



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Optical
Communications
Group



How data analytics can help to manage services: An example for CDN

Luis Velasco

lvelasco@ac.upc.edu

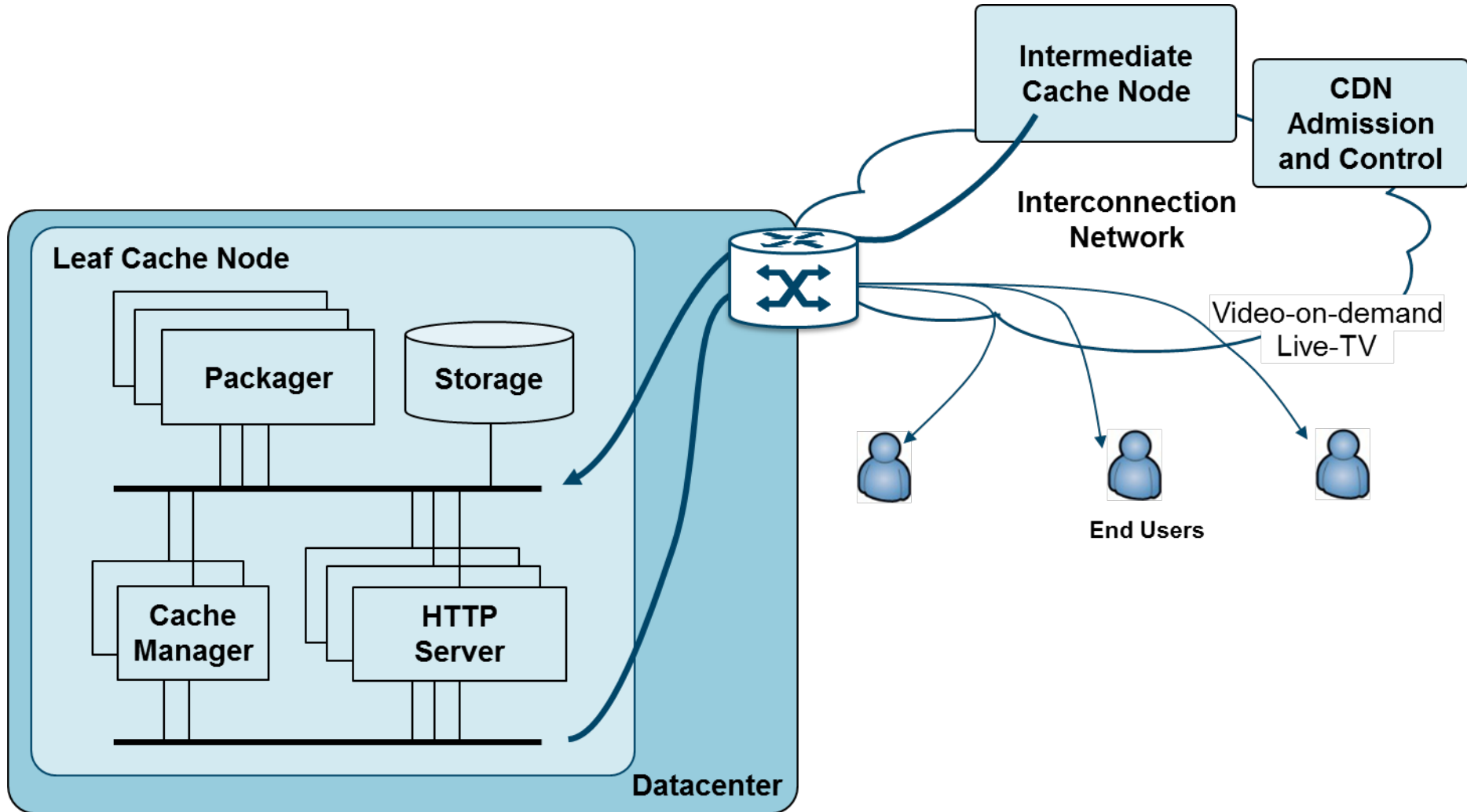
Workshop "Big Data for networking: opportunities, challenges, and requirements"

