



## Case Studies in Application Placement and Infrastructure Optimisation

*Miguel Angel López-Peña, Hector Humanes,  
Johan Forsman, Thang Le Duc, Peter Willis,  
and Manuel Noya*

**Abstract** This chapter presents four case studies each illustrating an implementation of one or more RECAP subsystems. The first case study illustrates how RECAP can be used for infrastructure optimisation for a 5G network use case. The second case study explores application

---

M. A. López-Peña (✉) • H. Humanes  
Sistemas Avanzados de Tecnología, S.A. (SATEC), Madrid, Spain  
e-mail: [miguelangel.lopez@satec.es](mailto:miguelangel.lopez@satec.es); [hector.humanes@satec.es](mailto:hector.humanes@satec.es)

J. Forsman • T. Le Duc  
Tieto Product Development Services, Umeå, Sweden  
e-mail: [johan.forsman@tieto.com](mailto:johan.forsman@tieto.com); [thang.leduc@tieto.com](mailto:thang.leduc@tieto.com)

P. Willis  
BT plc, London, UK

M. Noya  
Linknovate, Palo Alto, CA, USA  
e-mail: [manuel@linknovate.com](mailto:manuel@linknovate.com)

optimisation for virtual content distribution networks (vCDN) on a large Tier 1 network operator. The third case study looks at how RECAP components can be embedded in an IoT platform to reduce costs and increase quality of service. The final case study presents how data analytics and simulation components, within RECAP, can be used by a small-to-medium-sized enterprise (SME) for cloud capacity planning.

**Keywords** Infrastructure management • Network management • Network functions virtualisation • Virtual content distribution networks • Smart cities • Capacity planning • Application optimisation • Infrastructure optimisation • Big Data analytics • 5G networks

## 6.1 INTRODUCTION

This chapter illustrates how RECAP's approach to the management and optimisation of heterogeneous infrastructure across the cloud-to-edge spectrum can address challenges to a wide range of cloud actors and applications. Four illustrative case studies are presented:

- Infrastructure Optimisation and Network Management for 5G Networks
- Application Optimisation for Network Functions Virtualisation for Virtual Content Distribution Networks
- Application and Infrastructure Optimisation for Edge/Fog computing for Smart Cities
- Capacity Planning for a Big Data Analytics Engine

For each case, we can show that insights and models generated by RECAP can help the stakeholders to better understand their application and infrastructure behaviour. Preliminary results suggests cost savings of more than 25%, up to 20% reduction in bandwidth consumption, and a 4% performance increase.

## 6.2 CASE STUDY ON INFRASTRUCTURE OPTIMISATION AND NETWORK MANAGEMENT—5G NETWORKS

### 6.2.1 *Introduction*

This case study envisions a system that provides communication services for a variety of industry verticals including eHealth, eCommerce, and automotive. To facilitate the communications of diverse services in

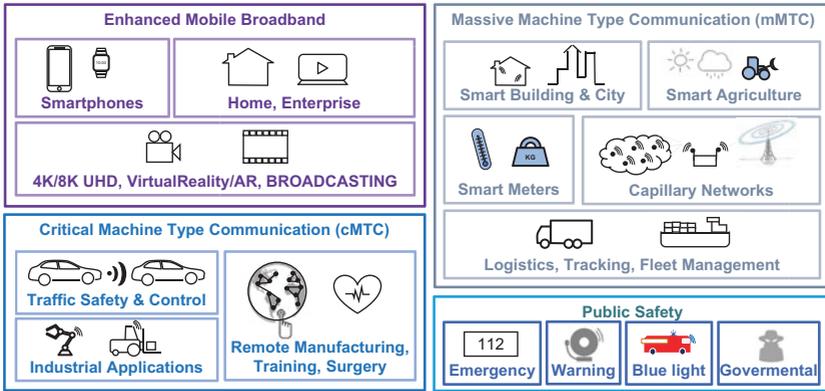


Fig. 6.1 Categories of communication services and example of 5G use cases

different scenarios within the world of fifth generation (5G) networks, the communication system has to support various categories of communication services illustrated in Fig. 6.1.

Each service is needed for a specific type of application serving a particular group of customers/clients. This introduces different sets of characteristics and requirements corresponding to each type of communication service as presented in Table 6.1.

The emergence of 5G mobile networks and the rapid evolution of 5G applications are accelerating the need and criticality of optimised infrastructure as per this case study. Additionally, the management and operation of a 5G infrastructure and network are complex not only due to the diversity of service provisioning and consumer requirements, but also due to the involvement of many stakeholders (including infrastructure service providers, network function/service providers, and content and application service providers). Each of these stakeholders has their respective and different levels of demands and requirements. As a result, a novel solution is required to enable:

- the adoption of various applications under different scenarios on a shared and distributed infrastructure;
- on-demand resource provisioning considering increased network dynamics and complexity; and
- the fulfilment of Quality of Service (QoS) and Quality of Experience (QoE) parameters set and agreed with service consumers.

**Table 6.1** Characteristics and requirements of communication services

<i>Service</i>	<i>Characteristics and requirements</i>
Mobile BroadBand (MBB)	Extremely high throughput and user/device mobility
Massive Machine Type Communication (mMTC)	Supports to diverse and massive number of mobile devices, and to enable energy-efficient communications
Mission-critical Machine Type Communication (cMTC)	Ultra-reliable low latency, but high availability and reliability in communications
Public safety (blue light)	Intensely high integrity and availability in services

### 6.2.2 *Issues and Challenges*

The communication services between user mobile devices and content services/applications are realised with a set of network functions and numerous physical radio units. In the context of virtualised networks, network functions are virtualised and chained to each other to form a network service providing network and service access to user devices through radio units. A network function virtualisation (NFV) infrastructure is required to accommodate the network function components. Within this infrastructure, virtualised components are deployed in a distributed networking region including the access network, edge network, core network, and remote data centres. Figure 6.2 illustrates a typical forwarding graph of a network service in an LTE network. The network service is composed of multiple virtual network functions (VNFs): eNodeB, Mobility Management Entity (MME), Serving Gateway (SGW), and Packet Data Network Gateway (PGW), and Home Subscriber Server (HSS).

The adoption of such a distributed architecture for the network and its services introduces four major challenges when rolling out a network service:

1. The communication system facilitates various types of applications/services, namely voice/video calls, audio/video streaming, web surfing, and instant messaging. This introduces a high **complexity in understanding individual network services and associated dynamic workloads**.
2. The placement and autoscaling of VNFs are needed by the communication system in order to enable dynamic resource provisioning. VNF components in control and user planes have different features and requirements. As such, to fully address the placement and auto-

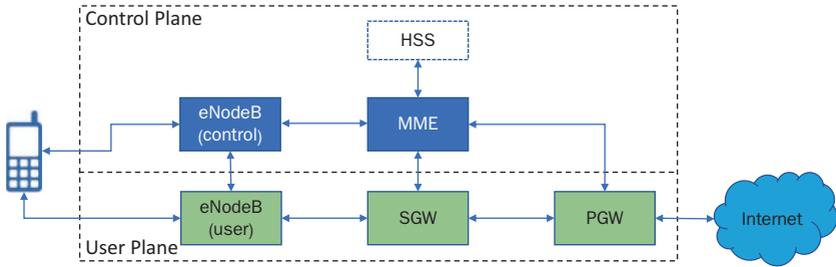


Fig. 6.2 A forwarding graph of a network service in an LTE network

scaling, it is necessary to understand and predict not only the variation of workload and resource utilisation but also the characteristics of the components. **Diversity in requirements and implementation, together with the dispersion of components across the network infrastructure, makes placement and autoscaling of VNFs a significant challenge.**

3. User behaviour needs to be explored for accurate workload predictions. To obtain knowledge of user behaviour, data communicated in control and user planes need to be analysed, and correlations thoroughly investigated. **This analysis is challenging, as one requires domain knowledge regarding behaviour of network services and the telecommunication network** more broadly.
4. Multi-tenancy is demanded in emerging 5G mobile networks where multiple network services are deployed and operating on top of a shared infrastructure. Different communication services come with different QoS requirements that desire a capability of adaptation and prioritising in resource allocation and management. In short, **5G brings complexity in the shape of mixed criticality and scale.**

### 6.2.3 Implementation

#### 6.2.3.1 Requirements

To fully address all the aforementioned challenges, it requires a complete control loop from data collection and analysis to optimisations on both infrastructure and application levels, and further up to the deployment of

optimisation plans. This places an overall requirement on a system, such as RECAP, to enable a wide range of automation tasks as listed below:

- Profiling network/service functions and infrastructural resources
- Automated service and infrastructure deployment
- Automated orchestration and optimisation of services and the infrastructural resource planning and provisioning
- Observability of behaviours of the system and services at run-time

To ensure end-to-end QoS (and by inference required service availability and reliability) in a complex large-scale use case, the control loop together with automated solutions need to be capable of resource planning and provisioning in the short- and long term, e.g. in minutes or in months. The solutions are also required to satisfy constraints of multi-tenancy scenarios and multiple network services competing for shared infrastructural resources. In addition, the optimisation aspects of the solution need to manage resource provisioning that achieve utilisation and service performance goals. Moreover, the simulation aspects of the solution need to support evaluation, i.e. impact of changes in planning rules prior to any real deployments.

Table 6.2 summarises the requirements. For each requirement, a set of targeted solutions is presented to illustrate the requirements are met. A simplified mapping is presented but the solution for any single requirement could be derived from one or a combination of multiple solutions listed.

### 6.2.3.2 *Implementation*

To demonstrate and validate the RECAP approach, a software/hardware testbed in Tieto is used. The testbed, deployed in a lab environment, emulates a real-world telecommunication system to facilitate the development and evaluation of optimisation solutions for end-to-end communications in a 5G network and its applications. Figure 6.3 presents an overview of the testbed in which a distributed software-defined infrastructure is emulated. This is achieved with heterogeneous resources collected from multiple physical infrastructures, located in a wide range of vertical regions, to provide communication and contents services to various applications that form different network services.

