User-Defined Cloud

Yiying Zhang, Ardalan Amiri Sani, Guoqing Harry Xu
$623 billion market by 2023
15 years of evolvement  =>  175 services, 331 EC2 instance types
Lifecycle of Cloud Services

1. Provider identifies a new type of application/hardware
2. Provider develops/adapts software/hardware infrastructure
3. Provider launches a new service or extends an existing one

Problems?
An Example in the Health Industry

A hospital trying to move its digital data and computation to the cloud
Need to securely store medical records and images AND securely process them on accelerators, on-demand

Need to use different consistency levels and replication factors for different parts of the application

Today’s cloud:

- No on-demand (serverless) accelerator: pay extra with VMs
- No secure accelerator: use secure CPU and run slower
- No fine-grained tuning of systems feature: build your own
No right service for niche applications

Users have to build a local cluster or use a third-party service
Cloud providers “define” the cloud to accommodate the user needs they deem popular
Users know what is needed to run their workloads and should be allowed to get what they need.
Idea

Users define the computing resources and features of these resources to run their workloads.

Cloud providers supply software and hardware infrastructures under the hood.
User-Defined Cloud
- Giving control to cloud users and keeping management for cloud providers

Application Semantics
- Hardware Resource Specification
- Execution Environment & Security Specification
- Distributed Semantics Specification

System Software
- Hardware Platform

User Defined

Cloud Managed

Application Developer Team
IT Team
What can users define?
New images taken

Medical Images

Image Preprocessing

Image Classification

Patient Medical Records

Natural Language Processing

Anonymized Patient Medical Records

Staff Information

Geospatial data processing

Predict Hospital Staffing Plan

Decide Patient Acceptance

Emergencies
Hardware Resource Specification

New images taken

- **Medical Images**
  - Image Preprocessing
    - FPGA
  - Image Classification
    - GPU
  - Diagnosing
    - CPU

- **Patient Medical Records**
  - Natural Language Processing
    - GPU
  - Anonymized Patient Medical Records
    - SSD

- **Staff Information**
  - Anonymized Patient Medical Records
    - SSD

- **Weather and Environment**
  - Geospatial data processing
    - GPU
  - Predict Hospital Staffing Plan
    - CPU

- **Decide Patient Acceptance**
  - CPU

- **Emergencies**
Execution Environment & Security Specification

New images taken

Medical Images → SSD

Image Preprocessing → FPGA (single tenant)

Image Classification → GPU (single tenant)

Diagnosing → CPU

Patient Medical Records → HDD+SSD

Natural Language Processing → GPU

Anonymized Patient Medical Records → TEE

Staff Information → HDD

Geospatial data processing → GPU

Predict Hospital Staffing Plan → CPU

Weather and Environment → HDD

Decide Patient Acceptance → CPU

Emergencies
Distributed Semantics Specification

New images taken

Medical Images
- SSD
- 2x Rep
- FPGA
- single tenant

Image Preprocessing
- GPU
- single tenant

Image Classification
- Checkpoint

Patient Medical Records
- HDD+SSD
- 3x Rep
- GPU

Natural Language Processing
- TEE

Anonymized Patient Medical Records
- SSD

Staff Information
- HDD

Weather and Environment
- HDD
- Weak Consistency

Geospatial data processing
- GPU

Predict Hospital Staffing Plan
- CPU

Decide Patient Acceptance
- CPU

Diagnosing
- Checkpoint
-Sequential Consistency

Checkpoint

FPGA

GPU

SSD

HDD

HDD+SSD
How to achieve user definitions?
Fine-grained hardware and software building blocks put together on demand like Lego pieces
Hardware Resource Disaggregation:

Breaking monolithic servers into distributed, network-attached hardware components
Distributed Semantics Specification

New images taken

- Medical Images
  - SSD
  - 2x Rep
  - FPGA
  - single tenant
  - Image Preprocessing
  - GPU
  - single tenant
  - Image Classification
  - Checkpoint

- Patient Medical Records
  - HDD+SSD
  - 3x Rep
  - GPU
  - Natural Language Processing
  - TEE
  - Checkpoint
  - Sequential Consistency
  - Diagnosing
  - CPU
  - TEE + single tenant

- Anonymized Patient Medical Records
  - SSD

- Staff Information
  - HDD
  - TEE
  - Checkpoint
  - CPU

- Weather and Environment
  - HDD
  - Weak Consistency
  - GPU
  - Geospatial data processing
  - CPU
  - Emergencies
  - CPU

- Predict Hospital Staffing Plan
  - CPU

- Decide Patient Acceptance
  - CPU
<table>
<thead>
<tr>
<th>Possibilities</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAG of modules, Actor model, Autoscaled modules</td>
<td>Finding the scope of modules, Properly handle module dependency</td>
</tr>
<tr>
<td>Hardware resource disaggregation</td>
<td>Integration with other cloud infra Performance, Scalability</td>
</tr>
<tr>
<td>Virtualization environments, TEEs, Data encryption</td>
<td>Verifying user definitions are met Adapting virtualization/security techniques to disaggregation</td>
</tr>
<tr>
<td>APIs and program annotation with dist impl underneath</td>
<td>Conflicting user specifications Fine-grained distributed system</td>
</tr>
</tbody>
</table>

