Advanced service-based data analytics: Models, Elasticity and APIs

Hong-Linh Truong
Distributed Systems Group, TU Wien

truong@dsg.tuwien.ac.at
dsg.tuwien.ac.at/staff/truong
@linhsolar
Outline

- Principles of elasticity for advanced service-based data analytics
- Data analytics within a single system
- Data analytics across multiple systems
- APIs management
Recall data-as-a-service and data marketplaces are key elements for data-driven economy
PRINCIPLES OF ELASTICITY FOR DATA ANALYTICS
Complex dependencies in (big) data analytics

- More data $\rightarrow$ more computational resources (e.g. more VMs)

- **More types of data** $\rightarrow$ more computational models $\rightarrow$ more analytics processes

- Change quality of analytics
  - Change quality of data
  - Change response time
  - Change cost
  - Change types of result (form of the data output, e.g. tree, visual, story, etc.)
Complex dependencies in (big) data analytics

a) Data analytics with the same type of data

b) Data analytics with multiple types of data

c) Impact of quality of results on data, analytics processes, and computational models


ASE Summer 2016
Elasticity principles can be used to support dynamic quality of analytics
Elasticity Principles: Elasticity of data and computational models

- Multiple types of objects from different sources with complex dependencies, relevancies, and quality
- Different data and computational models for the same analytics subject
- New analytics subjects can be defined and analytics goals can be changed
- Decide/select/define/compose not only computational models for analytics subjects but also data models based on existing ones

Management and modeling of elasticity of data and computational model during the analytics
Elasticity Principles: Elasticity of data resources

- Data provided, managed and shared by different providers
- Data associated with different concerns (cost, quality of data, privacy, contract, etc.)
- Static data, open data, data-as-a-service, opportunistic data (from sensors and human sensing)
- Not just centralized big data and total data ownership

Data resources can be taken into account in an elastic manager: similar to VMs, based on their quality, relevancy, pricing, etc.
Elasticity Principles: Elasticity of humans and software as computing units

- Human in the loop to solve analytics tasks that software cannot solve
- Human-based compute units can be scaled up/down with different cost, availability, and performance models
- Human-based compute units + software-based compute units for executing computational models
- Elasticity controls can be also done by humans

Provisioning hybrid compute units in an elastic way for computational/data/network tasks as well as for monitoring/control tasks in the analytics process
Elasticity Principles: Elasticity of quality of analytics

- Definition of quality of analytics
  - Trade-offs of time, cost, quality of data, forms of output
- Using quality of analytics to select suitable computational models, data resources, computing units
- Multi-level control for the elasticity based on quality of analytics

Able to cope with changes in quality of data, performance, cost and types of results at runtime
Advanced service-based analytics – which are fundamental engineering questions?
Advanced service-based data analytics (1)

Cities, e.g. including:
10000+ buildings
1000000+ sensors
Advanced service-based data analytics -- fundamental concepts

Domain 1

Applications

Part A

Domain 2

Part B

... 

Domain n

Part N

System infrastructures

IoT

Edge servers

Local Cloud

Public cloud

ASE Summer 2016
Design questions

- Which system infrastructures are used?
- Which interfaces are suitable for units?
- Which programming models are used within units?
- Which are fundamental units to be used?
- How do different units interact?
- Which non-functional parameters are important and how to measure them?
Fundamental concepts – system infrastructure unit

System infrastructures

- Human-based Computing
- Cloud Services
- Fog/Edge Computing Nodes
- Cluster
- Grid
- High Performance Server
Fundamental concepts – unit functions
Fundamental concepts – programming model within units

Programming model

- Spark
- MapReduce
- MPI
- Parallel Database
- Workflow
- Other solutions
Fundamental concepts – interfaces between units

- Interface
  - Style
    - REST
    - SOAP
  - APIs
    - Specific APIs
    - Standard APIs
  - Interaction
    - Pull
    - Push
Fundamental concepts – services and data concerns

- Service and data concerns
  - Data concerns
    - Quality of data
    - Pricing
    - Data Right
  - Service Concerns
    - QoS
    - Pricing
    - ...

