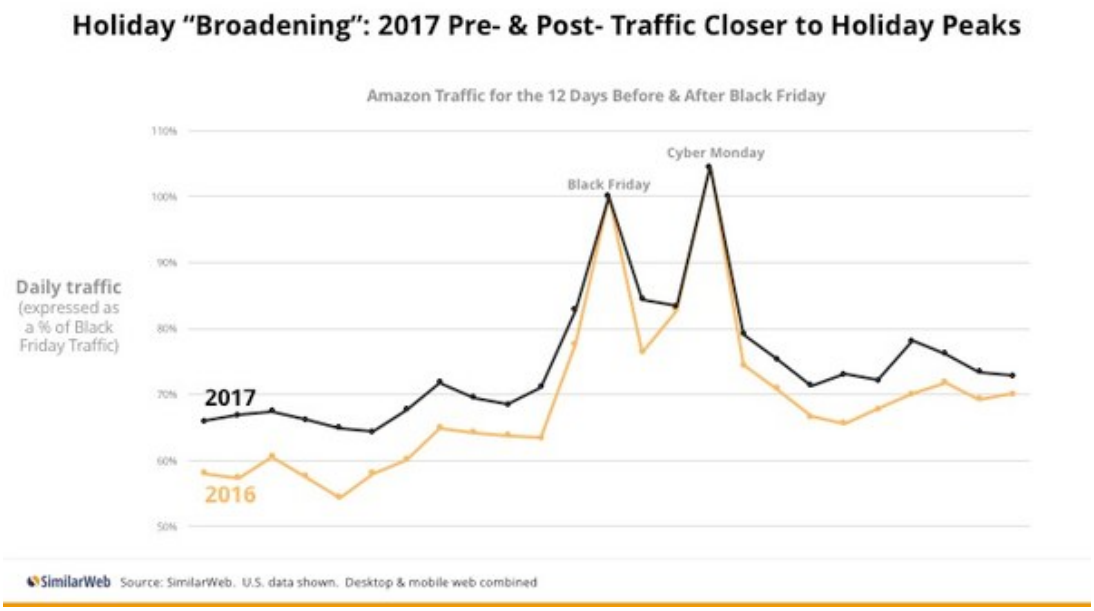
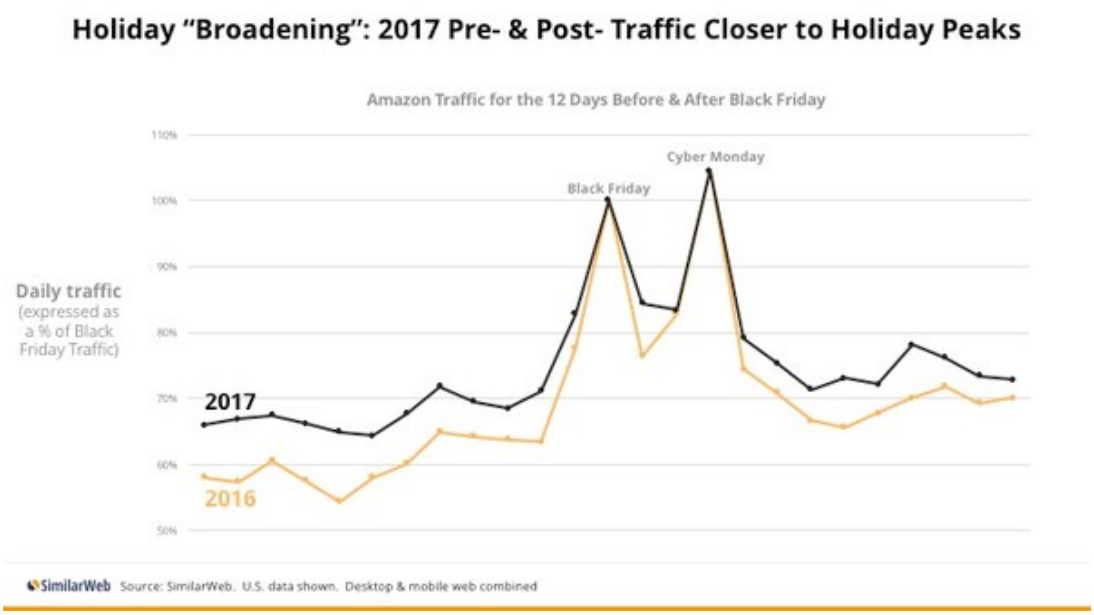
Filesystems for Cloud Services