Cartagena, 2016

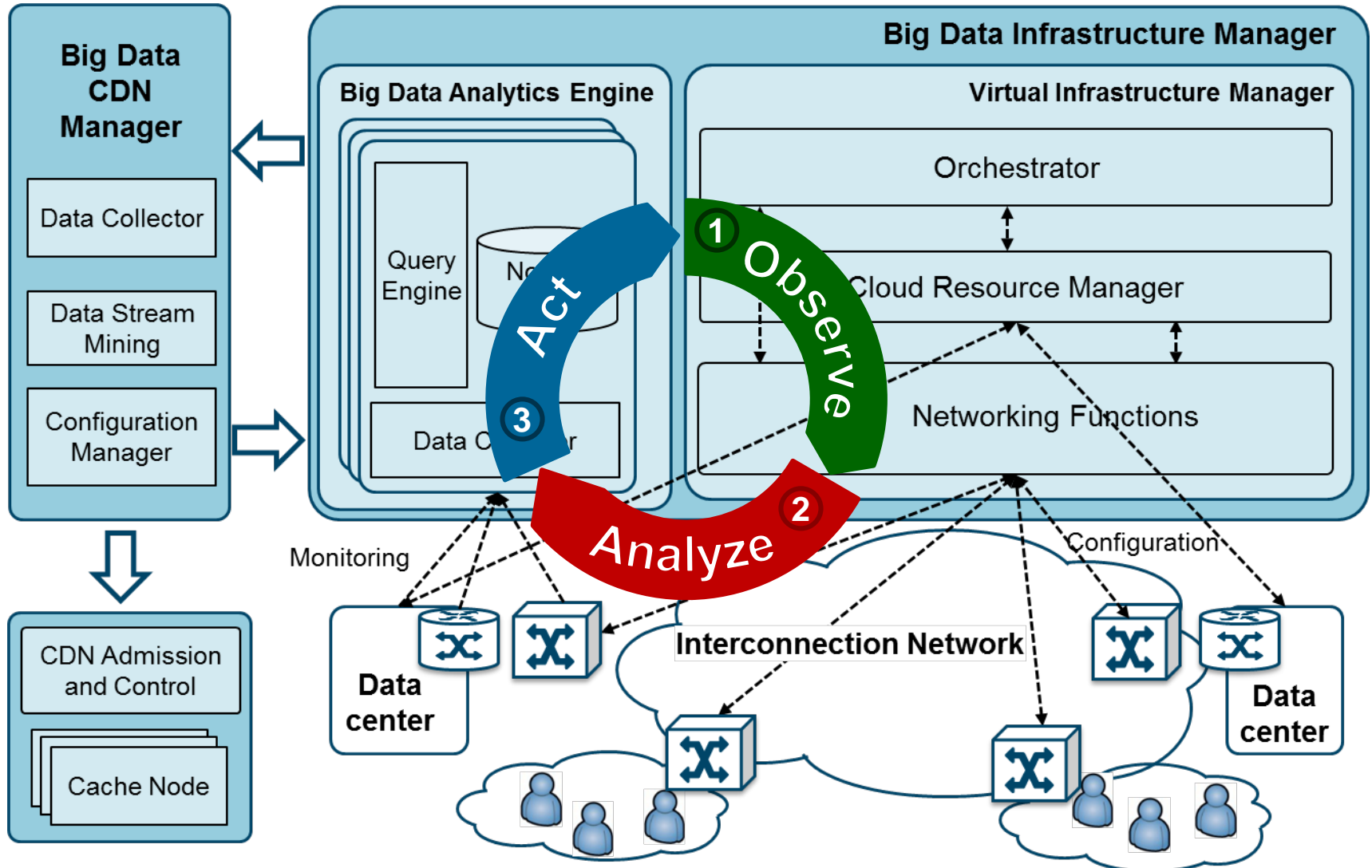
Motivation

- Telecom operators are starting the deployment of CDN to better **control and manage video contents** injected into the network.
 - Cache nodes are placed close to end users to adapt them to users' devices
 - Video traffic in the core is reduced
- By adopting the standardized MPEG-DASH technique, video contents can be delivered over HTTP.
 - **HTTP servers** can be used to serve contents,
 - **packagers** running as software can prepare live contents.