You see we need to deal with many techniques and frameworks
WE NEED TO START FROM DATA ANALYTICS WITHIN A SINGLE SYSTEM
Data analytics within a single system

- They are complex enough but do not meet all requirements
- In a single domain
  - Tightly coupled computing infrastructures
    - E.g., in the same cloud
  - Computation and data are close
  - Several concerns can be by-passed

Not always provisioned under the „Service Unit“ model
Data analytics within a single system

Some papers


Data analytics within a single system – some examples

- Message Passing Interface (MPI) + Cluster-based File system
- MapReduce + Google File System
- Hadoop + HDFS
- Dryad+LINQ
- Parallel Database (SQL/NonSQL)
- Yahoo S4
- Spark
- Scientific/Business Workflow


ASE Summer 2016 25
WHY SHOULD ANALYTICS UNITS BE „CLOSED“ TO DATA UNITS?
WHICH CONCERNS COULD BE IGNORED IN SINGLE SYSTEM DATA ANALYTICS?
WHICH ARE THE ISSUES THAT WE NEED TO CONSIDER WHEN OUR DATA UNITS ARE IN DIFFERENT SYSTEMS?
Data analytics across multiple systems – design choice

- Programming models for data analytics service
- Data service units
- Supporting middleware units
Data analytics across multiple systems – programming models (1)

- Static data

Input data → Local input data

MapReduce/Hadoop Workflow MPI Other solutions

Servers/Cloud/Cluster

Analytics Results

Output data

What are our design concerns?
Data analytics across multiple systems – programming models (2)

- Near-realtime data

![Diagram of data analytics process]

- Stockmarket
- Social media
- M2M

What are our design concerns?

ASE Summer 2016
Data analytics across multiple systems – programming models (3)

- Near-realtime data

What are our design concerns?
Data analytics across multiple systems – data service units

- **Data Analytics Unit**
  - Read/write data via direct, low-level read/write via IO
  - Cluster or cluster of clusters
  - Can be very large

- **Interface**
  - Read/write data via direct, low-level read/write via IO

- **System**
  - Cluster or cluster of clusters
  - Can be very large

- **Programming model**
  - Usually parallel processing

- **Files and File Systems**
  - NFS
  - Lustre
  - Hadoop File System
  - Google file system

ASE Summer 2016
Data analytics across multiple systems – data service units

Data Analytics Unit

- Direct data transfer via REST/SOAP APIs
- Decouple between analytics and storage
- May require middleware for data transfer
- Request via SOAP/REST
- Real data transfer done by external middleware
- A rich set of programming models can be used

Storage-as-a-Service

- Amazon S3 (SOAP/REST API)
- Google Storage Service (REST API)

Interface

System

Programming model
Data analytics across multiple systems – data service units

Data Analytics Unit

Database-as-a-Service

queries

data

Interface

• REST/SOAP APIs
• Mainly for commands and results

System

• Decouple between analytics unit and database
• Database as a service can be very large

Programming model

• Analytics can be done at both sides
• Analytic units can use any programming models
• Database-as-a-service can perform a lot of analytics
• Parallel database operations

Technology

MongoDB/MongoLab
Amazon DynamoDB
Amazon SimpleDB
Cloudant Data

SkySQL
Amazon RDS
Microsoft SQL Azure
Clustrix DBaaS
Data analytics across multiple systems – data service units

**Technology**
- Infochimps
- Microsoft Azure
- Xively
- GNIP

**Data Analytics Unit**

**Interface**
- Data transfer can be uni or bi-direction
- REST/SOAP APIs

**System**
- Both systems for DaaS and for analytics units can be very large

**Programming model**
- Can be any
Middleware service unit for transferring large data -- GlobusOnline

Figure 1. Globus Online architecture.

Figure 2. Globus Connect architecture.

Middleware service units for messages/queuing

- Advanced Message Queuing Protocol (AMQP)
- Simple (or Streaming) Text Orientated Messaging Protocol (STOMP)
- Specific protocols/APIs

StormMQ
RabbitMQ
Amazon SQS
So many types of services from different providers. Anyway to simplify the management of service units for the developer/operator?
API MANAGEMENT
Ecosystem view for advanced service engineering

- Complex data analytics applications → need to understand potential service units from an ecosystem perspective
  - Interdependent systems: Social computing, mobile computing, cloud computing, data management, etc.
  - Different types of information are linked
  - Blending vertical and horizontal analytics
  - Different functions (analytics, visualization, communications, etc.)
  - Too many different types of customers (and their interactions)
APIs

- APIs are key! Why?
  - Enable access to data and function from entities in your ecosystem
  - Virtualization

- An API is an asset
  - We need to have lifecycle, pricing, management, etc.