Amazon Holiday Traffic



https://www.mediapost.com/publications/article/312409/from-cyber-monday-to-cyber-month-the-broadening-o.htm

Amazon Holiday Traffic



This is only a 12-day outlook! The peak is likely much higher compared to March traffic

https://www.mediapost.com/publications/article/312409/from-cyber-monday-to-cyber-month-the-broadening-o.htm

Amazon Web Services

https://docs.aws.amazon.com/aws-technical-content/latest/jenkins-on-aws/images/current-aws-global-infrastructure.png

Amazon Web Services

EC2

Compute services

- Amazon maintains many thousands of servers. Each server hosts many virtual machines
- You can sign up for EC2 and rent virtual machines with a certain number of CPU cores and a certain amount of memory

Amazon Web Services

EC2

Compute services

EBS

Block storage

(like a local filesystem, but accessed over a network)

- Amazon maintains large network of storage arrays
- Disk arrays are networked so that even if one array fails, the system will stay up
- You can mount any EBS volume from any EC2 instance in the same datacenter
- The EBS volume appears as if it's a normal hard drive. An EBS volume can only be mounted to one EC2 instance at a time

Amazon Web Services

EC2

Compute services

EBS

Block storage

(like a local filesystem, but accessed over a network)

S3

Object storage

(sort of like Google Drive)

Amazon Web Services

EC2

Compute services

EBS

Block storage

(like a local filesystem, but accessed over a network)

S3

Object storage

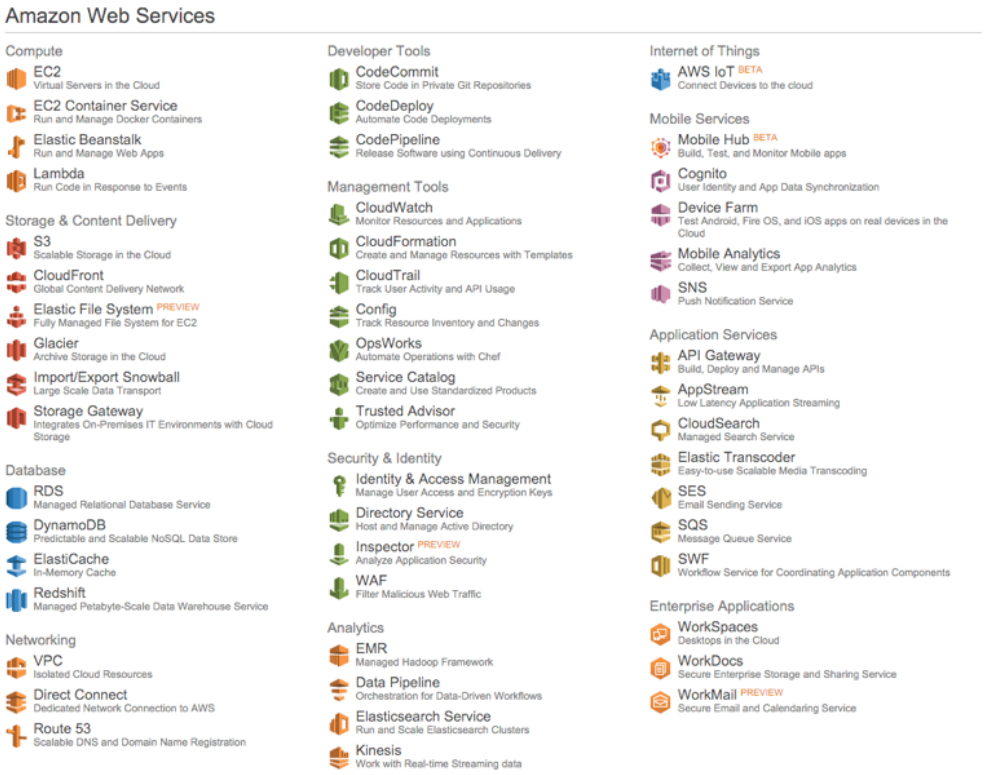
(sort of like Google Drive)

Glacier

Archive storage

(like S3, but cheap and glacially slow)