**Application Semantics**

**Hardware Resource**

**Execution Environment & Security**

**Distributed Semantics**
Benefits of User-Defined Cloud
Users:

• Customize a public cloud for exactly what they need
• No need to wait for providers to prepare new features
  ➡ Fast time to market, better performance
• No need to build or manage any software/hardware infrastructures
• Pay only for what is used
  ➡ Low IT cost

Providers:

• Build one set of configurable software/hardware infrastructure
  ➡ Cheaper to develop and maintain than today’s sea of services
• Attract niche users
• Charge a higher unit resource price
  ➡ More profit
Conclusion

User-Defined Cloud: Giving control to users

• Seemingly appealing for cloud users
• Seemingly feasible to build

➡ Time for building one

• How appealing is it for cloud providers?
• Is it worth building a whole new set of infrastructures?

➡ Reasons for academia to try it out first
Sky Computing

Ion Stoica
2022
This talk

What is Sky?

Why and how (we believe) Sky will happen?

What if Sky happens?
What is Sky?

Internet for clouds
Internet abstracts away different networks.
Sky Example

ML Pipeline

Data proc → Training → Serving

Sky

Requirement
- Need process confidential data and remove PII (personal identifiable information)

Preference
- Want to minimize cost

Sky abstracts away different clouds
Sky Example

ML Pipeline

- Use Azure Confidential Computing (ACC) for secure data processing
- Use Google Cloud for training on TPUs
- Use AWS for serving on Inferentia
Cloud Computing $\rightarrow$ Sky Computing

**Abstract away clouds:** An application runs on multiple clouds transparently using services in the compatibility set.

**Set of public services,** possibly implemented by multiple clouds.

**Compatibility Set**
- Cloud A
- Cloud B

**Intercloud Broker**

Apps

(Free) peering
How is **Sky** different from “multi-cloud”?
Today’s multi-cloud

Partitioned multi-cloud

- different apps from different teams on different clouds

Enterprise

- Synapse
- Vertex AI
- Redshift
Today’s multi-cloud

**Portable multi-cloud**
- Same app on different clouds
- Not cloud transparent: first pick cloud, then region
Today’s multi-cloud

**Portable** multi-cloud
- Same app on different clouds
- Not cloud transparent: first pick cloud, then region

**Uniform** layer
- Running on all clouds
- Typically, low-level (e.g., k8s, docker)
Sky Computing

**Transparent** multi-cloud
- Abstract away clouds
Sky Computing

**Transparent** multi-cloud
- Abstract away clouds

Compatibility Set
- *Heterogeneous*: different clouds support different sets of services
- *All* layers of software stack
Compatibility Set

Kubernetes: every cloud provides a hosted version, e.g.
- AKS (Azure), GKS (GCP), EKS (AWS)
Compatibility Set

Third-parties providing multi-cloud services (open-source):

- Databricks (Apache Spark), Confluent (Apache Kafka)
Compatibility Set

Includes proprietary services supported by a single cloud, e.g.,

- Azure ML, BigQuery, SageMaker
Intercloud Broker: Two-sided market

Match app demands & preferences to cloud services

Application 1 → Intercloud Broker → Application 2

Compatibility Set
Cloud A

Compatibility Set
Cloud B
Intercloud Broker

Intercloud Broker

Service Catalog

Service publisher

APIs, cost, ...

Compatibility Set

Cloud A

Service publisher

Compatibility Set

Cloud B
Intercloud Broker

Job specification (e.g., airflow like),
User preferences (e.g., minimize cost, latency, ...)

Service Catalog

Cloud A
Compatibility Set

Cloud B
Compatibility Set

Service publisher

APIs, cost, ...

APIs, cost, ...

23
Intercloud Broker

Intercloud Broker

Service Catalog

Optimizer

Service publisher

Cloud A

Compatibility Set

Service publisher

Cloud B

Compatibility Set

Job specification (e.g., airflow like), User preferences (e.g., minimize cost, latency, ...)

APIs, cost, ...

APIs, cost, ...
Intercloud Broker

Job specification (e.g., airflow like),
User preferences (e.g., minimize cost, latency, …)

Service Catalog
Optimizer
Billing

Service publisher
Compatibility Set

APIs, cost,
...
...
$$_{\text{Cloud A}}$$

Service publisher
Compatibility Set

APIs, cost,
...
...
$$_{\text{Cloud B}}$$
This talk

What is Sky?

Why and how (we believe) Sky will happen?

What if Sky happens?
Three conjectures

Compatibility set is growing quickly

Sky can start with no help from existing clouds

Once started, market forces will do the rest
Compatibility set is growing quickly

Open-source software (OSS) is driving it

All “actors” want it (at some degree):

- Customers
- Third parties
- Clouds themselves
Compatibility set is growing quickly: How?