Check [http://www.apiacademy.co](http://www.apiacademy.co) for some useful tutorials
API Fasade

Source: Web API Design, Brian Mulloy
http://apigee.com/about/resources/ebooks/web-api-design

Source:
https://en.wikipedia.org/wiki/Facade_pattern
Managing APIs ecosystems

Customer 1
Complex service n

Customer 2
Complex service m

API Management Service

Enterprise 1
Service Units

Enterprise k
Service Units

Enterprise q
Service Units

Clouds

Cloud/On-premise

Clouds

ASE Summer 2016
Development of APIs

- Not just the functions behind the APIs
  - This we have learned since a long time
- Emerging (business/service) management aspects
  - Usage control and security
  - Any where from any device for any customer
    - Interfaces (communications, inputs/output formats)
- APIs as a service:
  - Availability and reliability of APIs are important – think APIs are similar to a service that your client will consume
Prevent too many accesses?

```
REST_FRAMEWORK = {
    'DEFAULT_THROTTLE_CLASSES': (
        'rest_framework.throttling.AnonRateThrottle',
        'rest_framework.throttling.UserRateThrottle'
    ),
    'DEFAULT_THROTTLE_RATES': {
        'anon': '100/day',
        'user': '1000/day'
    }
}
```

Code: http://www.django-rest-framework.org/api-guide/throttling/#how-throttling-is-determined
How can we use API management for data/service contracts?
Issues on APIs management

- Publish
  - Business and operation planning
    - API usage schemes (e.g., pricing, data concerns)
    - API payload transform policies
    - API throttling
  - API publish and discovery (like service discovery?)

- Management
  - Management roles in enterprises, versions, etc.

- Monitoring and analytics
  - monitoring and analytics information (availability, types of customers, usage frequencies, etc.)
Some well-known frameworks

- http://apigee.com
- http://wso2.com/api-management/
- [https://www.mashape.com/](https://www.mashape.com/)
- http://apiaxle.com/
Build your own APIs ecosystem

- Which APIs you need? Which ones are crucial for you to build complex services?
  - Data APIs
    - Data collection
    - Visualization
    - Analytics APIs
  - Communication
  - Coordination of tasks
- API marketplaces → your APIs
- Using existing API platforms to manage your APIs
Examples of an API marketplace
Use API Management for your mini project?

Apigee Edge
API Management for Businesses of Every Size

Deploy Quickly
Build and launch your production API with self-serve tools and analytics.

Proven Technology
Deploy on the platform trusted by the world’s most demanding enterprises.

Scale Your Success
Migrate to Apigee Edge for billions of API calls as your business grows.

edge Trial
30-day free evaluation (renewable)
Free
Start Free Trial

edge Startup
Production-grade APIs for startups
from $300/mo.
Select

edge SMB
Powerful API management for your business
from $2250/mo.
Select

edge
Enterprise-class API Management
Call Us!
Contact Us

INCLUDED SERVICES

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Apigee Cloud</th>
<th>Apigee Cloud</th>
<th>Apigee Cloud</th>
<th>Apigee or Private Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>Free</td>
<td>from $300/mo.</td>
<td>from $2250/mo.</td>
<td>Contact us</td>
</tr>
<tr>
<td>API Calls</td>
<td>Up to 1 million</td>
<td>2 million per month</td>
<td>8 million per month</td>
<td>250 million per quarter and up</td>
</tr>
</tbody>
</table>
Exercises

- Read mentioned papers
- Analyze the relationships between programming models and system infrastructures for data analytics across multiple domains
- Examine [http://cloudcomputingpatterns.org](http://cloudcomputingpatterns.org) and see how it supports data analytics patterns
- Develop some patterns for data analytics across multiple systems
- Setup an API management platform for your work
Thanks for your attention

Hong-Linh Truong
Distributed Systems Group, TU Wien
truong@dsg.tuwien.ac.at
dsg.tuwien.ac.at/staff/truong
@linhsolar