Amazon Web Services



https://codentrick.com/aws-amazon-web-services-overview/

Amazon Web Services

- Estimated 1.3 million servers[1] in 68 datacenters[2]
- Custom routers. 100 Gbps interconnects between data centers, 25Gbps connections to each server
- Custom server design, custom motherboard chipsets, custom GPUs and FPGAs
- Custom storage servers. Each rack contains 1110 hard drives, 8.8 petabytes of storage

1: https://www.zdnet.com/article/aws-cloud-computing-ops-data-centers-1-3-million-servers-creating-efficiency-flywheel/
2: https://www.forbes.com/sites/johnsonpierr/2017/06/15/with-the-public-clouds-of-amazon-microsoft-and-google-big-data-is-the-proverbial-big-deal/

Benefits of "cloud computing"

- **Benefits to AWS users:**
  - No huge up-front infrastructure investment
  - No need to hire dedicated systems administrators
  - Stability benefits of globally distributed infrastructure
  - Flexibility in handling load… Pay only for what you need and avoid getting slammed in a high-load event
- **Benefits to Amazon:**
  - Rent out unused storage capacity, make lots of money
  - Infrastructure investments benefit Amazon as well
  - $$$$$$$$$

Amazon earnings report

AMAZON.COM, INC.
Segment Information
(in millions)

	Three Months Ended December 31,		Twelve Months Ended December 31,	
	2016	2017	2016	2017
	(unaudited)			
North America				
Net sales	$26,240	$37,302	$ 79,785	$106,110
Operating expenses	25,424	35,610	77,424	103,273
Operating income	$ 816	$ 1,692	$ 2,361	$ 2,837
International				
Net sales	$13,965	$18,038	$ 43,983	$ 54,297
Operating expenses	14,452	18,957	45,266	57,359
Operating income (loss)	$ (487)	$ (919)	$ (1,283)	$ (3,062)
AWS				
Net sales	$ 3,536	$ 5,113	$ 12,219	$ 17,459
Operating expenses	2,610	3,759	9,111	13,128
Operating income	$ 926	$ 1,354	$ 3,108	$ 4,331

https://www.zdnet.com/article/all-of-amazons-2017-operating-income-comes-from-aws/

Amazon earnings report

AMAZON.COM, INC.
Segment Information
(in millions)

	Three Months Ended December 31,		Twelve Months Ended December 31,	
	2016	2017	2016	2017
	(unaudited)			
North America				
Net sales	$26,240	$37,302	$ 79,785	$106,110
Operating expenses	25,424	35,610	77,424	103,273
Operating income	$ 816	$ 1,692	$ 2,361	$ 2,837
International				
Net sales	$13,965	$18,038	$ 43,983	$ 54,297
Operating expenses	14,452	18,957	45,266	57,359
Operating income (loss)	$ (487)	$ (919)	$ (1,283)	$ (3,062)
AWS				
Net sales	$ 3,536	$ 5,113	$ 12,219	$ 17,459
Operating expenses	2,610	3,759	9,111	13,128
Operating income	$ 926	$ 1,354	$ 3,108	$ 4,331

https://www.zdnet.com/article/all-of-amazons-2017-operating-income-comes-from-aws/

Users of AWS

Adobe, Airbnb, Alcatel-Lucent, AOL, Acquia, AdRoll, AEG, Alert Logic, Autodesk, Bitdefender, BMW, British Gas, Canon, Capital One, Channel 4, Chef, Citrix, Coinbase, Comcast, Coursera, Docker, Dow Jones, European Space Agency, Financial Times, FINRA, General Electric, GoSquared, Guardian News & Media, Harvard Medical School, Hearst Corporation, Hitachi, HTC, IMDb, International Centre for Radio Astronomy Research, International Civil Aviation Organization, ITV, iZettle, Johnson & Johnson, JustGiving, JWT, Kaplan, Kellogg's, Lamborghini, Lonely Planet, Lyft, Made.com, McDonalds, NASA, NASDAQ OMX, National Rail Enquiries, National Trust, **Netflix**, News International, News UK, Nokia, Nordstrom, Novartis, Pfizer, Philips, Pinterest, Quantas, Sage, Samsung, SAP, Schneider Electric, Scribd, Securitas Direct, Siemens, Slack, Sony, **SoundCloud**, **Spotify**, Square Enix, Tata Motors, The Weather Company, Ticketmaster, Time Inc., Trainline, Ubisoft, UCAS, Unilever, US Department of State, USDA Food and Nutrition Service, UK Ministry of Justice, Vodafone Italy, WeTransfer, WIX, Xiaomi, Yelp, Zynga, more……

If we were to rethink filesystems built for cloud services, what would they look like?