 - The CDN function can be **virtualized**.
- We propose a Big Data architecture to adapt the virtualized CDN function to current and future demand.
 - It controls virtualized components while collecting and pre-processing data.
 - Optimization problems minimize costs and ensure the highest video quality.
 - Machine learning techniques produce estimation of future scenarios.

Cache architecture and cache node main components



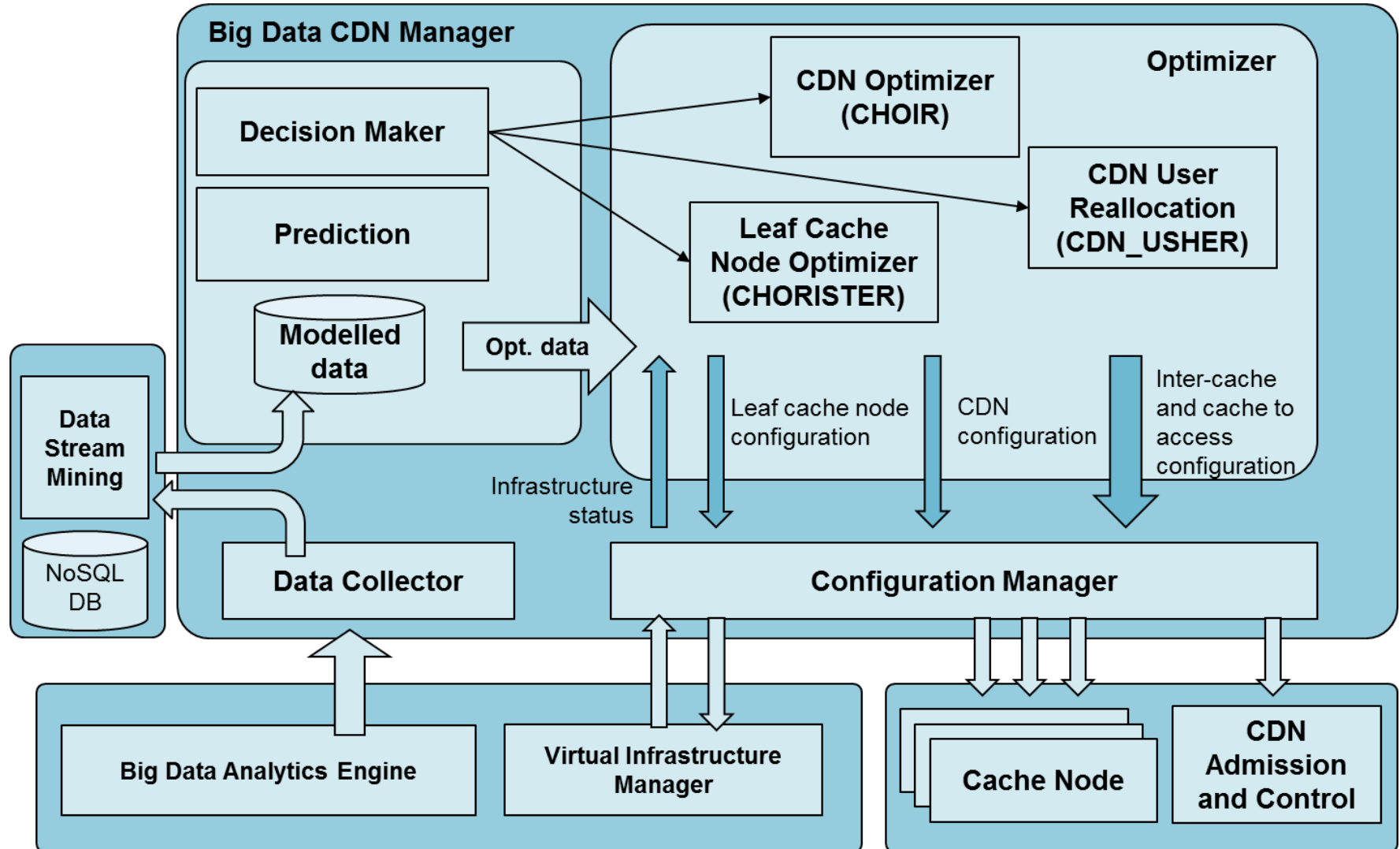
Architecture supporting the Big Data CDN Manager