From the testbed's perspective, the entire RECAP platform is represented through the RECAP Optimiser, an external component that

**Table 6.2** Use case requirements and corresponding RECAP solutions

<i>Requirement</i>	<i>RECAP solution</i>
<ul style="list-style-type: none"> <li>• Allocation of infrastructural resources to uphold the QoS of a VNF for a provisioning of resource-efficient products</li> <li>• Allocation of a right amount of resources at right locations</li> <li>• Automation of instant capacity checks to support the rollout of new communication services in a timely fashion</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Workload and workload propagation models</i> enable estimations of bandwidth and resource utilisation for each VNF, service functions at application and infrastructure levels</li> <li>• The <i>load translation mapping model</i> enables quantification of <i>infrastructural resource utilisations</i></li> <li>• <i>Models for QoS metric assessment</i> are integrated with the above to fulfil all requirements</li> </ul>
<ul style="list-style-type: none"> <li>• Automation of the optimisation in VNF deployment and autoscaling for service availability and reliability and the minimisation of the overhead and resource utilisation of communication services</li> <li>• Automation of service remediation and infrastructure recovery to uphold required service availability and reliability</li> </ul>	<ul style="list-style-type: none"> <li>• The consolidation of aforementioned models and <i>optimisation models</i> facilitates the production of optimisation plans and recommendations for system autoscaling</li> <li>• The <i>RECAP platform</i> with automated optimisers empowers the realisation and execution of optimisation plans.</li> </ul>
<ul style="list-style-type: none"> <li>• Predictions of future infrastructural resource demand for resource planning and provisioning in a proactive manner</li> <li>• Detection of resource overbooking for VNFs and service functions to serve optimisation of the resource allocation</li> </ul>	<ul style="list-style-type: none"> <li>• Workload-related models and load translation mapping model enable the predictions of future workload and resource demands</li> <li>• A combination of quantification and predictions of resource demands facilitates overbooking detection</li> </ul>
<ul style="list-style-type: none"> <li>• Support infrastructure and communication service providers to maximise the utilisations of shared infrastructures</li> <li>• Method to prove a VNF is behaving as required on shared infrastructures</li> </ul>	<ul style="list-style-type: none"> <li>• Optimisation plans produced by optimisers with a consideration of scenarios of multi-tenancy and multiple network services enable the maximisation of the utilisation of <i>shared infrastructural resources</i></li> <li>• The aforementioned models combined with telemetry enable VNF and service function performance monitoring and management</li> </ul>

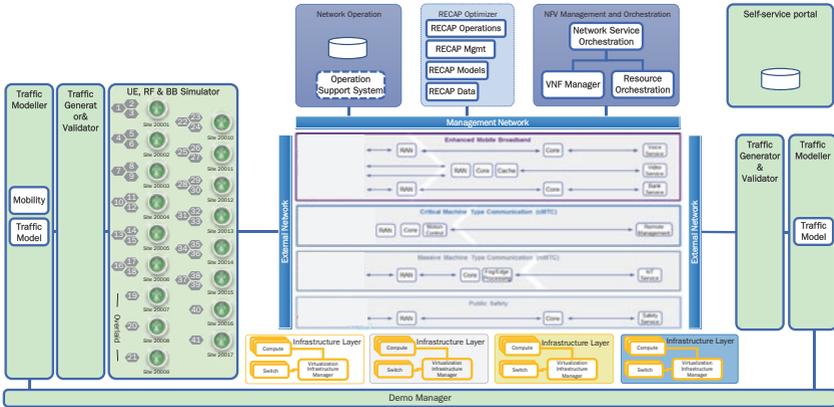


Fig. 6.3 Logical view of the testbed

produces and enacts optimisation plans (through its enactor) on the testbed. Each plan presents the in-directions of VNF placement and autoscaling across the emulated network infrastructure and is executed by the testbed. Ultimately, results, in terms of both application and infrastructure performance, are collected and evaluated. These results are also fed back to the RECAP platform for further investigation and model improvement.

### 6.2.3.3 Deliverables and Validation

To facilitate validation, multiple validation scenarios covering all the requirements presented in Table 6.2 were defined:

- **Scenario 1:** the placement and autoscaling of VNFs to fulfil QoS constraints required by a given single communication service.
- **Scenario 2:** the placement and autoscaling of VNFs to fulfil QoS constraints required by multiple communication services under multi-tenant circumstances.
- **Scenario 3:** the capability of RECAP simulation and optimisation tools in supporting the offline initial dimensioning the (physical) infrastructure according to traffic demands.
- **Scenario 4:** the capability of RECAP simulation and optimisation tools supporting the offline planning by identifying future (physical) infrastructure needs.

**Table 6.3** RECAP deliverables to address validation scenarios

<i>Model/component</i>	<i>Usage</i>
Workload model	To facilitate the implementation of the optimisers as well as optimisation solutions/plans in order to accomplish all the validation scenarios
Application model	
Infrastructure model	
Load translation model	
Infrastructure optimiser	To produce optimisation plans of VNF deployment across the network infrastructure; to directly support validation Scenarios 3 and 4, and together with the application optimiser to address all the scenarios
RECAP optimisation platform	To orchestrate all models and components and enact optimisation plans of VNF deployment that are fed to the testbed

- **Scenario 5:** the observability and fulfilment of given QoS requirements from the VNF level put in the resources provided by the infrastructure.

The relevant models and components that form the solutions to be validated against these scenarios are summarised in Table 6.3.

The application model is developed based on the network services deployed in the Tieto testbed, and workload models are constructed using the synthetic traffic data collected from various experiments carried out within the testbed. The infrastructure network model pertains to the city of Umeå in Sweden but is influenced by BT's national transport network and includes four network tiers (MSAN, Metro, Outer-Core, and Inner-Core). The network topology of the infrastructure is kept symmetrical, without including customisation for real-world aspects for asymmetrical node capacity, for asymmetrical node interconnection, and for asymmetrical link latencies.

### 6.2.4 Results

This section presents exemplar validation results for the application placement and infrastructure optimisation (Chap. 4). It addresses the problem of VNF placement across the network infrastructure. For the case study,

the RECAP Simulator (Chap. 5) was used to calibrate the models used by the Infrastructure Optimiser.

In the experiments, given a network service (Fig. 6.2), the eNodeB is deployed as two separate units on different planes: the Central Unit-User plane (CU-U) and the Central Unit-Control plane (CU-C). Additionally, the SGW and PGW VNF components are located on the user plane and are termed Service Gateway-User plane (SGW-U) and Packet Data Network Gateway-User plane (PGW-U).

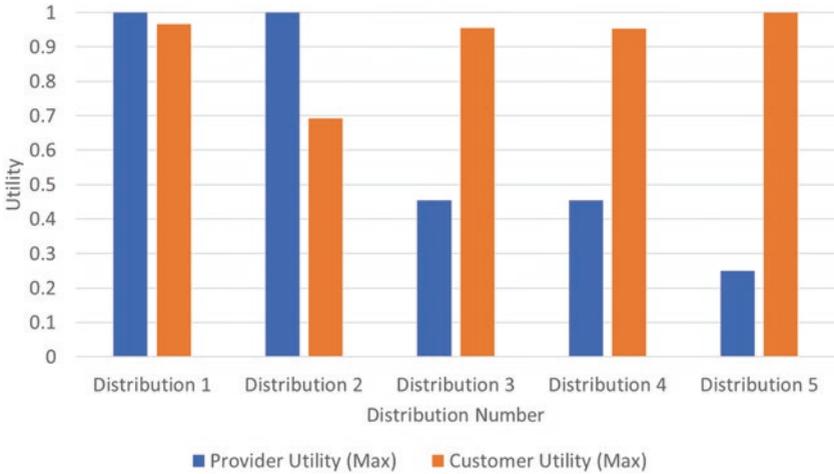
The optimisation solutions presented address Scenarios 3 and 4 concerned with placement and infrastructure optimisation. Five placement plans/distributions are identified as the input to the algorithm assuming (1) one forwarding graph per MSAN tier with CU-U as the user-request entry point, and (2) no currently deployed infrastructure. Table 6.4 describes these five placement distributions.

Results obtained are evaluated based on a comparison of provider and customer utility.

In Fig. 6.4, maximum provider vs. customer utility is normalised [0,1] for all distributions. Distributions 1 and 2 only use physical hardware

**Table 6.4** Initial placement plans of VNFs

<i>Placement plan</i>	<i>Description</i>
Distribution 1	<ul style="list-style-type: none"> <li>• CU-U, CU-C, SGW-U, and PGW-U VNFs placed on the MSAN resource sites</li> <li>• MME/SGW-C/PGW-C VNF placed on the Outer-Core resource sites</li> </ul>
Distribution 2	<ul style="list-style-type: none"> <li>• CU-U, CU-C, SGW-U, and PGW-U VNFs placed on the Metro resource sites</li> <li>• MME/SGW-C/PGW-C VNF placed on the Outer-Core resource sites</li> </ul>
Distribution 3	<ul style="list-style-type: none"> <li>• CU-U and CU-C VNFs placed on the MSAN resource sites</li> <li>• SGW-U and PGW-U VNFs placed on the Metro resource sites</li> <li>• MME/SGW-C/PGW-C VNF placed on the Outer-Core resource sites</li> </ul>
Distribution 4	<ul style="list-style-type: none"> <li>• CU-U and CU-C VNFs placed on the MSAN resource sites</li> <li>• SGW-U, PGW-U, and the MME/SGW-C/PGW-C VNFs placed on the Outer-Core resource sites</li> </ul>
Distribution 5	<ul style="list-style-type: none"> <li>• CU-U VNF placed on the MSAN resource sites</li> <li>• CU-C and SGW-U VNFs placed on the Metro resource sites</li> <li>• PGW-U and the MME/SGW-C/PGW-C VNFs placed on the Outer-Core resource sites</li> </ul>



**Fig. 6.4** Maximum provider and customer utility of each distribution

across two tiers (as per Table 6.4) and hence exhibit the highest provider utility; this decreases by approximately 50% as the distributions start to include more than two tiers in the placement. This is understandable as more infrastructure needs to be deployed and maintained. Customer utility is highest for Distribution 5, which has three tiers MSAN, Metro, and Outer-Core (as per Table 6.4) included in the distribution. The lowest utility is when no VNFs are placed at the edge, i.e. Distribution 2, which has no MSAN; this is to be expected as the end-user request faces larger processing latency in travelling further into the network.

Figure 6.4 maps the normalised provider utility and normalised customer utility of each VNF placement. The figure shows that a provider can manage its deployments by fixing the provider utility or customer utility in a way that balances business considerations.

For example, in Fig. 6.5 the provider utility is centred on 50% to ensure customer utility is centred on 75%. The intersection of threshold lines (the highlighted section in grey) identifies a set of placements that are optimal for each individual forwarding graph of the use case while satisfying defined constraints including application and infrastructure provider perspectives. The provider could choose Distribution 1, 2 or 4. However, Distribution 2 has poorer customer utility (no MSAN, higher latency) and so is disregarded. Distributions 1 and 4 utilise MSAN and Outer-Core

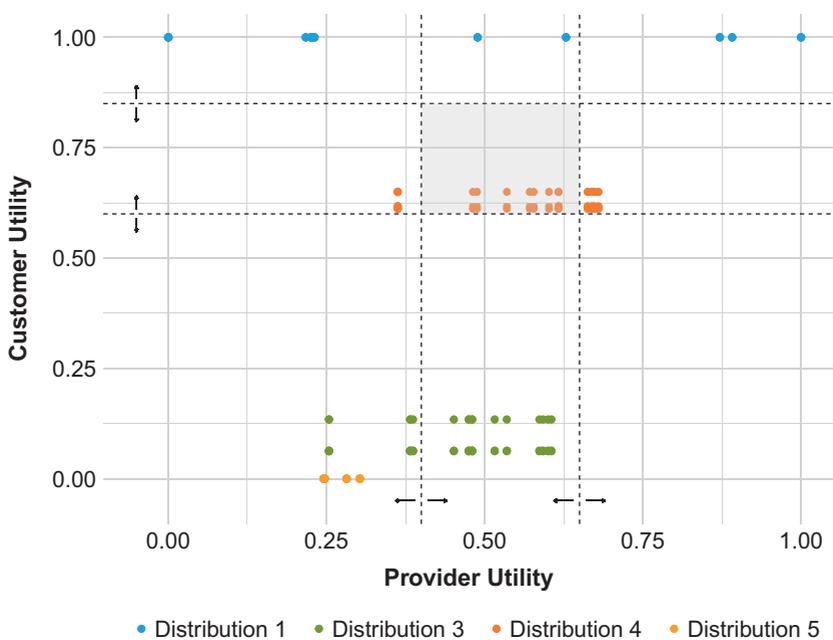


Fig. 6.5 Provider utility vs. customer utility for different distributions

infrastructure and have comparable customer utility. However, Distribution 1 has the higher provider utility, and thus would be the best option with caveats. It would be the best option if “consolidation” was the most important factor to the business, but not necessarily the best option if “flexibility” to service future requests was more important. In the latter case Distribution 4 is a better option because one’s current customer is satisfied (compared to Distribution 1) but the provider has significant up-swung capacity.