Cloud-Native File Systems

Remzi H. Arpaci-Dusseau
Andrea C. Arpaci-Dusseau
University of Wisconsin-Madison

Venkat Venkataramani
Rockset, Inc.

How And What We Build Is Always Changing

Earliest days

- Assembly programming on single machines

Big single-machine advances

- Unix: A standard (and good) OS!
- C: A systems language!

Same thing, one level up: Distributed systems

- Collect group of standard machines, build something interesting on top of them

Commonality: **New System on Fixed Substrate**

Whether a single machine/distributed, we tend to build **new systems** on a **fixed set of resources** with **fixed (sunk) cost**

- Machine: X CPUs, Y GB memory, Z TB storage
- Buy many such machines
- Build new system of interest on those machines

But the world is changing…

Welcome To Cloud

Cloud is a reality

- Can **rent** cycles or bytes as needed
- Per-unit **cost** is defined and known
- Not just raw resources: **services** too

Many new systems are being realized only in cloud

- Excellent example: **Snowflake** elastic warehouse [sigmod '16]

Thus, Questions

Cloud-native thinking:
How should we build systems given the cloud?

- What new opportunities are available?
- What new systems can we realize?
- What can we stop worrying about?

In This Talk

Cloud-native principles

- Guidelines for how to think about building systems in the era of the cloud

Cloud-native file system

- Case study: How to transform a local file system into a cloud-native one

Principles

Storage principles

CPU principles

Overarching principle

(just highlights; more in paper)

Storage Reliability

Storage reliability principle:
Highly replicated, reliable, and available storage can (should?) be used (The "S3" principle)

- 11 "9s" of durability!

Implication: Build on top of this, don't build YARSS (Yet Another Replicated Storage System)

- Example (kind of): BigTable on GFS

Storage Cost and Capacity

Storage cost principle:
Storage space is generally inexpensive

- At cheapest, $4 / month / TB

Storage capacity principle:
A lot of storage space available

- "The total volume of data and number of objects you can store are unlimited" (Amazon)

Implication: Use space as needed to improve system

- Example: Indices for added lookup performance

Storage Hierarchy

Storage hierarchy principle: Storage is available in many forms, with noticeable differences in performance and cost across each level

- Example: Amazon Glacier vs S3

Implication: Must manage data across levels

- Can improve performance, reduce costs

CPU Parallelism

CPU parallelism principle (or A x B = B x A): It should cost roughly the same to execute on A CPUs for B seconds as it does to execute on B CPUs for A seconds

- Granularity of accounting might limit you…

Implication: Do everything you can in parallel

CPU Capacity

CPU capacity principle:
Large numbers of CPUs are available

- As with storage, essentially “unlimited”

Implication: Use as many CPUs as you need

- Scale up to solve tasks quickly

CPU Scale-Up/Down

CPU scale-up/scale-down principle:
One should only use as many CPUs as needed for a task, and not more

- While cheap, CPUs are not free either

Implication: Must monitor usage, turn off CPUS when unused

CPU Remote Work

CPU remote-work principle:
When possible, use remote CPU resources
to do needed work

- Shared data store makes this easier

Implication: Can separate foreground/background

- Improve predictability of former,
  use parallelism for latter

CPU Hierarchy

CPU hierarchy principle: CPU is available in different forms, with differences in performance, cost, and reliability across each level

- Normal vs. spot instance for example

Implication: CPU types must be managed

- Pick CPU right for given task

Overarching Principle

Overall performance/cost principle:
Every decision in cloud-native systems is ultimately driven by a cost/performance trade-off

- Can't make decisions without cost/perf knowledge
- Extremes are interesting:
  highest performance, or lowest cost
- But middle ground is important too:
  "reasonable" cost/performance

Implication: Cost must be fundamental part of systems (and even applications above)

Implications

Replicated storage: Don't reinvent the wheel

Extra space is cheap: Use for performance?

