfer, including internal transfer (S3, Kinesis, etc.).
8. Variable by port speed and location. Data processing charges apply for each gigabyte sent to the AWS Transit Gateway – whether from a VPC, Direct Connect or VPN.
9. Global Accelerator charges a **$15-$105/TB** charge on top of existing data transfer rates, in whichever direction the data flow is more expensive.

Inspired by Open Guide to AWS's data transfer diagram
github.com/open-guides/og-aws

https://www.duckbillgroup.com/understanding-data-transfer-in-aws/

Seriously, if you have AWS cost issues, these are the guys to look up!
https://www.duckbillgroup.com

Also check out their fine newsletter if you're working with AWS:
https://www.lastweekinaws.com/

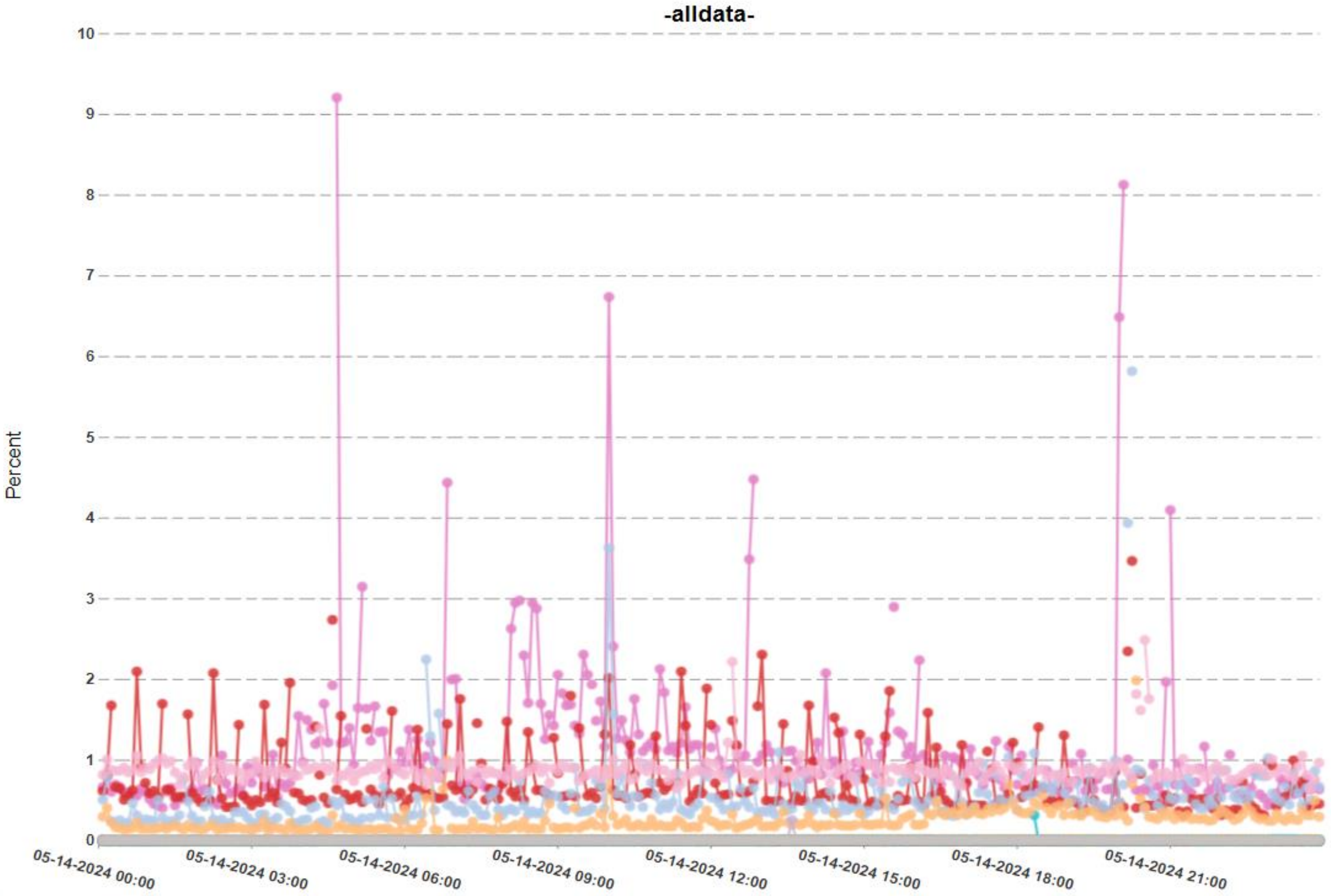
Batteries not included

- What about backup?
  - In most services, that's extra
  - In some cases, the backups can cost more than the storage on a per GiB basis
- Isn't the cloud highly available by default?
  - Nope: HA must be explicitly configured (& paid for) in most services
  - "Real" HA configs may incur additional data transfer costs too
  - Also: some AWS services may have hidden single-region dependencies
- Did you want support with that?
  - That will be extra 3-10% extra on top of your bill
- Do you want an account manager?
  - I hear you might get one if you're spending $1M/year

Performance can be variable

- Things listing “burst” performance especially prone to variability
  - Certain EBS volume types list steady state and burst performance service levels
  - Certain instance types can burst to their capacity for only a limited number of hours
  - Be wary of running out of your burst “bucket”: impact can be significant!
- EBS volumes are also network-attached storage
  - Can see some variation because of that
  - Can pay for better performance (can get expensive)
  - Altering a volume may result in impact for some time
    - May also take quite some time on larger volumes before you can issue another change
- CPU performance fairly stable if you’re not using a burstable instance
  - Spectre/Meltdown issue in 2018 was an exception to that
- Other services also have a “pay for more performance” model

Server CPU Steal%



"Noisy Neighbors" can be a problem for the burstable instances.

Steal time is percent of time spent waiting to get on a physical CPU.

Not generally seen for instance types that don't have "burst" capacity.

Burstable instances savings may not be worth the potential variability in service.

Mainframe Strengths:
Things I wish I had from z/OS

Some of these are due Linux limitations more so than AWS limitations

JES(!)

- Most of what we do is batch-oriented and z/OS excels at batch management
- There are (now) some AWS services that can help with batch management
  - Really is managing containerized application executions or things like Lambda
  - May be “fine”, but I haven’t really looked at them in detail
- We use the open source Rundeck as our scheduler to run jobs on servers
  - Our servers are closer to pets than cattle…
  - Except for our “eagles” which fly in, do some work, then leave ☺