Figure 6.6 below illustrates simulation results for the same distributions without Distribution 2 which was disregarded due to no MSAN capacity. The graph represents all infrastructure and all remaining distributions. Utility is combined for simplicity (y-axis) and is graphed against 3 scenarios (x-axis) “normal day”, “event”, and “24% growth”. It should be more apparent that for the same scenarios and constraints as above, Distributions 1 and 4 remain the best options. Distribution 1 remains the best option and for this simulation exercise could cope with the defined *event* and *growth* scenarios. But what is a little less obvious is that its utility remained essentially static while

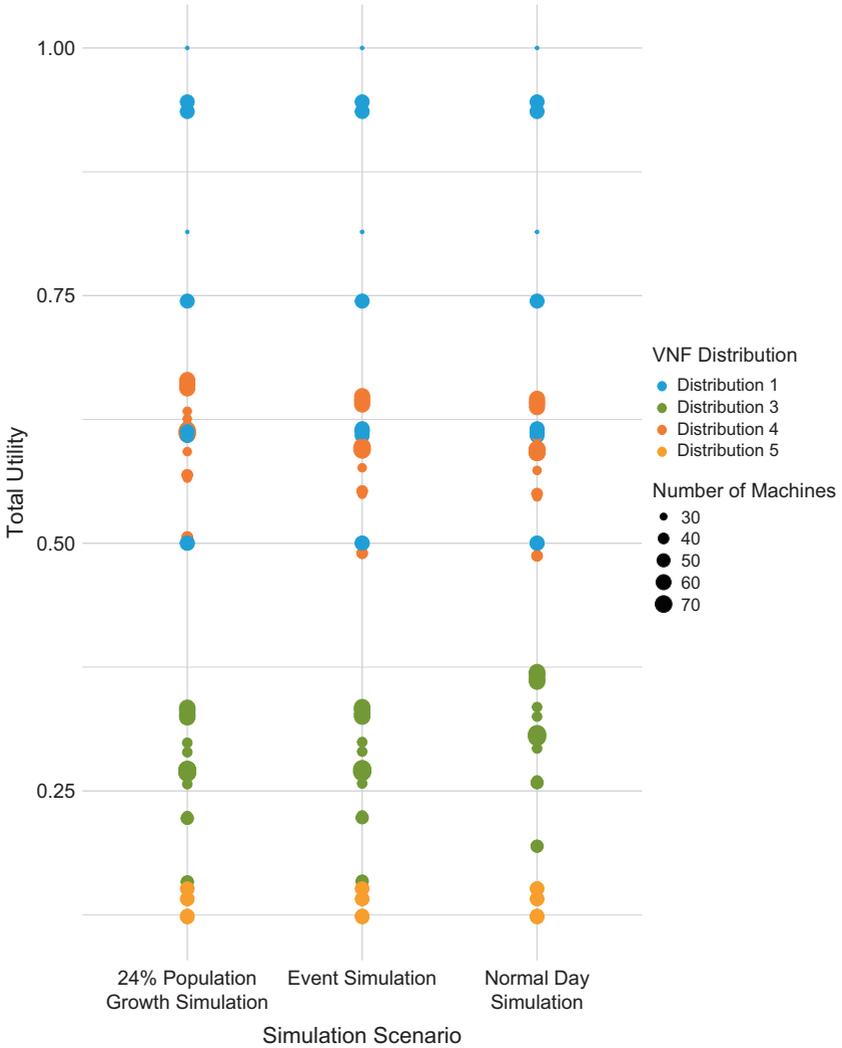


Fig. 6.6 Total utility for normal day, event, and 24% growth scenarios

the utility for Distribution 4 starts to trend upwards from the *normal day* to *event* to *24% growth* scenario. This is primarily driven by provider utility improving as utilisation of physical assets improve; this scenario offers considerably greater capacity for future *growth/event* scenarios.

### 6.2.5 *Summary*

For this case study, an extensive evaluation and validation were performed for the scenarios outlined and utilising the models, components, and technologies described in Chaps. 1, 2, 3, 4, and 5. Current results contribute to more effective decision making for infrastructural resource dimensioning and planning for future 5G communication systems. The value presented is supporting informed, automated (if desirable) decisions similar to the consolidation vs. flexibility example illustrated in the Distribution 1 versus Distribution 4 options above.

## 6.3 CASE STUDY IN NETWORK FUNCTIONS VIRTUALISATION: VIRTUAL CONTENT DISTRIBUTION NETWORKS

### 6.3.1 *Introduction*

Network Function Virtualisation (NFV) replaces physical network appliances with software running on servers. Content Distribution Networks (CDNs) offer a service to content providers that puts content on caches closer to the content consumers or end users. Traditional Content Distribution Network (CDN) operators install customised hardware caches across the globe—sometimes within an Internet Service Provider’s network and sometimes in third-party co-located data centre facilities. Each CDN operator develops its own caching software with unique features, e.g. transcoding methods, management methods, and high availability solutions.

As of now, network operators such as BT have hardware from each of several CDN operators deployed at strategic points in its network. However, this creates several potential issues:

- it is hard to organise sufficient physical space (in “telephone exchange” or “central office” buildings for instance) to support all the CDN operators;
- a lot of energy is needed to power and cool all the equipment; and
- a lot of physical effort is needed when a new CDN operator arrives or an existing one disappears.

Such factors make it attractive to consider a Virtual CDN (vCDN) approach that aims to replace the multiple customised physical caches with

standard servers and storage running multiple virtual applications per CDN operator. This lowers CDN and network operator costs and allows the content caches to be put closer to the consumer, which improves customer experience. Also, the barriers to entry for new virtual CDN operators are likely to be lower than for a physical CDN operator.

### 6.3.2 *Overview and Business Setting*

Broadband traffic on BT's network of 50% of broadband traffic on BT's network originates from the content caches operated by the CDN operators. At the time of writing, BT hosts CDN operators customised cache hardware in two to eleven compute sites in the UK to reduce the amount and cost of Internet peering traffic. If the caches were installed in BT's thousand edge nodes (also known as Tier-1 MSANs (Multi-Service Access Nodes) or "Telephone Exchanges"), the cost of delivering content would be reduced by 75% and BT would reduce its network load significantly. However, the CDN operators are unlikely to want to install their hardware at up to a thousand locations in the UK; for some international CDN operators, a single compute site in London is sufficient for the entire UK.

The vCDN proposition is that BT could install the compute infrastructure at its edge sites and offer an Infrastructure-as-a-Service (IaaS) offering tailored towards CDN operators. The CDN operators would install and manage their own software on BT IaaS and thus they would maintain their unique selling points and ownership of the content provider customers. This is a potential win-win scenario: the network and CDN operators reduce operating costs and consumers get better service (Table 6.5). There are, however, several technical challenges to designing and operating a vCDN service, not least performance, orchestration, optimisation, monitoring, and remediation. These are discussed later in Sect. 6.3.3.

An abstract representation of the BT UK network topology is shown in Fig. 6.7. The real locations of BT network sites are shown in Fig. 6.8; the black dots represent BT's 5600 local exchanges, of which c. 1000 are MSANs and c. 100 BT's Metro sites (Ofcom 2016). The 4500 local exchanges that are not MSANs are considered unsuitable for deployment of caches as they do not contain enumerates the most important vCDN use case requirements.

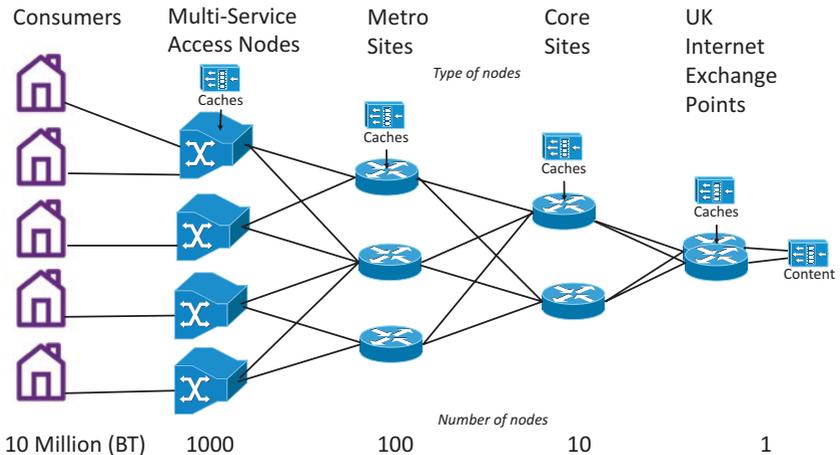
**Table 6.5** vCDN use case requirements and corresponding RECAP components

<i>Requirement</i>	<i>RECAP solution</i>
<ul style="list-style-type: none"> <li>• Optimise cost of compute and storage infrastructure vs. cost of network bandwidth.</li> <li>• Take account of uneven distribution of consumers and traffic.</li> <li>• Different CDN operators may have different optimal locations for their caches</li> <li>• Network operator must take into account the potential demands across multiple CDN operators</li> <li>• Content traffic has a 2:1 peak-to-mean ratio and is highly seasonal with daily, weekly, and annual patterns. Power could be saved by turning off infrastructure when not required.</li> <li>• CDN operators will need tools to support a near real-time decision to activate or deactivate their caches.</li> </ul>	<ul style="list-style-type: none"> <li>• CDN traffic forecast using a Seasonal Autoregressive Integrated Moving Average (SARIMA) model enables accurate prediction of demand.</li> <li>• CDN load translation models to calculate how much compute resource is required for the forecast workload.</li> <li>• Cache placement optimisation using state-of-the-art AI and genetic algorithms.</li> <li>• CDN application model to calculate where caches should be placed in the BT topology.</li> <li>• Infrastructure model of BT's network and compute infrastructure.</li> <li>• Simulator to calculate utility of infrastructure placement options.</li> <li>• The RECAP methodology addresses this (further work is required to add customer distribution into the optimisations).</li> <li>• The application model takes account of BT's topology.</li> <li>• Application model can be run per CDN operator.</li> <li>• Infrastructure model can aggregate demand from the application model.</li> <li>• Application cache placement optimiser is dynamic and can adjust according to traffic load.</li> </ul>

*(continued)*

**Table 6.5** (continued)

<i>Requirement</i>	<i>RECAP solution</i>
<ul style="list-style-type: none"> <li>Total bandwidth consumed by content is consistently increasing (c. 50% per annum); therefore, the network operator needs to constantly invest in adding more transmission or more vCDN infrastructure nodes and capacity to the network. Network operators need to improve the accuracy of future traffic predictions and where investments should be made so that infrastructure gets installed just in time and customer experience is always good.</li> </ul>	<ul style="list-style-type: none"> <li>The collection of RECAP workload predictors, models, and optimisers find solutions, which optimise cost and performance according to network operator preferences.</li> </ul>

