Edge Intelligence
Convergence of Humans, Things, and AI

18 June 2019, SummerSOC, Crete

Schahram Dustdar

dsg.tuwien.ac.at
Ecosystems: People, Systems, and Things

**Complex system** with networked dependencies and intrinsic adaptive behavior – has:

1. **Robustness & Resilience mechanisms**: achieving stability in the presence of disruption

2. **Measures of health**: diversity, population trends, other key indicators

3. **Built-in coherence**

4. **Entropy-resistence**
Ecosystems for IoT Systems
Perspectives on the IoT: Edge, Cloud, Internet

(a) A cloud-centric perspective: Edge as “edge of the cloud”

(b) An Internet-centric perspective: Edge as “edge of the Internet”
Cloud-centric perspective

Assumptions

- Cloud provides core services; Edge provides local proxies for the Cloud (offloading parts of the cloud’s workload)

Edge Computers

- play supportive role for the IoT services and applications
- Cloud computing-based IoT solutions use cloud servers for various purposes including massive computation, data storage, communication between IoT systems, and security/privacy

Missing

- In the network architecture, the cloud is also located at the network edge, not surrounded by the edge
- Computers at the edge do not always have to depend on the cloud; they can operate autonomously and collaborate with one another directly without the help of the cloud
Internet-centric perspective

Assumptions

• Internet is center of IoT architecture; Edge devices are gateways to the Internet (not the Cloud)
• Each LAN can be organized around edge devices autonomously
• Local devices do not depend on Cloud

Therefore

• Things belong to partitioned subsystems and LANs rather than to a centralized system directly
• The Cloud is connected to the Internet via the edge of the network
• Remote IoT systems can be connected directly via the Internet. Communications does not have to go via the Cloud
• The Edge can connect things to the Internet and disconnect traffic outside the LAN to protect things -> IoT system must be able to act autonomously
Smart City Example

Everything-as-a-Service (EaaS)
Dynamic Analytics (e.g., Smart City)
Vertical vs. Horizontal Edge Architecture

Paradigm 1: Elasticity (Resilience)

(Physics) The property of returning to an initial form or state following deformation

**stretch** when a force stresses them
- e.g., *acquire* new resources, *reduce* quality

**shrink** when the stress is removed
- e.g., *release* resources, *increase* quality
Elastic Computing > Scalability

Resource elasticity
Software / human-based computing elements, multiple clouds

Quality elasticity
Non-functional parameters e.g., performance, quality of data, service availability, human trust

Costs & Benefit elasticity
rewards, incentives

High level elasticity control

#SYBL.CloudServiceLevel
Cons1: CONSTRAINT responseTime < 5 ms
Cons2: CONSTRAINT responseTime < 10 ms
WHEN nbOfUsers > 10000
Str1: STRATEGY CASE fulfilled(Cons1) OR fulfilled(Cons2): minimize(cost)

#SYBL.ServiceUnitLevel
Str2: STRATEGY CASE ioCost < 3 Euro:
maximize( dataFreshness )

#SYBL.CodeRegionLevel
Cons4: CONSTRAINT dataAccuracy>90% AND cost<4 Euro


Elasticity Model for Cloud Services

Elasticity space functions: to determine if a service unit/service is in the “elasticity behavior”

Elasticity Pathway functions: to characterize the elasticity behavior from a general/particular view

Edge & Blockchains

- **Granularity** (Fine/Coarse): Blockchain assigned to single level (fine) or multiple (coarse)
- **Accessibility** (Private / Public): Private BC or public open system; analyze BC by different agents?
- **Deployment Model** (Virtual/Real): Virtual BCs materialized inside a regular BC

Edge & Blockchains – Integration aspects

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Accessibility</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fine</td>
<td>Hybrid</td>
</tr>
<tr>
<td>2</td>
<td>Coarse</td>
<td>Hybrid</td>
</tr>
<tr>
<td>3</td>
<td>Fine</td>
<td>Public</td>
</tr>
<tr>
<td>4</td>
<td>Coarse</td>
<td>Private</td>
</tr>
</tbody>
</table>
Smart contracts are executed within a single Blockchain.