- None of the above have the decades of experience JES does
  - JCL may be ugly, but it works!
  - And JES has a lot of options for managing work, including WLM-managed inits
  - And Enterprise schedulers have almost as much experience as JES

Db2

- Best database in the world: Db2 on Parallel Sysplex!
- A lot of the large “scalable” database solutions usually have compromises:
  - Scaling for reads but not writes
  - Poor insert/update/delete performance (see above)
  - Eventual consistency
  - Lack of actionable/useful performance and management metrics
    - Q: “So how do we know when a table should be reorged?” A: “When it seems slow”
  - Rudimentary query optimization
  - Deviations from ANSI SQL norms
- Those other DBs do of course work well enough for many businesses
  - Especially if the workload is predominantly read
  - But “well enough” may require some significant tuning effort
- Migrating between DBs can be far harder than OS migrations!

WLM (of course!)

- Workload Manager is an awesome piece of technology
- Much, much more difficult to manage disparate workloads without it!
- General mitigation is to run at low utilizations
  - Also, separate workloads to different instances
  - Which of course means more instances of the OS to manage, patch, backup, etc.

Vertical Scalability

- Running “everything” in one system does simplify things!
  - Network latency within an AZ is low, but is not cross-memory fast
  - And can be variable…
- AWS does have very large instances, but due to OS limitations will probably only be able to run one type of work there
  - E.G. you could have a very large DB server, but probably wouldn’t run the application and web server on the same OS (at least at significant size)
  - No WLM to manage the work!
  - There may be some scaling issues on those larger instances
    - I haven’t really explored this, but I have some questions about Linux with large CP counts
- Shifting work with Parallel Sysplex makes maintenance easier
  - To be fair, the container orchestrators apparently can do this well enough too
  - But doing this for always-on servers gets complicated

Data replication

- Mainframe data replication technologies are exceptional!
- Yes, they can be complicated and expensive, but for the most part they work without getting in the way
- You can build *somewhat* similar capabilities on some file systems in Linux
  - But being consistently only a few seconds behind is probably not realistic
- Building out a full DR solution was way more complicated in AWS
  - Popular pattern today seems to avoid DR in favor of globally distributed applications
    - May still have “control plane” single points of failure
      - AWS themselves have experienced this, causing widespread issues
    - Probably are also relying on eventual consistency, which may not fit all use cases

Enqueues

- The idea that the OS should only allow one process to update a data set at a time is an under-rated capability of z/OS
  - "We'll never have two processes updating that file at the exact same time" will eventually be wrong
- You can build a locking mechnism, but watch out for the sharp edges!