**Fig. 6.7** Abstract representation of BT UK network topology

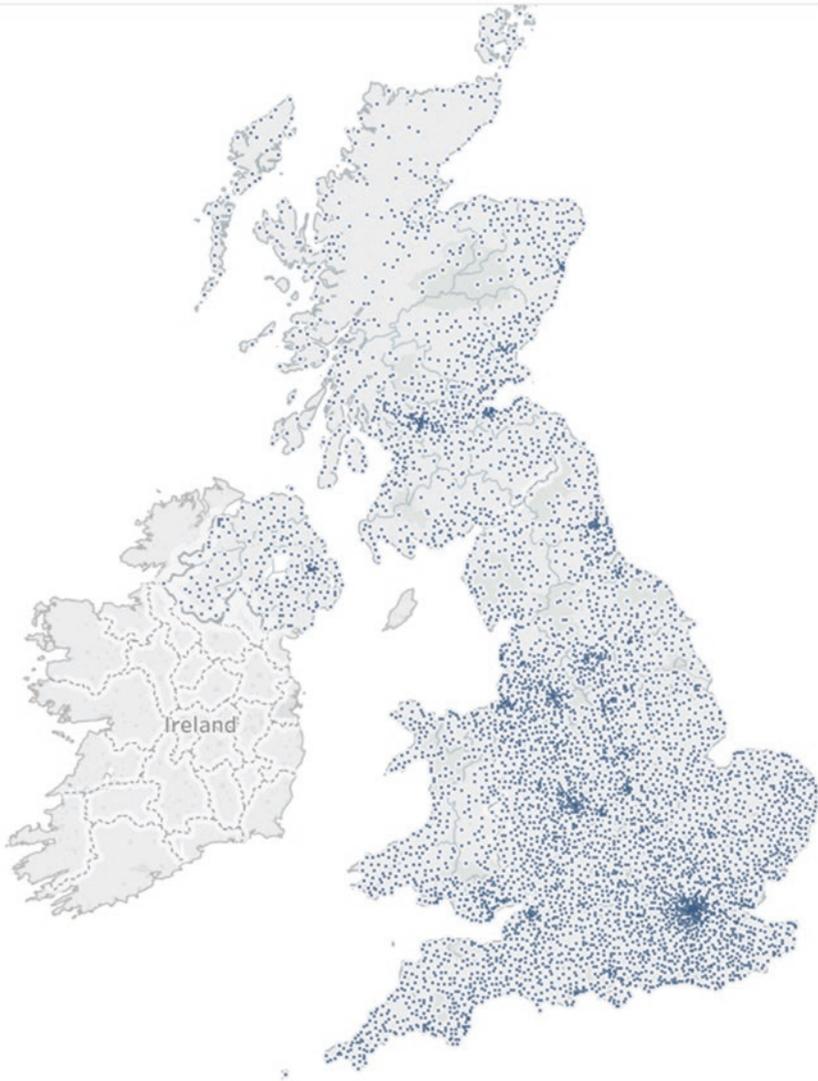


Fig. 6.8 BT network locations in UK

### 6.3.3 Technical Challenges

The vCDN technical challenges can be grouped into several areas as shown in Table 6.6 below.

**Table 6.6** vCDN technical challenge and RECAP solution

<i>Technical challenge</i>	<i>RECAP solution</i>
Performance of virtualisation technologies, especially virtualised storage.	BT is conducting experiments to verify and improve the performance and orchestration of CDN virtualisation technologies.
Orchestration of a multi-tenant vCDN service and infrastructure.	BT is building an orchestration system proof of concept using OSM and OpenStack for its vCDN service.
<ul style="list-style-type: none"> <li>• Optimisation of placement and scaling of vCDN system.</li> <li>• BT decides where to install and grow or reduce the infrastructure.</li> <li>• The CDN operators decide where to install, grow, and reduce their virtual machines or containers for their caches.</li> <li>• BT will have to optimise how much of its infrastructure it dedicates to CDN operators.</li> <li>• CDN operators will have to optimise how much resource they need to consume.</li> <li>• Each CDN operator will be independent and may experience different traffic loads.</li> <li>• Installing infrastructure requires the planning of the provisioning of hardware many months in advance, although once installed servers may be turned on and off, to reduce power consumption, in ~15 minute time periods.</li> <li>• CDN operators may activate and deactivate their cache virtual machines or containers very quickly in the 5 minutes to sub-second timeframes and hence need more real-time optimisation than the infrastructure.</li> <li>• CDN operators may choose to deactivate their software to reduce infrastructure rental charges.</li> </ul>	<p>RECAP methodology can automate the decision making for the optimisation, scaling, monitoring, and repair of vCDN systems using modelling and statistical techniques.</p> <ul style="list-style-type: none"> <li>• RECAP methodology is built around “separation of concerns” addressing the need for network and CDN operators to be treated separately.</li> <li>• RECAP forecast model can enable decisions to be made and acted upon just in time to optimise power consumption.</li> <li>• RECAP application model adjusts to traffic dynamics.</li> </ul>

(continued)

**Table 6.6** (continued)

<i>Technical challenge</i>	<i>RECAP solution</i>
<ul style="list-style-type: none"> <li>• Monitoring and repair of the vCDN system.</li> <li>• Each CDN operator will have proprietary methods for the monitoring and remediation of their CDN software.</li> <li>• Many operate an architecture that is fault tolerant, with a caching hierarchy, where loss of a leaf will result in content being served from a cache higher in the hierarchy.</li> <li>• CDN operators also have advanced load balancing mechanisms where a consumer's initial request is switched to the best cache, according to load and location, and content is "chunked" and distributed so any failure mid-session will be recovered from.</li> <li>• Detection and mitigation of impact of "noisy neighbours".</li> </ul>	<ul style="list-style-type: none"> <li>• The RECAP methodology can automate the decision making for the optimisation, scaling, monitoring, and repair of vCDN systems using modelling and statistical techniques</li> <li>• The CDN operators' architectures permit re-optimisation of CDN cache locations and scale with minimal impact on service.</li> <li>• Proprietary solutions exist to monitor the quality of the content delivered to consumers.</li> <li>• Further work is required to feedback these quality measurements into a CDN cache placement optimisation and orchestration solution.</li> </ul>

### 6.3.4 *Validation and Impact*

The RECAP consortium is engaged with various CDN operators to develop fine-grained infrastructure and application models to develop optimisation strategies for Virtual Content Distribution Networks (vCDN). The resulting strategies will aid BT to improve the accuracy of their planning and forecasting, reducing infrastructure investment while still giving BT's customers a superior web browsing and video streaming experience. RECAP methods will reduce the amount of human support BT's vCDN planning process requires, enabling BT to be more agile and cost efficient (Fig. 6.9).

Preliminary experimentation results are promising regarding the utility of RECAP for BT. They suggest:

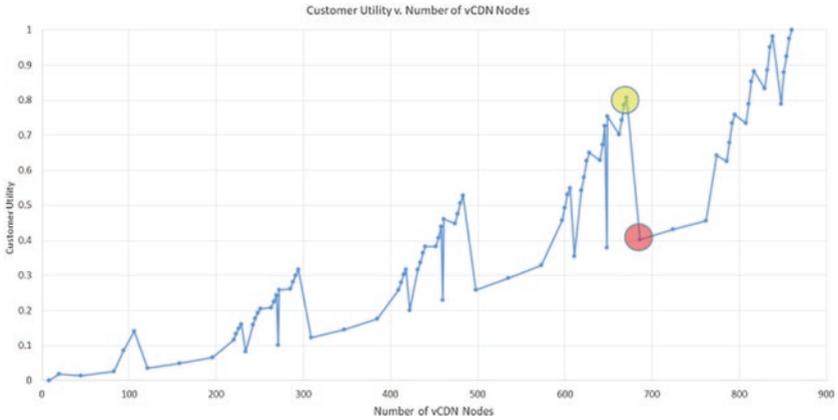
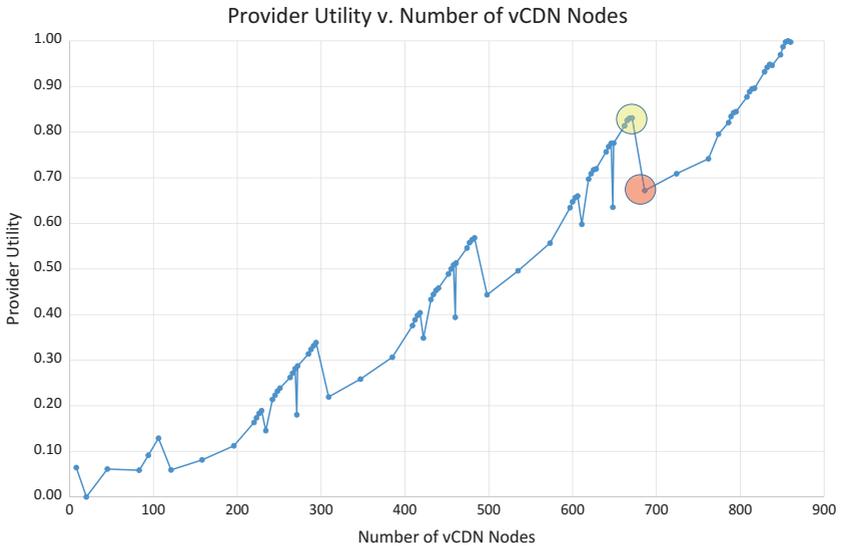


Fig. 6.9 Customer utility vs. number of vCDN sites

1. The SARIMA models provide one-hour ahead workload forecasts of 11.5% accuracy with 90% confidence. This should be sufficient for CDN operators to pre-emptively adjust the sizes and number of caches running while infrastructure operators should be able to shut down or power up servers to minimise power wastage.
2. The RECAP DTS framework demonstrated the value of a caching hierarchy when compared to a single layer of caches at the MSAN. The results suggest a caching hierarchy can improve Provider Utility by up to 24% (see Fig. 6.10) and doubles Customer Utility at intermediate stages of infrastructure deployment, as illustrated in Fig. 6.9. Further, when compared to BT's own optimisation strategy, it improved Provider Utility by up to 6.4% at certain intermediate stages of infrastructure deployment. The BT and RECAP optimisation both converged on deploying the maximum number of 860 nodes because the caching business case is very compelling, i.e. deploying a cache at a site always saves money and improves customer utility. For a use case where the business case is more marginal, the RECAP methodology can find a solution more significantly optimal.
3. The RECAP Application Optimisation (autoscaling and simulation) systems provide BT with the ability to both improve on baseline cache deployment scenarios (through comparative analysis of alternative deployment scenarios) and in run-time adapt to unfore-



**Fig. 6.10** Provider utility vs. number of vCDN nodes

seen and unexpected changes in workload. Simulation allows experimentation with alternative deployment strategies and evaluates the impact of changes in infrastructure, as well as application topologies and caching strategies. Validation experiments demonstrate a 4% improvement in cache efficiency when serving realistic workloads as well as the ability to efficiently adapt the amount of cache capacity deployed in heterogeneous and hierarchical networks to changes in request and network traffic patterns.