- Resources
- Quality
- Cost

Make data available across 2 different chains (virtual or regular) with horizontal (same level) or vertical (different levels).
<table>
<thead>
<tr>
<th>Granularity</th>
<th>Accessibility</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>Public</td>
<td>Virtual</td>
</tr>
</tbody>
</table>

Fragments of a Blockchain: Block i to Block i+7
<table>
<thead>
<tr>
<th>Granularity</th>
<th>Accessibility</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>Public</td>
<td>Virtual</td>
</tr>
</tbody>
</table>

Fragments of a Blockchain: Block i to Block i+7

S1: IoT sensor data
Granularity | Accessibility | Deployment
---|---|---
Coarse | Public | Virtual

Fragments of a Blockchain: Block i to Block i+7

S1: IoT sensor data

S2: IoT sensor data
Granularity | Accessibility | Deployment
---|---|---
Coarse | Public | Virtual

Fragments of a Blockchain: Block $i$ to Block $i+7$

**S1**: IoT sensor data

**S2**: IoT sensor data

**EAC**: Elastic Analytics Contract

e.g. creating a service for “presence prediction” (alpha) for the next steps
EGC: Elastic Glue Contract
2 types of EGCs:

a) **EGC1**: aggregates Level 1 info + brings it as new data to Level 2

a) **EGC2**: (implemented as an oracle) imports data off-chain to Level 2
Towards Edge Intelligence

Computational Fabric

- dispersed resources allow training, monitoring, serving of models
- Heterogeneity of applications and models requires
  - (1) flexible and modular infrastructure and
  - (2) intelligent operations mechanisms (due to the scale of the infrastructure)

Operationalization

- Automated AI application lifecycle management to the Edge

Fabric for Edge Intelligence

1. **Sensing (Sensor Data as a Service)**
   - Large number, dynamic and mobile nature of publishers/subscribers of sensor data + QoS requirements of edge intelligence -> rethink centralized messaging services such as AWS IoT or MS Azure IoT Hub
     - Management and governance of such a distributed/decentralized sensing infrastructure

2. **Edge computer network with modular AI capabilities**
   - New AI accelerators for edge devices (e.g., Google Edge TPU with an application specific integrated circuit; MS BrainWave with field-programmable gate arrays (FPGAs); Intel Neural Compute Stick; Baidu Kunlun, Huawei Atlas AI Platform

3. **Intelligent orchestration mechanisms for decentralized and distributed infrastructure**
Edge self-adaptive middleware & scheduling Wfs


Paradigm 2: Osmotic Computing

- In chemistry, “osmosis” represents the seamless diffusion of molecules from a higher to a lower concentration solution.

- Dynamic management of (micro)services across cloud and edge datacenters
  - deployment, networking, and security, ...
  - providing reliable IoT support with specified levels of QoS.

IoT/Data/Application Orchestration
Osmotic movement of MELs in Clouds, Edge, Things

Legend:
MEL...Micro Element
IoT Mircoelements (MELs)

1. **MicroServices** (MS), which implement specific functionalities and can be deployed and migrated across different virtualized and/or containerized infrastructures (e.g., Docker) available across Cloud, Edge, and Things layers.

2. **MicroData** (MD), encodes the contextual information about (a) the sensors, actuators, edge devices, and cloud resources it needs to collect data from or send data to, (b) the specific type of data (e.g., temperature, vibration, pollution, pH, humidity) it needs to process, and (c) other data manipulation operations such as where to store data, where to forward data, and where to store results.

3. **MicroComputing** (MC), executing specific types of computational tasks (machine learning, aggregation, statistical analysis, error checking, and format translation) based on a mix of historic and real-time MD data in heterogeneous formats. These MCs could be realized using a variety of data storage and analytics programming models (SQL, NoSQL, stream processing, batch processing, etc.).