Dynamic CDN adaptation

- Based on future load, the CDN can be optimized to minimize total costs while serving contents to the end users ensuring the highest QoS.
 - A *prediction module* (PROMPTER) based on machine learning and time series modelling is proposed to predict future load.
- Three optimization problems have been devised:
 - the *Global CDN Optimization* (CHOIR) problem targets at serving users with the highest QoS level from leaf cache nodes with the minimum cost;
 - the *CDN User Reallocation* (CDN_USHER) problem focuses on reallocating users among leaf cache nodes, just updating connections between cache nodes and metro areas;
 - the *Leaf Cache Node Optimizer* (CHORISTER) problem that scales a cache node down.
- Analyzing current and predicted load distribution, a *decision maker module* (TUNER) is responsible of **triggering the most appropriate** optimization problem as well as **selecting meaningful input data**.

Components of the Big Data CDN Manager



Global CDN Optimization (CHOIR) problem

Given:

- A set IC of intermediate cache nodes.
- A set of cache node types: $\{void, TV, VoD+TV\}$.
- A set LC of locations where leaf cache nodes are deployed and the allowable cache node types.
- A set A of metro areas with users consuming contents.
- The set O of contents being consumed. Contents include VoD and live-TV channels.
- A set U of user groups. Each group u contains all users inside a metro area that are currently playing a specific content with a specific device.

Output:

- Configuration of every $l \in LC$, including creating or releasing leaf cache nodes,
- Assignment (u, l) for every $u \in U$,
- Connections to be created/released/reconfigured.

Objective: minimize the CDN cost from setting up resources in leaf caches and the needed connections.

CDN User Reallocation (CDN_USHER) problem

Given:

- A set LC of locations where leaf caches are currently deployed.
- A set O of contents currently being served.
- A set A of metro areas with users consuming contents.
- A set U of user groups. Each group u contains all users inside a metro area that are currently playing a specific content with a specific device.

Output:

- Assignment (u, l) for every $u \in U$,
- Connections to be created/released/reconfigured.

Objective: minimize the cost from the new connections to be established and the total number of users reallocated.

Leaf Cache Node Optimizer (CHORISTER) problem

Given:

- The current size of the HTTP servers' pool (s^{cur}) and its current bandwidth utilization k^{cur} (in %).
- A target bandwidth utilization (th), e.g. 60%.

Output: Target size (s^{tgt}) of the HTTP servers' pool.

Objective: minimize the size of the HTTP servers' pool.