These results suggest that by implementing RECAP, BT and CDN operators can benefit from both decreased cost and increased competitiveness through:

1. Providing more accurate modelling, infrastructure dimensioning, and resource allocation across the chain of service provision to support better infrastructure planning.
2. Rapid accurate autoscaling to support fluctuations in demand and avoid under and over booking of resources.

3. Leveraging existing infrastructure and avoiding additional capital expenditure.
4. Reducing staffing requirements and freeing up valuable IT expertise.
5. Increased revenue through:
  - (a) Delivering and maintaining high QoS.
  - (b) Shortening the time for CDN operators to access infrastructure and accelerate revenue generation.
  - (c) Reducing time to market for infrastructure and applications deployment.

## 6.4 CASE STUDY IN EDGE/FOG COMPUTING FOR SMART CITIES

### *6.4.1 Introduction*

This case study integrates RECAP mechanisms related to resource reallocation and optimisation in a proprietary distributed IoT platform for smart cities, called SAT-IoT. Hence, it demonstrates the capabilities to integrate RECAP components into third-party systems as well as the immediate benefits of using the RECAP approach for optimisation.

Being IoT-centric, this case study deals with hardware-software infrastructures and vertical applications as well as mobile entities and devices that move over the area of city. It is built on the assumption that smart cities provide infrastructure for handling IoT network traffic in a zone-based manner as shown in Fig. 6.11; wireless networks complement wired networks to form a hybrid network (Sauter et al. 2009); and these hybrid networks include the cloud nodes, edge nodes (IoT gateways), and further mid (fog) nodes. Mid nodes are connected to the cloud and to each other forming a mesh network. Edge nodes receive data from wireless devices located in the same geographical area. Groups of edge nodes are connected to a mid node. Edge nodes are usually not connected to each other. In this kind of scenario, it is necessary to manage the IoT network topology to adapt to moving users and changing data streams. Such a topology administration will facilitate the dynamic deployment of distributed IoT applications, the interconnection of devices in the IoT platform, and the data exchange among platform network nodes.

Consequently, this use case study requires:

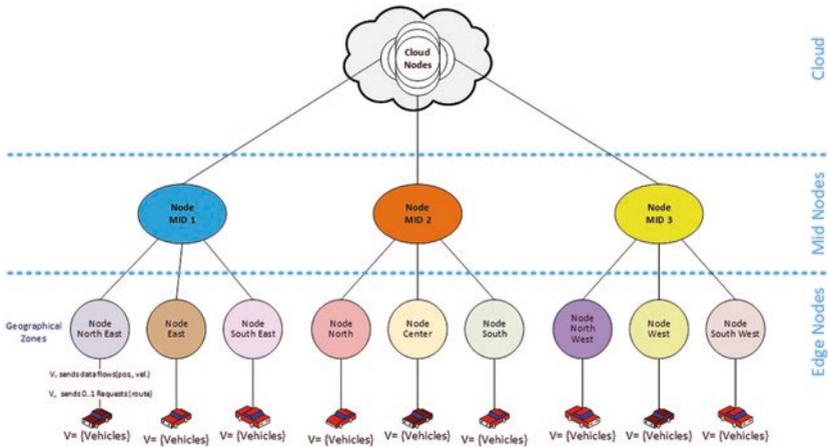


Fig. 6.11 Example of IoT hybrid network for mobile devices

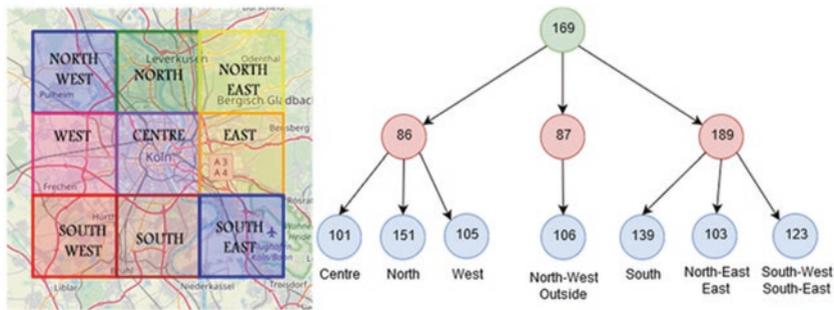


Fig. 6.12 Smart city structure

- The capability to dynamically optimise the applications’ communication topologies under an “Edge/Cloud Computing Location Transparency” model. In particular, it requires the optimal reallocation of data flows periodically at run-time to reduce bandwidth consumption and application latencies.
- A mechanism to consider limitations of the physical topology when planning the virtual topology.

For the sake of demonstration and validation purposes, the SAT-IoT platform in RECAP is running a distributed Route Planning and City Traffic Monitoring application. Figure 6.12 illustrates this scenario based

on the city of Cologne. The city area is distributed into nine geographical regions, each of which has its own edge compute node.

### 6.4.2 *Issues and Challenges*

The formulated scenario presents some challenges with respect to the current cloud IoT/smart city systems, the edge/fog computing paradigm, and the ICT architecture optimisation. The issues and challenges addressed in this case study include:

- Adoption of edge computing models to process the massive data generated from different IoT devices at their zone edge nodes in order to improve performance.
- Integration of edge, fog, and cloud paradigms to develop dynamically configurable IoT systems, to achieve better optimisation results for both applications and underlying infrastructure.
- Dynamic management of smart city environments based on distribution of mobile devices and users and their resource demands.

### 6.4.3 *Implementation*

#### 6.4.3.1 *Requirements*

In order to improve the performance of the IoT system, the edge computing model seeks to process the massive data generated from different IoT devices at their zone edge nodes. Only the processing results are transmitted to the cloud infrastructure or to the IoT devices, reducing the bandwidth consumption, the response latency, and/or the storage needed (Ai et al. 2018).

Considering an IoT system that uses a hybrid network similar to Fig. 6.11, any application that processes data from zones, North and Centre, cannot naively run data processing in the zones since it requires information from both zones. A conventional cloud computing architecture is not well suited to applications where the location of devices changes, where the volume of data received in each edge node varies dynamically, or where the processing needs data from different geographical areas. Car route planners and city traffic analysers are good examples of smart city applications that make calculations with the information received from connected cars located in different zones of the city.

Thus, to process the data from, for example, the North and Centre zones, it is necessary to send it to a processing node. As these edge nodes are not connected to each other, the data from one or both zones need to pass through Mid86. Indeed, Mid86 would be the closest node to run processing for both affected zones. When considering the scenario on a larger scale with multiple zones and applications that require data from all zones to complete the processing,<sup>1</sup> the task of finding the most suitable processing node is non-trivial. Furthermore, where there are significant constraints on deployment capacity constraint, the underlying infrastructure becomes a factor too. Consequently, an IoT platform for smart city applications must be able to:

- integrate of cloud, fog, and edge computing models;
- manage the smart city data network topology at run-time;
- use optimisation techniques that support processing aggregated data by geographical zones; and
- monitor the IoT system and the optimisation process in run-time.

#### 6.4.3.2 *The SAT-IoT Platform*

The IoT platform which forms the basis of this case study is based on the SAT-IoT platform. Its core architectural concept is edge/cloud computing location transparency. This computational property allows data to be shared between different zones and to be processed at any of the edge nodes, mid nodes, or cloud nodes.

The concept of edge/cloud computing location transparency is realised by two of the entities in the SaT-IoT architecture, the IoT Data Flow Dynamic Routing Entity, and the Topology Management Entity (see Fig. 6.13). They support a cloud/fog programming model with the capability of managing the network topology at run-time while also providing the necessary monitoring capabilities to understand the usage pattern and capacity limitations of the infrastructure. While they provide the necessary capabilities to reconfigure the topologies and data flow, they lack the capability to derive the best-possible placement of the data processing logic. This is realised by integrating the RECAP Application Optimiser in the SAT-IoT platform.

<sup>1</sup> SAT-IoT is capable of managing and supporting multiple applications over the same IoT data network.

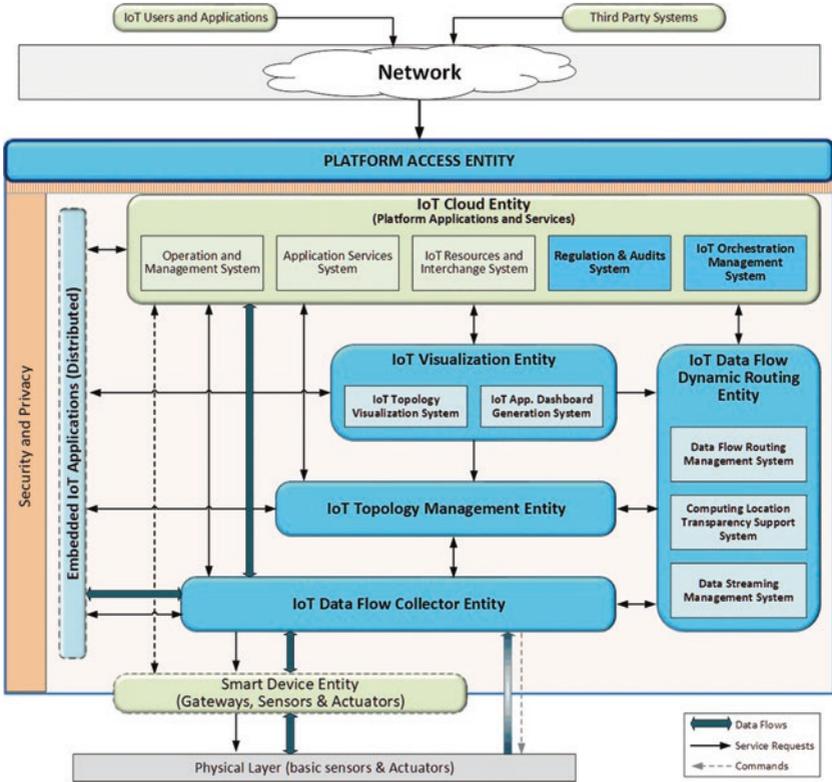


Fig. 6.13 SAT-IoT platform architectural model

### IoT Topology Management Entity

IoT Data Flow Dynamic Routing is the cornerstone of SAT-IoT. It dynamically manages IoT data flows between processing nodes (cloud nodes, edge nodes, and smart devices). In addition, this entity includes a distributed temporary data storage system to support data streaming and local processing services. In this case study, data flows are both the sets of data sent by the cars (position, speed, fuel consumption, etc.) and the route calculation requests between two locations in the city.

The IoT Data Flow Dynamic Routing Entity comprises:

- **Data Streaming Management System:** Provides the mechanisms to transfer IoT data flows directly from nodes (e.g. edge nodes or smart devices) to other internal or external services and applications that request them (on a publish/subscribe model).
- **Computing Location Transparency Support System:** Wraps the RECAP Application Optimiser for its integration into the SAT-IoT platform.
- **Data Flow Routing Management System:** Responsible for setting the routing of the data flows to the optimum computation node after inquiring about the best computation node for the data flow from the Computing Location Transparency Support System.

### IoT Topology Management Entity

The IoT Topology Management Entity is responsible for the definition of an application network topology in every IoT system deployed by the platform. This application network topology defines which SAT-IoT entity communicates with which other SAT-IoT entities. The communication structure is based on the available underlying IT infrastructure (computation nodes and data network).