Open-source software (OSS) is driving it

- OSS dominates at many layers of software stack
- Can run on any cloud → de facto compatibility set

Applications

Cluster orchestration

App packaging
Compatibility set is growing quickly

Customers want to

- Satisfy data and operational sovereignty
- Leverage best-of-breed services and hardware
Compatibility set is growing quickly

Customers want to

- Satisfy data and operational sovereignty
- Leverage best-of-breed services and hardware
- Aggregate resources across clouds
- Reduce cost, latency
- Avoid lock-in
Compatibility set is growing quickly

Third parties provide multi-cloud services

- Reach more customers
- Better compete with public clouds

These services ➔ part of compatibility set
Compatibility set is growing quickly

Clouds themselves drive it

- Provide hosted versions of OSS projects, e.g., k8s, Apache Spark, Apache Kafka
- Provide their own stack on other clouds, e.g., Azure ARC, Google Anthos
- Support competitor’s APIs on their cloud, e.g., S3
Sky needs no help from clouds

Can start with an open source Intercloud Broker

Can start with easy cases; no need to support everything
Once started, market will do the rest

More services

Larger Compatibility Set

More services

Larger Compatibility Set

New services can become part of compatibility layer

More clouds providing interfaces in compatibility set to compete for workload
What Sky doesn’t try to do?
Sky doesn’t try…

… to define a uniform standard API for all clouds

Too hard, unclear whether it is even feasible:

- Cloud API is 10x larger than OS API, and this failed in the OS industry (e.g., Unix war in 90s; POSIX modest success)
- Clouds are not incentivized to support a uniform standard for fear of commoditization
- Would require a huge and lengthy standardization effort
Sky doesn’t even try…

… to impose standards for some services (e.g., same storage API implemented by different clouds)

Instead, it assumes “API == code”, e.g.,

- Spark 3.2, Kubernetes 1.8
- Similar to libraries in today’s programming languages
- Possible bolt-on/shim layers to provide uniform API
Writing a Sky app...

... similar to writing a program; just replace “library” to “service”

Developers are responsible for:

- Specifying the service, its version and config params (if any)
- Manage conflicts and dependencies*

Intercloud layer is responsible for:

- Instantiate service instance and manage its lifetime

*Not ideal, but it’s working today, so likely a good enough start
Sky doesn’t try…

… to support all apps from the beginning

Similar to serverless:

- Start with a few easy but useful use cases
- Add more workloads as Sky matures
This talk

What is Sky?

Why and how (we believe) Sky will happen?

What if Sky happens?
What if Sky happens?

Will lead to specialized clouds and accelerate innovation

● If a cloud is best for a workload, Intercloud Broker will use it
● Will make it easier to integrate on-prem/edge clouds
Sky: the rise of specialized clouds

Sky Computing

Intercloud Compatibility

Specialized cloud A
(Compute-optimized)

NVIDIA
Sky: the rise of specialized clouds

Sky Computing

Intercloud Compatibility

Specialized cloud A (Compute-optimized)

NVIDIA

Intercloud Compatibility

Specialized cloud B (storage-optimized)

DELL EMC

Dark / Lit Fiber

(Free) Peer agreement
Sky: the rise of specialized clouds

Sky Computing

Intercloud Compatibility

Specialized cloud A
(Compute-optimized)

Specialized cloud B
(storage-optimized)

Specialized cloud C
(AI cloud)

Dark / Lit Fiber

(Free) Peer agreement
Sky: the rise of specialized clouds

Intercloud Compatibility
Specialized cloud A
(Compute-optimized)

Intercloud Compatibility
Specialized cloud B
(storage-optimized)

Intercloud Compatibility
Public clouds

Intercloud Compatibility
Specialized cloud C
(AI cloud)

Sky Computing
Sky: the rise of specialized clouds

Sky Computing

Intercloud Compatibility
Specialized cloud A (Compute-optimized)
NVIDIA

Intercloud Compatibility
Specialized cloud B (storage-optimized)
DELL EMC

Intercloud Compatibility
Specialized cloud C (AI cloud)
Equinix
cerebras

Intercloud Compatibility
Edge / Private clouds
AT&T
verizon

Public clouds
IBM

Dark / Lit Fiber

(Free) Peer agreement
What if Sky happens?

Will lead to specialized clouds and accelerate innovation

Will accelerate cloud adoption

- Remove some customer’s concerns: data/operation sovereignty, lock-in
- Like the internet accelerated the growth of networking industry
What if Sky happens?

Will lead to specialized clouds and accelerate innovation

Will accelerate cloud adoption

Will accelerate the growth of software platforms

- E.g., Synapse could run on non-Azure clouds
What if Sky happens?

Will lead to specialized clouds and accelerate innovation

Will accelerate cloud adoption

Will accelerate the growth of software platforms

Will impact many research topics

- Similar with the impact the internet had on the networking research
Summary

Sky Computing

• An analogy to the Internet
• Three problems to be solved
  • Compatibility layer
  • Intercloud layer
  • Peering between clouds
Discussion and Concluding Thoughts
Future of Cloud Computing

• Do you think Sky Computing would be the future?

• Do you think UDC would be the future?

• A combination of them?

• Any other problems with today’s cloud?

• What do you think would be the future or cloud computing? Any major obstacles to that?

• Academia’s role in cloud computing?