Tuner Decision Algorithm

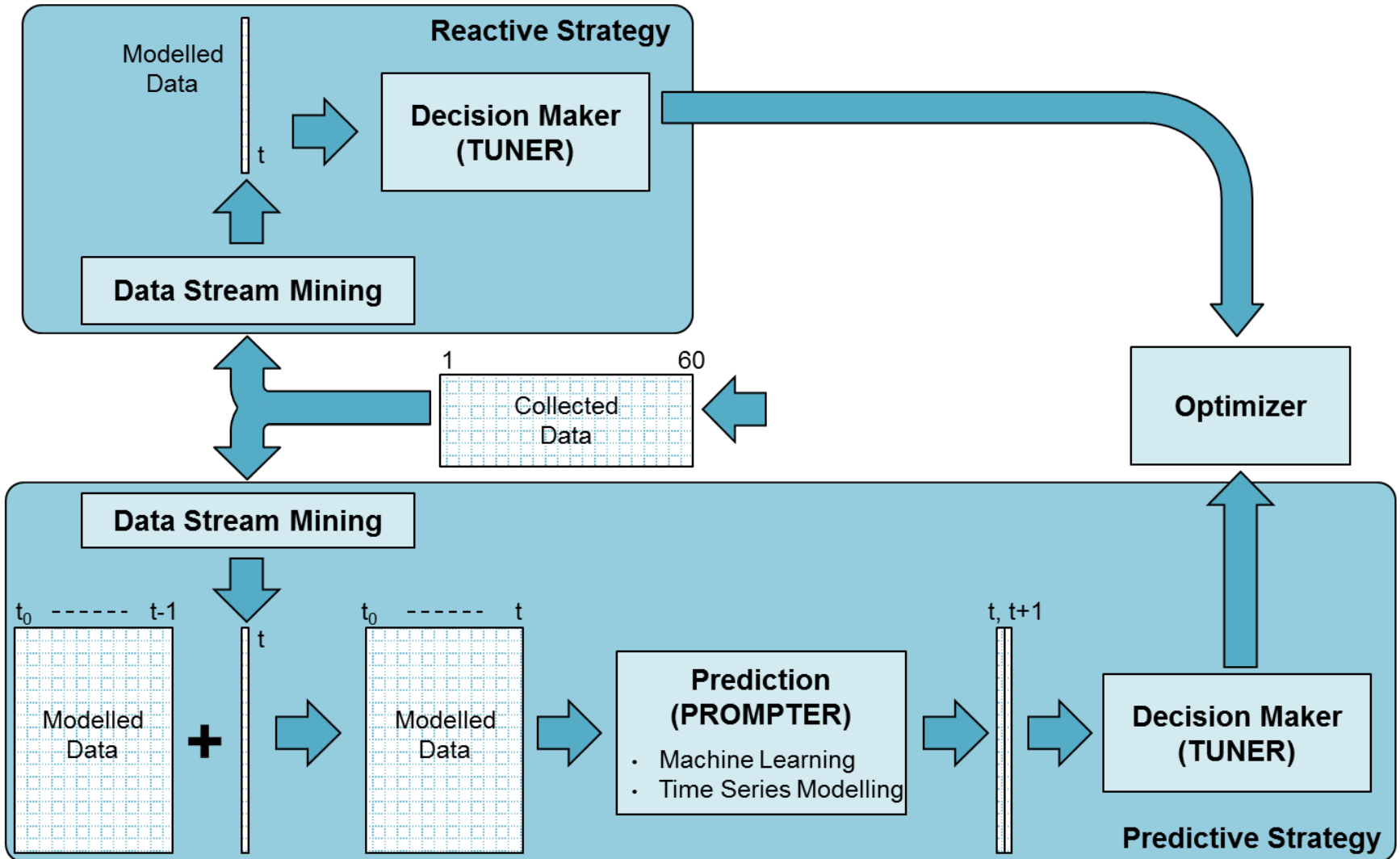
- q_l Average video quality metric provided to users by cache node $l \in LC$.
 k_l Average bandwidth utilization (in %) of interfaces in VMs running HTTP servers in cache l .
 k_e Average bandwidth utilization (in %) of link $e \in E_{LC-A}$.
 s_l Size of cache node $l \in LC$

```

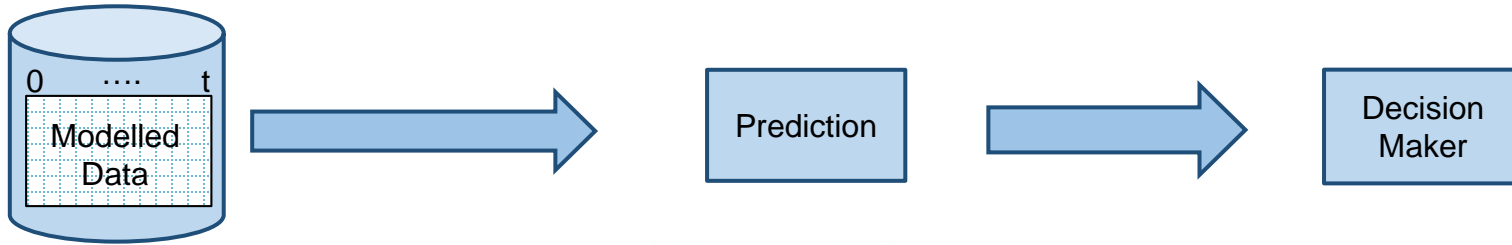
1: UnderuL, UnderuE ← false
2: for each  $l$  in  $LC$  do
3:   if  $s^{tgt} < s^{cur}$  then UnderuL ← true
4:   for each  $e$  in  $E_{LC-A}$  do
5:     if  $k_e^{max} < th_e$  then UnderuE ← true
6:   for each  $l$  in  $LC$  do
7:     if  $q_l^{min} < th_q$  AND (UnderuL OR UnderuE) then
8:       if CDN_USHER (input data) = FEASIBLE then
9:         return
10:  for each  $l$  in  $LC$  do
11:    if  $q_l^{min} < th_q$  then
12:      CHOIR (input data); return
13:  for each  $l$  in  $LC$  do
14:    if  $s^{tgt} < s^{cur}$  then CHORISTER (input data)
    
```