The application topology is defined as a graph of computing nodes and links between them, and it includes a variety of attributes like node features (CPU, Memory, etc.), data link features (bandwidth), geolocation of the node, use of resources (hardware and communication metrics), etc. With this definition, the system dynamically manages the physical hardware topologies, enables updating the logical structure of the topologies at any time, and includes a monitoring system that continuously provides the status of nodes and links in terms of performance metrics (consumption of CPU, memory, storage, bandwidth, etc., and also data flows crossing the network).

The three main functions of the **Topology Management Entity** are:

- **IoT Topology Definition:** It enables the modelling of the IoT architecture as an enhanced graph in which nodes are the hardware elements with processing capabilities. The nodes in the graph are defined containing all their attributes (node type, CPU, RAM, location, etc.). Edges in the graph correspond to data links and have their attributes as well.

- **Topology Management:** It is a set of services to query and modify the IoT topology definition to maintain the consistency between the physical installations and their definition in the platform. It supports the model of Edge/Cloud Computing Location Transparency supported by the platform.
- **Topology Monitoring:** It continuously gathers and stores metrics of each node and edge. It also provides these metrics to other internal systems (IoT Topology Visualisation System or Embedded Applications) and external systems (third-party applications and systems).

### Application Optimisation

To find the optimal location of the data processing logic, the optimiser needs to consider response latency, bandwidth consumption, storage, and other properties. Furthermore, the selection of the computation nodes might change dynamically as the conditions of the system may vary over the time (shared data, application requests, data volume, network disruptions, or any other relevant issue).

#### 6.4.3.3 Implementation

For realising optimisation support, SAT-IoT integrates a RECAP application optimisation algorithm. Using this algorithm, SAT-IoT can decide, in real time, the optimum node of the IoT data network to process a given data flow. The integration of the application optimisation algorithm is implemented in the Computing Location Transparency Support System module, part of the IoT Data Flow Dynamic Routing Entity (see Fig. 6.14).

The application optimisation algorithm uses IT resource optimisation techniques, graph theory (based on the topology graph definition), and machine learning processes to predict the needs of the system in the short term. The prediction considers the current state of the systems, e.g. metrics, IT resources used, links bandwidth consumption, application latencies, distribution of nodes across the topology, and the data flows involved in each node. A **RECAP Non-Dominated Sorting Genetic Algorithm** is used to calculate the optimum node to process the application data flows received in the last time period.

The Application Optimiser systematically receives the virtual topology and the data flows (route calculation requests and information about them) for the last time period. It then calculates a cost function to move the flows to each server and finally selects the node with the minimum

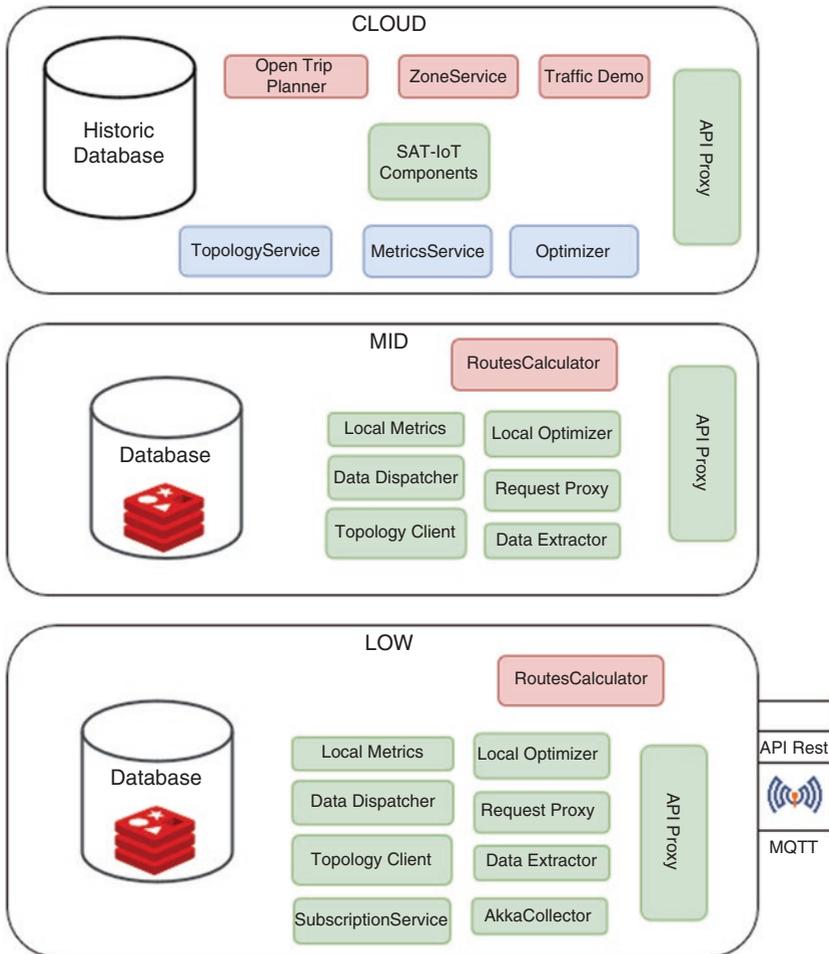


Fig. 6.14 SAT IoT platform high-level conceptual architecture

value of the cost function. The application receives the node selected and requests the platform to configure the data flow routing to send the data flows to the optimum computation node.

The city traffic monitoring application makes use of the optimisation service provided by SAT-IoT and provides a user interface to execute the optimisation on demand. In a production setting, the Application Optimiser would be run automatically based on intervals. Every time the

optimisation is executed, SAT-IoT automatically changes the virtual network routing configuration in order to send data flows to the optimum node for processing.

#### 6.4.4 *Validation Results*

To validate the system, SAT-IoT runs a distributed Route Planning and City Traffic Monitoring application using a dataset of Cologne city traffic edited by the Simulation of Urban Mobility (SUMO) Eclipse project as an input. Software entities emulate cars moving in the city and send data such as position, velocity, and road conditions captured from sensors periodically.

The IoT platform and application have been deployed in a virtual infrastructure with a topology as shown in Fig. 6.15 where seven edge servers associated to ten areas of Cologne city are used (nine city zones and an additional zone to cover traffic close to those defined zones). Three mid nodes have been deployed to group sets of three areas, and a group of virtual servers acts as the cloud infrastructure. Runs cover traffic simulation of two hours.

As discussed, SAT-IoT makes use of the RECAP Application Optimiser (see Fig. 6.13). Periodic optimisation is switched off and manual optimisation enabled to allow the user to execute the RECAP Application Optimiser on demand while the simulation scenario is running. This allows optimisations to be performed at different execution times where the conditions and status of the platform may vary. For instance, route requests and operational vehicles vary over time and executing the Application Optimiser at different points in time results in different optimisation results as shown at the top of Fig. 6.15. Here, the orange circle in the upper left diagram represents the optimal node for data processing at that point in time. Similarly, the chart at the upper right side shows the cost function for the selected node compared to cloud-based data processing.

In the lower left chart, Fig. 6.15 shows the values and results obtained from the optimisation process executed nine times during an experiment. In the first optimisation, the optimum destination node changes from the cloud to Node 86. The line chart on the right shows an immediate cost reduction. The table on the bottom left shows the cost saving/additional cost of moving the data to different nodes. In this case, the optimum node shows a reduction of 3311 cost points compared to moving the data to the cloud node. These results are evident in the bar chart at bottom right too.



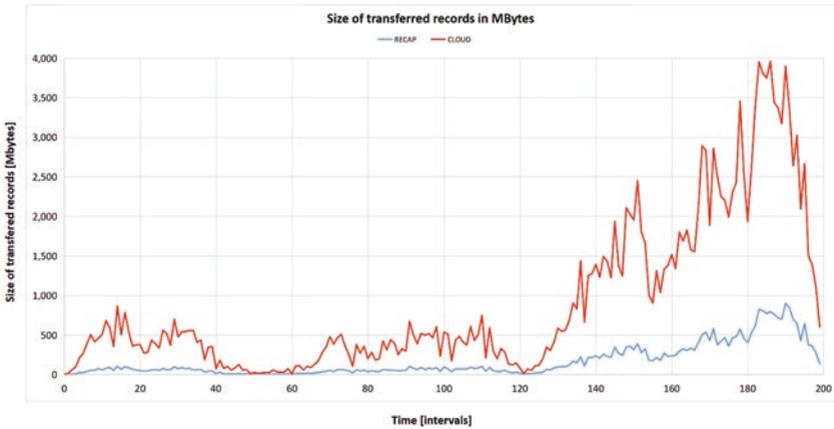
Fig. 6.15 Optimisation results

Figure 6.16 plots the overall amount of data transferred per time unit. It compares a standard cloud-based processing approach with an approach using RECAP-Optimised message routing. As can be seen, it shows a significant reduction of transferred data and hence bandwidth consumption.

### 6.4.5 Results

Implementation of the RECAP Application Optimiser (1) reduced bandwidth usage by up to 80% compared to a cloud-only processing of data, and (2) reduced the overall latency and improved the user experience by reducing the overall number of hops to send the data flows to the optimum processing node by up to 20%.

In summary, the benefits of the implementation of the RECAP Application Optimiser embedded in the SAT-IoT platform include (1)



**Fig. 6.16** Number of records transferred for SAT-IoT running route planning and city traffic monitoring application using cloud-based processing and RECAP-optimised processing

automated continuous optimisation; (2) enabling dynamic changes to computation nodes in an IoT network topology without administrator intervention to fulfil the efficiency criteria defined for the IoT system; and (3) global transparency across the entire IoT system, which together result in significantly reduced costs and increased quality of service.

## 6.5 CASE STUDY IN CAPACITY PLANNING: BIG DATA ANALYTICS SEARCH ENGINE

### 6.5.1 Introduction

This case study illustrates how components of the RECAP approach, namely data analytics as well as simulation and planning, can be used for capacity planning for and small-to-medium-sized enterprises.

Linknovate (LKN) is a Spanish SME that develops and markets a cloud-base data analytics and competitive intelligence platform and service. LKN's primary market is in the US. LKN generates knowledge insights by aggregating large amounts of (heterogeneous) research and scientific data using data mining and data analytics techniques for their clients. At the time of writing, LKN had indexed over 20 million documents, over 30

**Table 6.7** LKN platform components

<i>Nodes</i>	<i>Components</i>
Online—index and search nodes	ElasticSearch v5 providing the search functionality and storing processed and structured data: <ul style="list-style-type: none"> <li>• 9 data nodes (Azure DS12 v2) running Debian GNU/Linux with 4 processing cores and 28GB of RAM, where 24 shards of data with replicas of old and fresh data are stored and processed</li> <li>• 1 client node (Azure DS14 v2) running Debian GNU/Linux with 16 processing cores and 112GB of RAM, which coordinates and aggregates the search results.</li> </ul>
Online—web processing node	Azure D12 v2 running Debian GNU/Linux with four processing cores and 28GB of RAM. Service running: Nginx server with Django-based web app. This node serves as the LKN web platform.
Online—database node	Azure DS4 v2 virtualised machine) running Debian GNU/Linux with 8 cores and 28 GB memory running Postgresql, Cassandra and MongoDB. This node stores diverse user information, mailing management, and storing input form data from users.

million expert profiles, over 2 million entity profiles, and more than 200 million innovation topics.