Massive parallelism: Use for background tasks

Hierarchy: Continuous data migration to lower cost while keeping performance high?

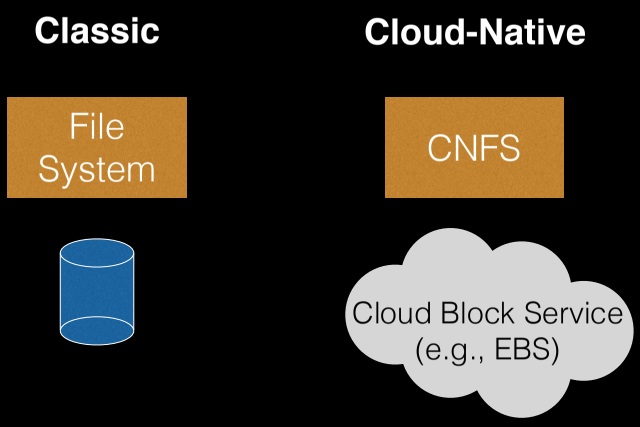
Cost: Have to know how much is OK to spend

Overall: Proper utilization of the cloud requires rethinking of how we build the systems above them

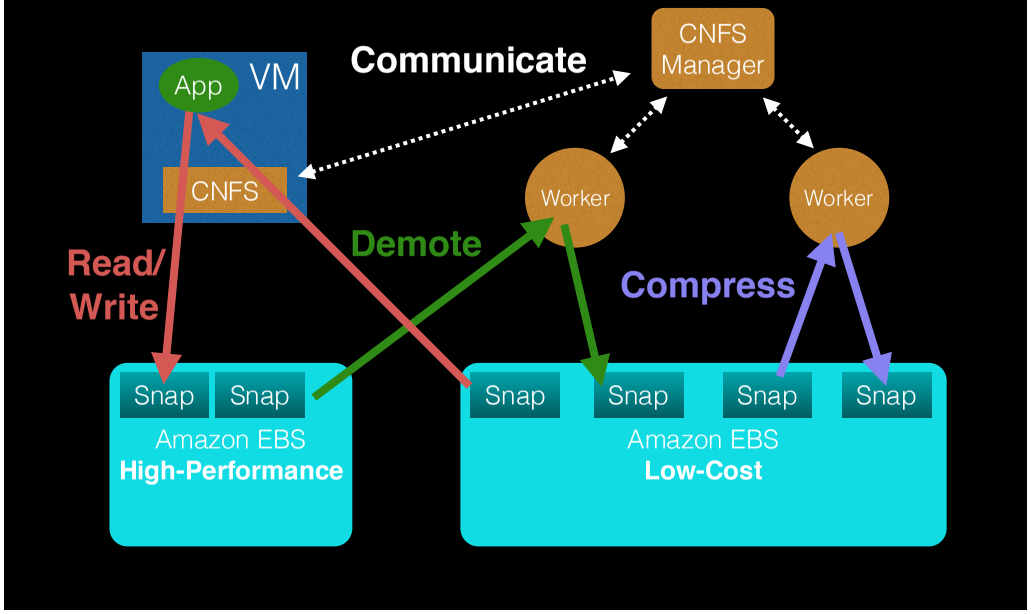
Case Study: CNFS

Case Study: CNFS

Case Study: **Cloud-Native File System (CNFS)**



CNFS Architecture



CNFS: Key Points

Copy-on-write (**COW**): Natural fit for cloud

- Enables background work on immutable storage

Storage work naturally offloaded from front end

- Enables predictable low-latency for foreground
- Adds massive parallelism for background

Can optimize for cost or performance or mix

- Need hints from above on what is important
- New APIs too

But, still needs help from cloud providers

- Example: Can't access EBS volumes from many clients (now)

Conclusions

Cloud Native

- New way to build systems upon substrate provided by Cloud

Principles: New guidelines for design

- **Higher-level services:** Don't reinvent the wheel
- **Flexible resources:** Can use a lot or a little
- **Different types of resources:** Costly/Fast vs. Cheap/Slow
- **Cost awareness:** Nothing is free

Case study: **CNFS**

- A local COW file system built to run on EBS (not a disk)
- Early prototype: Modified ext4 can migrate files across cloud volumes (but much still to be done)

Cloud-native thinking: How does it change your next system?

End