- TUNER compares the evolution of modelled variables against a low threshold ($th_{(\cdot)}$).
- The CDN-USHER problem is solved to reassign groups of users in the case that the **quality** of the video being served from **some cache nodes starts degrading**, whereas the capacity of some other cache nodes is underutilized.
- In the case that a **more in depth CDN** reconfiguration is needed, the CHOIR problem needs to be solved.
- Finally, the CHORISTER problem is solved to scale underutilized cache nodes down.

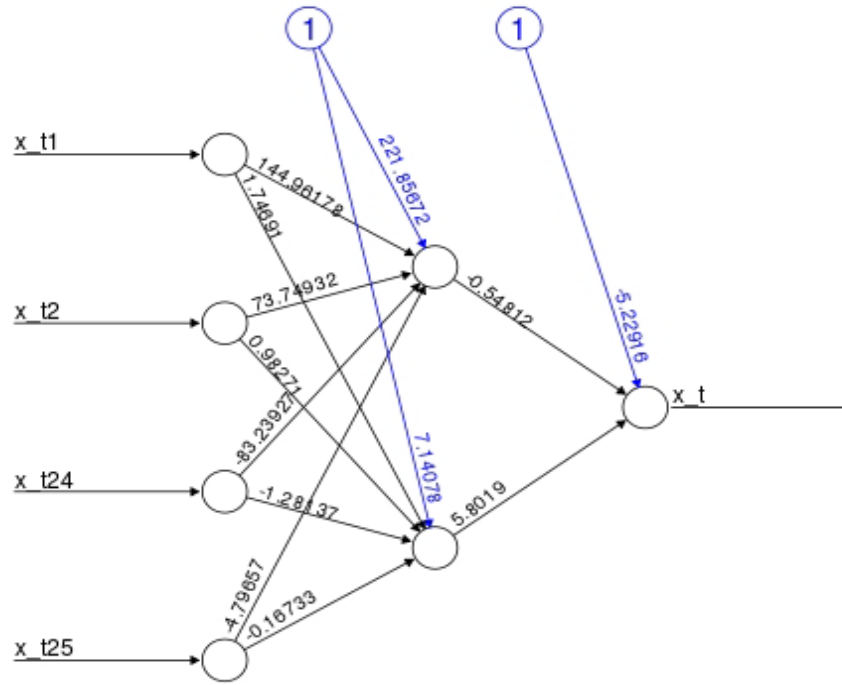
From collected data to making decisions



Load prediction and decision making

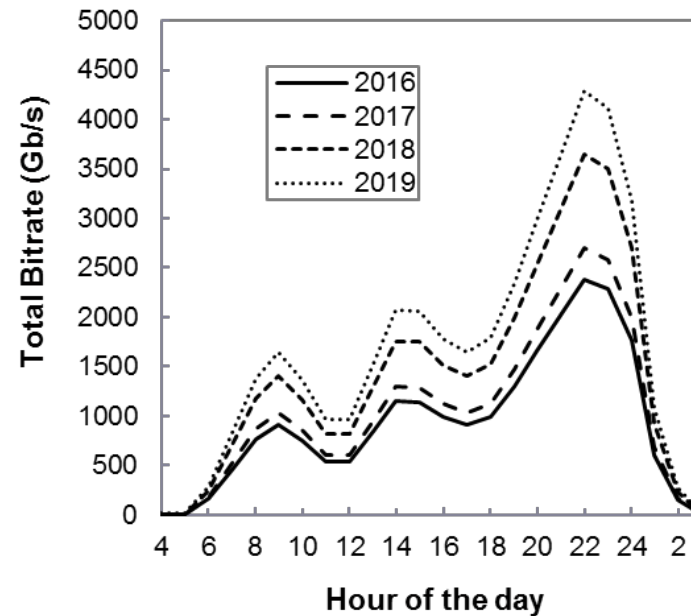
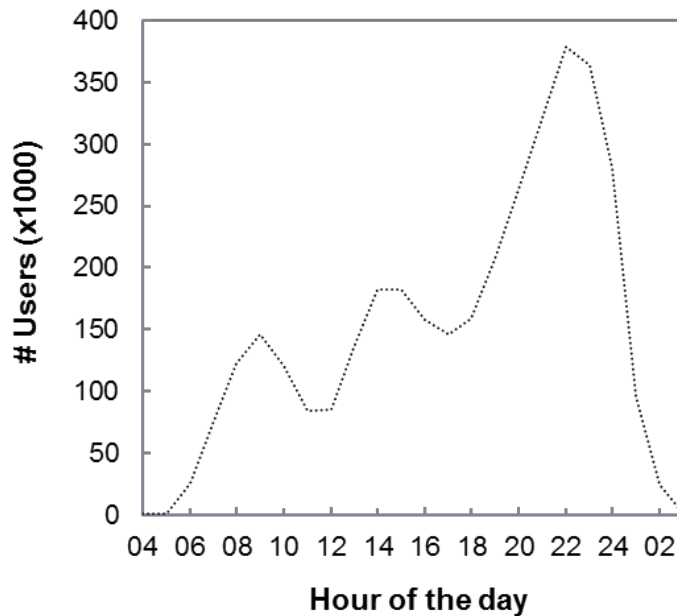
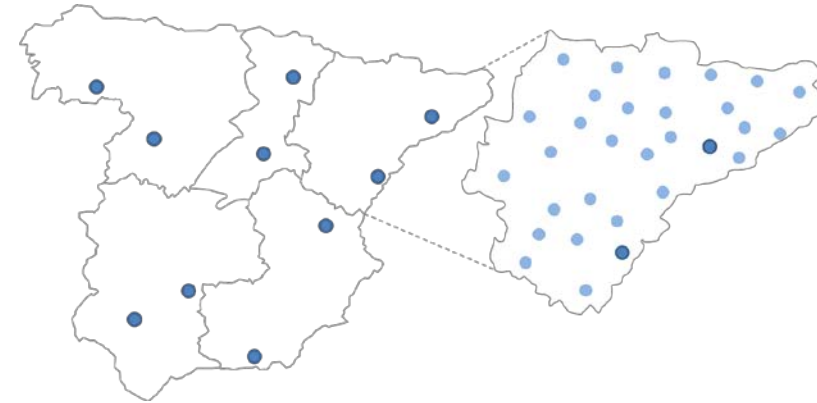


Artificial Neural Network (ANN)