LKN manages vast amounts of information through different offline and online layers. The offline layer, Data Acquisition, comprises several pre-processing components working in parallel over raw data to homogenise structure and identify entities and semantic relations. The online layer, Processing and Indexing, is done over a virtual cluster of search nodes based on ElasticSearch (ES). Finally, the Web and Search layer is where user queries execute several internal queries over LKN indices, retrieving the data to be displayed in the User Interface (UI). User queries are received by the virtual Nginx web server that also renders the results pages. The LKN platform is deployed on a heterogeneous technology stack on the Microsoft Azure cloud with three types of nodes: web processing nodes, database nodes, and the aforementioned index and search nodes. An overview of the LKN platform components is provided in Table 6.7.

### 6.5.2 *Issues and Challenges*

Small businesses typically operate in constrained business environments with a tension between scaling for growth and cashflow. While cloud computing provides significant benefits in terms of cashflow management and scalability, controlling consumption and managing complex cloud

infrastructure with a small IT team are significant challenges. Small businesses may not be able to accommodate reactive approaches to infrastructure provisioning (given the elevated warm-up times) and could save costs and improve QoS by using predictive solutions. Such solutions should allow effective and efficient provisioning/deprovisioning of cloud capacity by predicting spikes in demand in the short- and medium term and enabling boot-up instances in advance thereby addressing consume pattern prediction by geographic region and accurately anticipating periodic time-based traffic patterns.

In this case study, LKN overprovision nodes in Azure to cope with unexpected or irregular request peaks by users with a focus on serving the Eastern US market. LKN would like to optimise their cloud resources to reduce the cost of overprovisioning and avoid platform replication in non-core geographic markets. RECAP Data Analytics and Simulation and Planning methodologies and tools were used to support LKN in the capacity planning.

### 6.5.3 *Implementation*

#### 6.5.3.1 *RECAP Data Analytics*

##### **Step 1: Exploratory Data Analysis**

###### Web Server Error Analysis

Firstly, LKN data were evaluated from a quality perspective. Given that the workload is based on the number of user queries, the errors were evaluated (number of invalid requests) reported by the web server as per Table 6.8 below.

Although 78.78% of the queries were successfully answered by the search engine, the number of errors is very high for this kind of service.

###### Request Source Analysis

In a second step, the source of all requests was analysed and ordered by the number of requests. Table 6.9 below presents data based on the first 10 entries.<sup>2</sup> 45% of the requests are originated from a few IP addresses.

<sup>2</sup>IP addresses and other confidential data were anonymized by LKN before providing the data to RECAP.

**Table 6.8** Statistics of the response codes returned by the LKN search engine

<i>HTTP status</i>	<i>Description</i>	<i>Number of entries</i>
200	OK	1,692,070
302	Moved temporarily	175,577
301	Moved permanently	111,109
503	Service unavailable	82,905
404	Not found	45,439
304	Not modified	27,782
206	Partial content	8024
499	Client closed request	2497
403	Forbidden	1190
400	Bad request	699
500	Internal server error	117
405	Method not allowed	100
502	Bad gateway	72
504	Gateway timeout	19

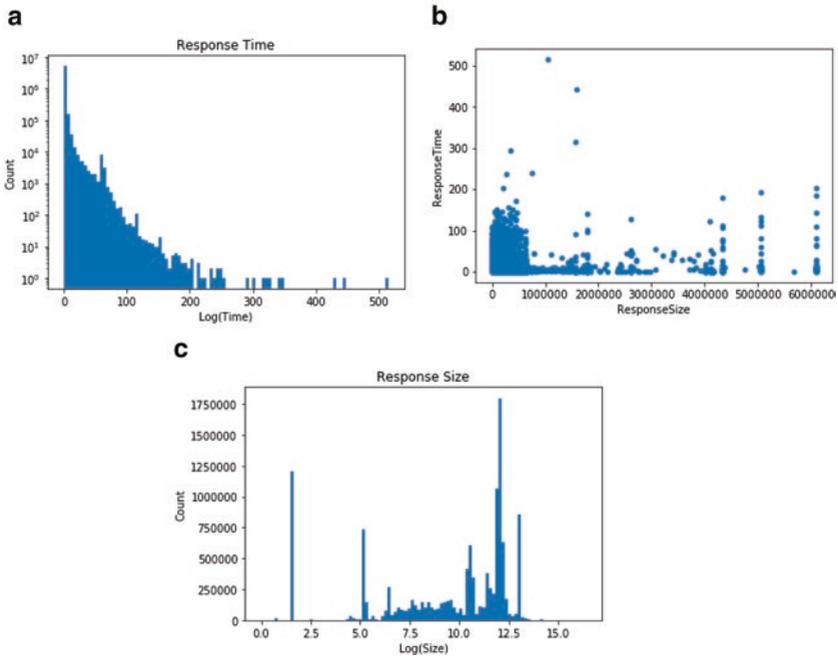
**Table 6.9** The top 10 IP addresses directing the largest number of requests to the LKN search engine

<i>Anonymous IP</i>	<i>Count</i>
IP_1	184,381
IP_2	124,093
IP_3a	90,711
IP_3b	85,367
IP_3c	78,317
IP_3d	70,576
IP_4	69,950
IP_5	55,806
IP_6a	44,459
IP_6b	29,771

These IPs correspond to web-spider bots from large search companies, e.g. Google and Yandex. As this provides visibility for LKN in search engine results, no remediation action was taken.

#### Response Size and Response Time

Figure 6.17 shows a histogram of the response volume of the LKN search engine. Figure 6.17(b) presents a histogram of the (total) time that the search engine requires to provide an answer to user queries. The

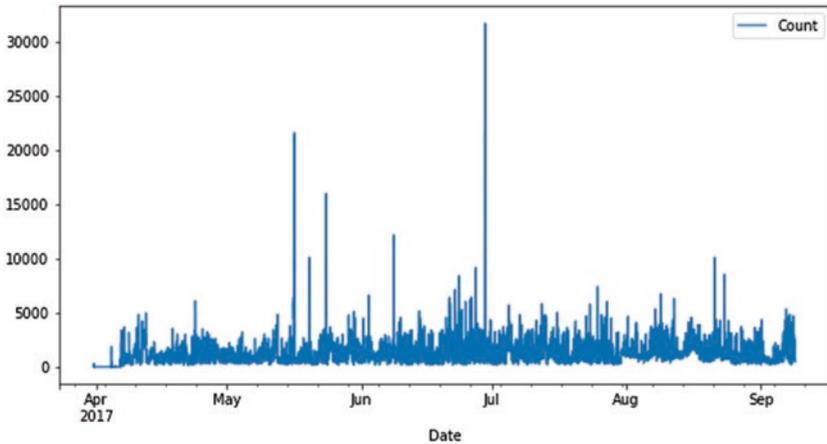


**Fig. 6.17** (a), (b), and (c): Histograms of the distribution of the responses, response time, and scatter plot of the response size and time for the LKN search engine

correlation between the response size and the response time was also studied and is shown in Fig. 6.17(c). The Pearson correlation coefficient is 0.18. If only 200 (OK) requests are considered, it is 0.20. Both 0.18 and 0.20 suggest a positive (although weak) correlation between both values.

### Number of Requests

The number of requests has also been characterised through a time series with the number of requests that the server receives aggregated over intervals of 30 minutes. This time series is the target workload that is analysed and modelled as part of the RECAP methodology. Peaks in this period can be explained by increased media attention during the period (Fig. 6.18).



**Fig. 6.18** Time series of the LKN's search engine workload (data aggregated over windows of 30 minutes)

### 6.5.3.2 Workload Predictor Model

From experience, the time to deploy a new data node in Azure clusters is about 30 minutes. The original dataset was aggregated in periods of 30 minutes, and a new feature, number of requests, was derived. Predicting the number of servers required for the next period of 30 minutes (workload prediction) to deal with the expected user requests is the goal of the model.

As a preliminary step before fitting a model to predict future workloads, the stationarity of the time series was examined. A visual inspection of a moving average and a moving standard deviation, together with a decomposition of the series in trends + cycles + noise, suggests a stationary time series. This intuition has been confirmed with a Dickey-Fuller test. The coefficients of the autoregressive integrated moving average (ARIMA) were estimated using an autocorrelation function (autoregressive part) and a partial autocorrelation function (for the moving average) where the already identified components were removed. The final ARIMA model trained corresponds to an autoregressive part of four periods, and a moving average of six periods, without requiring the integration of the original time series.

Figure 6.19 presents a sample dashboard with the Workload Predictor and the Application Modeller for LKN. The dashboard is composed of

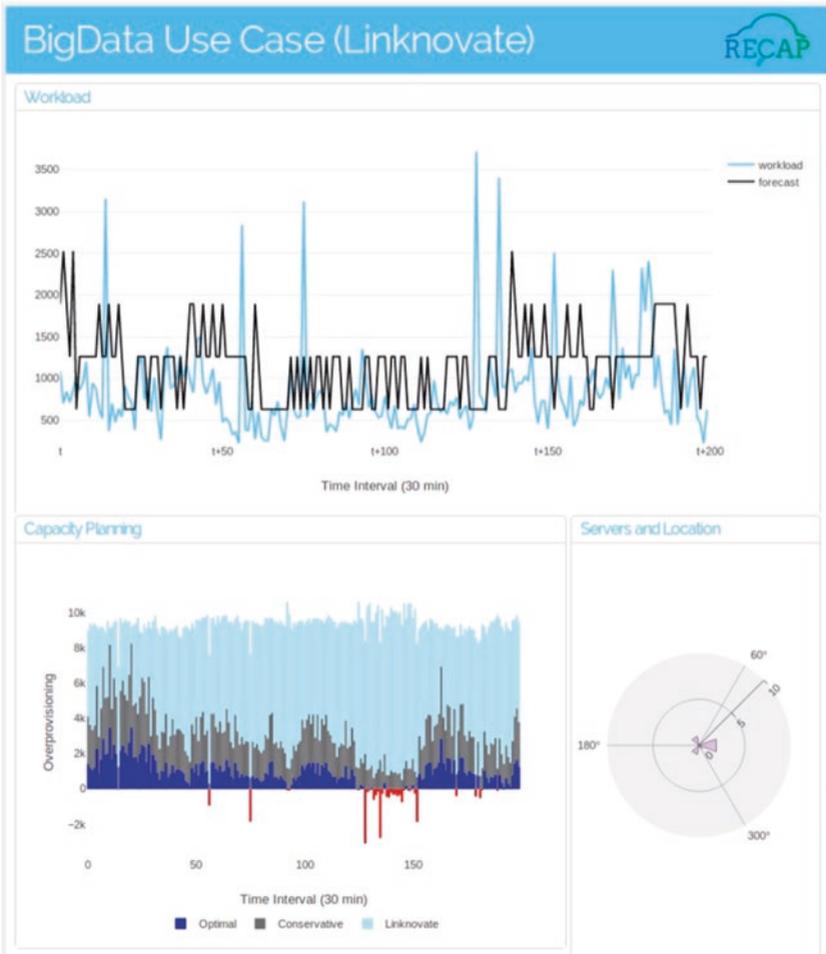


Fig. 6.19 Workload predictor dashboard

three independent but synchronised panes. The “Workload” pane, displayed in light blue, presents the actual workload (number of requests during each 30-minute interval) of the search engine, along with the forecast number of servers required to deal with the predicted workload (in black). The “Capacity Planning” pane presents the overprovisioning of resources deployed to deal with the actual workloads. The current overprovisioning of

the servers in production used by LKN is displayed in light blue, while the overprovisioning given the RECAP models is displayed in dark blue together with the underprovisioning in red. The grey display depicts a conservative model that does not underprovision resources. Finally, the “Servers and Location” pane presents the recommended number of servers and their geographical locations, as they are predicted by the application model.