SMF

- There are a lot of metrics available in Linux and AWS, but … not as broad and detailed as we have in SMF
  - Nor as easy to record in a centralized fashion
- Why was this system busy at this time is a lot easier to understand from the z/OS metrics
  - In effect, we can get from Linux data similar to *some* of the data in:
    - SMF 70, 74, 100, 113, 119, 120
  - I wish I could get data similar to:
    - SMF 30, 42, 72, 98, 99, 101
  - Some of this can be mitigated by analyzing application logs, but you have to build processes for that

What are the mainframe's challenges?

Relative to "the cloud"

Does not start at $0 (and goes too high)

- High initial costs makes it hard to attract small organizations to use z/OS
  - An idea and a credit card is not enough to start your business on z/OS
  - A new biz that has built on the cloud is unlikely to be convinced to switch to z/OS
- Existing customers can struggle with the software costs
  - Especially small to mid-size customers
  - No other platform jumps through as many hoops to manage software costs
- Ultimately, the mainframe can be cost competitive for large workloads
  - The comparison is not a pickup truck vs. a tractor trailer
  - The comparison is a box truck fleet vs. freight train
  - But if you're not hauling that much freight… those box trucks may be cheaper

Scalability is less dynamic

- The mainframe does have some scalability advantages
  - I.E. you can add more CPs to z/OS without an IPL as long as you defined reserved CPs
- But ultimately your scalability is limited to the hardware on the floor
  - Bringing in additional hardware is time consuming
  - Capacity Planning is much more important than in the cloud!
- Ability to spin up a server and a few TBs of disk space to test something out for a day at the total cost of a few dollars is nice!

That's it for the Mainframe's challenges

- What about lack of people?
  The experience isn't that much different:
  - It's just a (really great) technology to learn, like AWS or GCP or Linux or K8s or …
  - z/OSMF provides a web interface for an increasing amount of admin work
  - You can SSH into the mainframe just like any Linux server
  - More and more open source tools being ported to the mainframe https://zosopentools.org/#/
  - Pay people and they will come (but admittedly maybe not in droves live Java programmers)
- Isn't the hardware expensive?
  - Yes, but it also has a quality of service that may justify at least some of that cost
- But our application is written in COBOL and nobody can understand it!
  - In the 1950s programmers were worried that they would be out of work because COBOL would allow business people to write code! It's not that hard…

In conclusion…

If you want to learn / play with AWS

- Playing can be pretty cheap, but it can also balloon if you're not careful
  - Check and understand the pricing before using anything
    - Watch your daily costs in Cost Explorer but beware that costs *can* take 24-48 hours to appear
  - Know the limits for any free tier usage
  - Set up billing alerts
  - If an egregious surprise happens, reach out to AWS Billing Support
    - Unlike technical support, billing support is included
- If you're playing with an EC2 server, stop it when not playing
  - Can restart in seconds
  - Stops the EC2 charges, but EBS charges will remain

Things I wish were available (or I knew about) 10 years ago

- Things to do from the beginning (for a real business):
  - Set up an Organization and use SCPs (Service Control Policies) to enforce security standards over the accounts within the org
  - Set up SSO so users can use one logon across multiple accounts
  - Use Infrastructure As Code
    - Allows for more control, checking, change management, etc.
  - Come up with (and enforce) a tagging strategy/structure
    - Then set up billing reporting based on those tags
    - Absent this, understanding what is driving your costs will be more… interesting
- If you only are planning on doing IaaS, price out colocation and related services

Is the cloud the end of the mainframe?

- No: Client-server every other “threat” wasn’t the end of the mainframe
- No: Moving to the cloud means abandoning several mainframe strengths
- Yes: Starting at $0 captures all new / startup businesses
  - If businesses leave the mainframe and no new ones replace them… there will be a long slow decline in terms of number of mainframe customers
  - In the meantime (decades?), enjoy the world’s best computing platform… IBM Z

Questions?