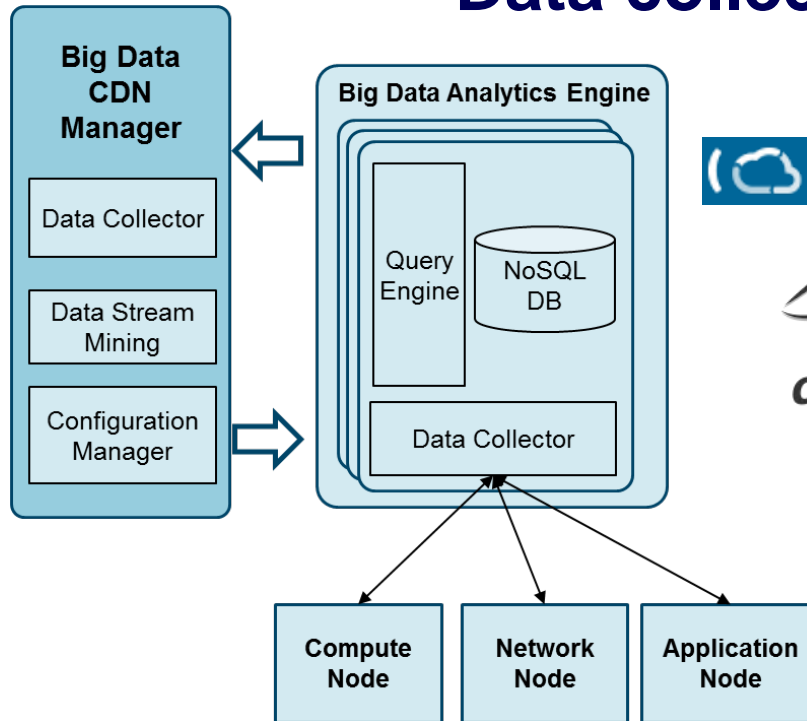


Adoption Scenarios

Quality (Mb/s)	2016	2017	2018	2019
SD (2.1)	28.6	22.0	13.6	7.0
HD (4)	38.6	34.3	28.2	23.7
Full HD (10)	27.1	37.7	42.3	47.3
4K UHD (25)	5.7	6.0	15.9	22.0



Data collection and analysis



Time	Src Addr	Dst Addr	Proto	Info
REF	CDNManager	OpenContrail	HTTP	POST /analytics/query HTTP/1.1 (application/json)
1.023655	OpenContrail	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)
1.024665	CDNManager	ApacheSpark	JSON-over-TCP	JSON-over-TCP: JSON Object: collected_data
61.088940	CDNManager	OpenContrail	HTTP	POST /analytics/query HTTP/1.1 (application/json)
62.101560	OpenContrail	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)
62.102837	CDNManager	ApacheSpark	JSON-over-TCP	JSON-over-TCP: JSON Object: collected_data
122.153510	CDNManager	OpenContrail	HTTP	POST /analytics/query HTTP/1.1 (application/json)
123.165446	OpenContrail	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)
123.166771	CDNManager	ApacheSpark	JSON-over-TCP	JSON-over-TCP: JSON Object: collected_data
174.206638	ApacheSpark	CDNManager	JSON-over-TCP	JSON-over-TCP: JSON Object: modelled_data

Conclusions

- A Big Data -backed virtualized CDN architecture to be deployed in the telecom cloud has been proposed and experimentally assessed.
 - It consists of a hierarchy of cache nodes: the centralized intermediate cache and leaf cache nodes, located close to the end users.
- Media content can be delivered over HTTP by using the standardized MPEG-DASH technique.
 - a virtualized leaf cache node would consists of a number of HTTP servers, packagers and a cache manager.
 - All components run as software inside VMs deployed in the same DC.
- A CDN manager is responsible for adapting the CDN function to current and future load.
- The PROMPTER module produces estimation of future scenarios
 - It uses statistical linear regression and machine learning techniques to estimate the value of modelled variables for the next period.



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Optical
Communications
Group



Thank you for your attention!

How data analytics can help to manage
services: An example for CDN

Luis Velasco

lvelasco@ac.upc.edu

M. Ruiz, M. Germán, L. M. Contreras, and L. Velasco
Big Data-backed Video Distribution in the Telecom Cloud
Elsevier Computer Communications, vol. 84, pp. 1-11, 2016.