RECAP ran a simulation of the LKN workload based on a historical dataset collected at the production server for a period of one month (August 2017). Based on the collected data, LKN was overprovisioning during that period of time by an average number of 13.8 cores of Azure DS12 v2 i.e. 86.6% of overprovisioning of data processing capacity. Applying the RECAP models would reduce overprovisioning to 3.5 cores, or the equivalent of 60% overprovisioning.

#### 6.5.3.3 RECAP Simulation and Planning Mode

The RECAP Simulation Framework supports a number of features that can help in ES-based system deployment and provisioning decisions. These include:

- Modelling and simulation of a distributed data flow with a hierarchical architecture;
- Custom policy implementation for distributing workload in the hierarchical architecture;
- Synchronous communication between search engine components for data aggregation; and
- Flexible modelling that can be easily adapted to integrate with other CloudSim extensions.

#### Modelling

Figure 6.20 shows the online virtual layers of the LKN search engine considered in the implementation of the RECAP Simulation Framework. As discussed earlier, the deployed LKN search service stack consists of a web server where the users input their queries and an ES cluster which is responsible for the search and returning the response to the user query. The ES cluster consists of an ES client node and data nodes. The ES node is responsible for: (1) passing and distributing the queries among the data nodes; (2) coordinating and aggregating the search results of different data nodes; (3) and returning the query result to the web server, which in turn returns it to the user. The data nodes are responsible for storing and processing old and fresh data.

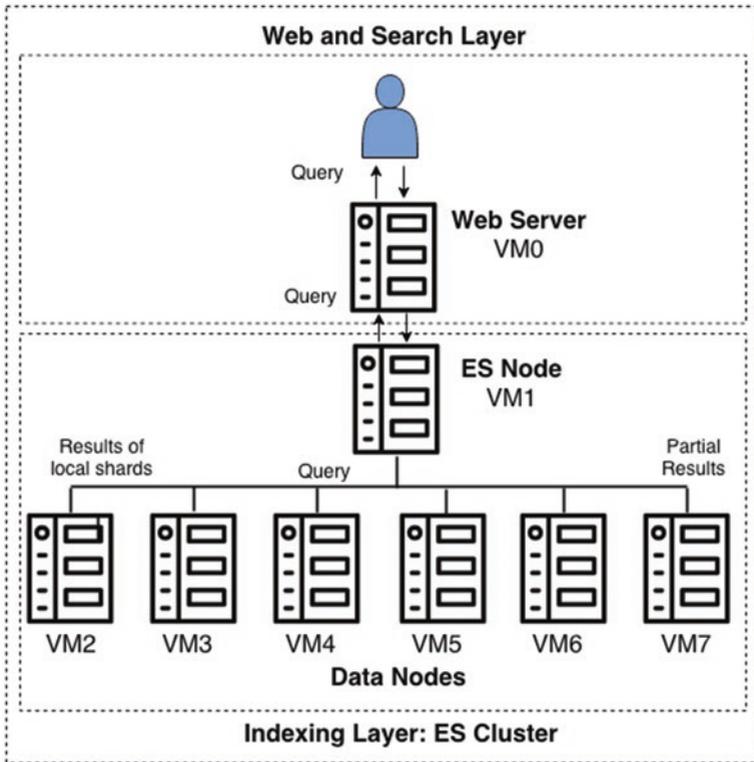


Fig. 6.20 LKN conceptual ElasticSearch (ES) architecture

A simulation model was built to reflect the behaviour of a real ES-based system deployed in a public cloud based on the LKN workload data as a reference. An ES-based search engine was then modelled and simulated using a Discrete Event Simulation (DES) approach. To do so, CloudSim, a widely used open source DES platform, was extended with the simulation model and then compared with KPI traces collected from LKN.

Figure 6.21 illustrates the ES workload flow. Within CloudSim modelling concepts, a cloudlet represents a task submitted to a cloud environment for processing. When a query is launched, a set of cloudlets is generated and executed in sequential manner. The first cloudlet is executed at a web server then the second cloudlet is executed at the ES node. From the ES node, a set of cloudlets (which is less or equal to the number

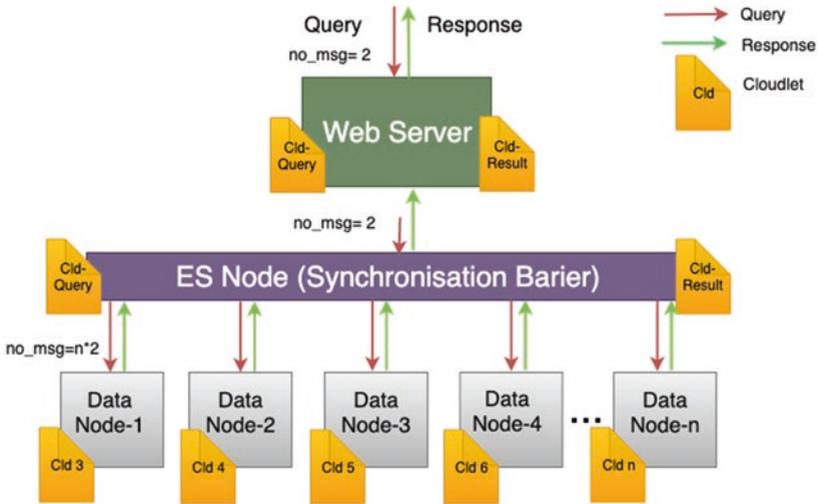


Fig. 6.21 ElasticSearch (ES) workload flow

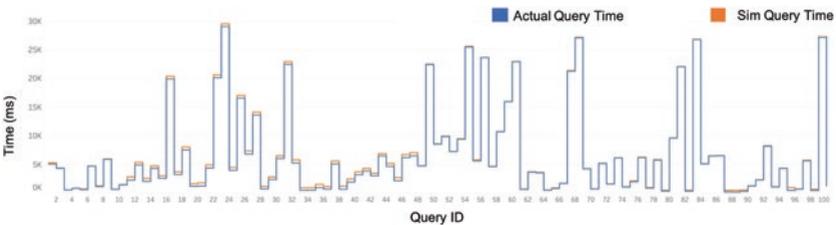


Fig. 6.22 Comparison of actual and simulation query response times

of data nodes) is distributed and executed at data nodes. Next, another cloudlet is executed again at the ES node to merge the partial results coming back from the data nodes. Finally, a last cloudlet goes from the ES node to the web server as a response to the user query.

### Results

The simulated response time of a query was compared to its actual time as collected from real system traces. A subset of 100 valid queries was extracted from the data set used. Figure 6.22 compares actual and simulation query response times across the 100 queries. As one can see, the

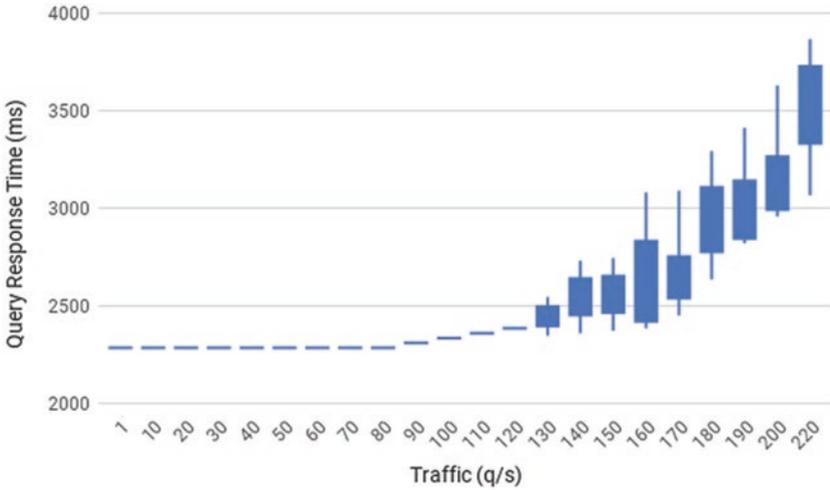


Fig. 6.23 LKN system performance under different traffic scenarios

actual query response times and the simulation query response times are very close and highly positively correlated across all the 100 queries tested.

The performance of the LKN system was analysed by running the simulation with different workloads (query traffic) to see how much traffic the LKN system could handle. Query response time was monitored while varying the number of queries per second (q/s) received by the system. Figure 6.23 is a box plot (min, max, lower quartile, upper quartile) that shows the query response time based on the number of queries per second the system receives.

With query traffic of up to 80 q/s, the query response time for all the queries is the same and it is equal to having one q/s. That means the system is capable of handling 80 q/s with no waiting time. Between 80 q/s and 120 q/s, a slight increase in the response time appears. However, this increase affects all the queries in the same way, i.e. there is no difference in response time between the queries. As we increase the query traffic beyond 120 q/s, a divergence in query response times becomes apparent. Between 130 q/s and 170 q/s, the system manages to execute several queries within a short time by delaying the excess of queries; however, with the increase in query traffic beyond 170 q/s, the system fails to execute even a single query in a short time.

### 6.5.4 Results

The analysis run for this case study has proven valuable in multiple ways. First, LKN was able to gain a better understanding of infrastructure planning, deployment, anomaly detection. Similar, workload prediction models estimate the potential cost savings due to improved resource consumption of 26.6%.

Using the RECAP DES simulator, it was possible to provide an insight for LKN into capacity planning in that it identifies thresholds at which point LKN's QoS starts degrading and additional resources must be provisioned. LKN can now use these data to reduce overprovisioning and at the same time set specific rules for scaling up and down in a cost-effective manner.

All these insights resulted in LKN changing their deployment strategy moving to a more advanced infrastructure configuration using only two data nodes (instead of nine) with NVMe (non-volatile memory express) storage.

## REFERENCES

- Ai, Y., M. Peng, and K. Zhang. 2018. Edge Computing Technologies for Internet of Things: A Primer. *Digital Communications and Networks* 4 (2): 77–86.
- Ofcom. 2016. Business Connectivity Market Review—Volume I—Review of Competition in the Provision of Leased Lines. [https://www.ofcom.org.uk/\\_data/assets/pdf\\_file/0015/72303/bcmr-final-statement-volume-one.pdf](https://www.ofcom.org.uk/_data/assets/pdf_file/0015/72303/bcmr-final-statement-volume-one.pdf).
- Sauter, T., J. Jasperneite, and L. Lo Bello. 2009. *Towards New Hybrid Networks for Industrial Automation*. 2009 IEEE Conference on Emerging Technologies & Factory Automation, 1–8. IEEE.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