4. **MicroActuator** (MA), implementing programming interfaces (e.g., for sending commands) with actuator devices for changing or controlling object states in the IoT environment.
IoT Data Sources

1. **Representation**: Structure and represent the data to facilitate multiple modalities, exploiting the complementarity and redundancy of different data sources.

2. **Translation**: Interpret data from one modality to another, i.e., provide a translator that allows the modalities to interact with each other for enabling data exchange.

3. **Alignment**: Identify the relation among modalities. This requires identifying links between different types of data.

4. **Fusion**: Fuse information from different modalities (e.g., to predict).

5. **Co-learning**: Transfer knowledge among modalities. This explores the field of how the knowledge of a modality can help or enhance a computational model trained on a different modality.
IoT Programming Patterns needed

1. **Decomposing IoT data analysis activities into fine-grained activities** (e.g., statistics, clustering, classification, anomaly detection, accumulation, filtering), each of which may impose different planning and run-time orchestration requirements;

2. Identifying and integrating **real-time data from IoT devices and historical IoT data** distributed across Cloud and Edge resources;

3. Identifying **data and control flow dependencies** between data analysis activities focusing on coordination and data flow variables, as well as the handling of dynamic system updates and re-configuration;

4. Defining and tagging each **data analysis activity with runtime deployment constraints** (QoS, security and privacy).
Managing the AI Lifecycle

AI lifecycle pipeline with a rule-based trigger $e$ that monitors available data and runtime performance data to form an automated retraining loop.
<table>
<thead>
<tr>
<th>C2C</th>
<th>Training data is centralized</th>
<th>Models are large</th>
<th>Scalable learning infrastructure [39]</th>
<th>Image search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Massive data sets</td>
<td>Huge number of inferencing requests need to be load balanced</td>
<td>Data warehousing</td>
<td>Recommender systems</td>
</tr>
<tr>
<td>C2E</td>
<td>Training data is centralized</td>
<td>Inferencing may need to happen in near-real time</td>
<td>Model compression [42]</td>
<td>Surveillance systems</td>
</tr>
<tr>
<td></td>
<td>Inferencing data may be sensitive</td>
<td>Large number of model deployments</td>
<td>Latency/accuracy tradeoff [43]</td>
<td>Self driving cars</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Models run on specialized hardware</td>
<td>Distributed inferencing [44]</td>
<td>Fieldwork assistants</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Transfer learning [45]</td>
<td></td>
</tr>
<tr>
<td>E2C</td>
<td>Training data is distributed</td>
<td>Models can be centralized</td>
<td>Decentralized/federated learning [41]</td>
<td>Volunteer computing</td>
</tr>
<tr>
<td></td>
<td>Training data may be sensitive</td>
<td>Huge number of inferencing requests need to be load balanced</td>
<td></td>
<td>Novel Smart City use cases</td>
</tr>
<tr>
<td>E2E</td>
<td>Training data is distributed</td>
<td>Inferencing may need to be near-real time</td>
<td>Decentralized/federated learning</td>
<td>Industrial IoT (e.g., predictive maintenance)</td>
</tr>
<tr>
<td></td>
<td>Training and inferencing data may be sensitive</td>
<td></td>
<td>Distributed inferencing</td>
<td>Privacy-aware personal assistants</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Novel IoT use cases</td>
</tr>
</tbody>
</table>
Conclusions

• Need for an Edge Intelligence AI Fabric and a “clear” ecosystems understanding

• Better levels of integration between multiple (levels of) Blockchains and stakeholders can be achieved

• Integrate AI, IoT, and human collectives into processes distributed on the Edge and Cloud
Thanks for your attention

Prof. Schahram Dustdar
IEEE TCSVC Outstanding Leadership Award in Services Computing
Member of Academia Europaea
IBM Faculty award
ACM Distinguished Scientist
IEEE Fellow

Distributed Systems Group
TU Wien, Austria
dsg.tuwien.ac.at