Report from Dagstuhl Seminar 21243

Compute-First Networking

Edited by

Jon Crowcroft[1], Philip Eardley[2], Dirk Kutscher[3], and Eve M. Schooler[4]

1 **University of Cambridge, GB,** jon.crowcroft@cl.cam.ac.uk
2 **BT Applied Research – Ipswich, GB,** philip.eardley@bt.com
3 **FH Emden, DE,** dku@dkutscher.net
4 **Intel – Santa Clara, US,** eve.m.schooler@intel.com

Abstract

A Dagstuhl seminar on Compute-First Networking (CFN) was held online from June 14th to June 16th 2021. We discussed the opportunities and research challenges for a new approach to in-network computing, which aims to overcome limitations of traditional edge/in-network computing systems.

The seminar discussed relevant use cases such as privacy-preserving edge video processing, connected and automated driving, and distributed health applications leveraging federated machine learning. A discussion of research challenges included an assessment of recent and expected future developments in networking and computing platforms and the consequences for in-network computing as well as an analysis of hard problems in current edge computing architectures.

We exchanged ideas on a variety of research topics and about the results of corresponding activities in the larger fields of distributed computing and network data plane programmability. We also discussed a set of suggested PhD topics and promising future research directions in the CFN space such as split learning that is supported by in-network computing.

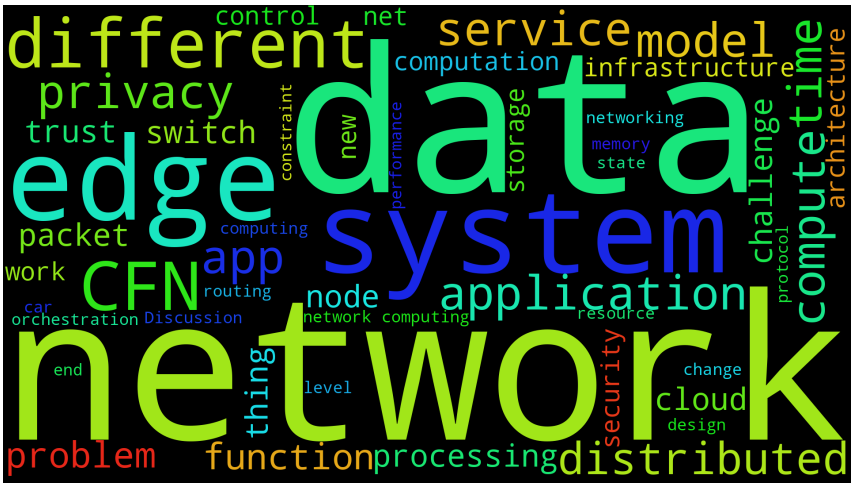
Seminar June 13–16, 2021 – http://www.dagstuhl.de/21243
2012 ACM Subject Classification Networks → Network architectures; Networks → Network design principles
Keywords and phrases Distributed Machine Learning, distributed systems, edge-computing, in-network computing, networking
Digital Object Identifier 10.4230/DagRep.11.5.54

Compute-First Networking, *Dagstuhl Reports*, Vol. 11, Issue 05, pp. 54–75
Editors: Jon Crowcroft, Philip Eardley, Dirk Kutscher, and Eve M. Schooler

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Dirk Kutscher



Edge- and more generally In-Network Computing are key elements in many traditional content distribution services today, typically connecting cloud-based computing to consumers. The advent of new programmable hardware platforms, research and wide deployment of distributed computing technologies for data processing, as well as new exciting use cases such as distributed Machine Learning and Metaverse-style ubiquitous computing are now inspiring research of more fine-granular and more principled approaches to distributed computing in the "Edge-To-Cloud Continuum".

The Compute-First Networking Dagstuhl seminar has brought together researchers and practitioners in the fields of distributed computing, network programmability, Internet of Things, and data analytics to explore the potential, possible technological components, as well as open research questions in an exciting new field that will likely induce a paradigm shift for networking and its relationship with computing.

Traditional overlay-based in-network computing is typically limited to quite specific purposes, for example CDN-style edge computing. At the same time, network programmability approaches such as Software-Defined Networking and corresponding languages such as P4 are often perceived as too limited for application-level programming. Compute-First Networking (CFN) views networking and computing holistically and aims at leveraging network programmability, server- and serverless in-network computing and modern distributed computing abstraction to develop a new system's approach for an environment where computing is not merely and add-on to existing networks, but where networking is re-imagined with a broader and ubiquitous notion of programmability.

We expect this approach to enable several benefits: it can help to unlock distributed computing from the existing silos of individual cloud and CDN platforms – a necessary condition to enable Keiichi Matsuda's vision of Hyper-Reality and Metaverse concepts where the physical world, human users and different forms of analytics, and visual rendering services constantly engage in information exchanges, directly at the edges of the network. It can also help to provide reliable, scalable, privacy-preserving and universally available platforms for Distributed Machine Learning applications that will play a key role in future large-scale data collection and analytics.

CFN's integrated approach allows for several optimizations, for example a more informed and more adaptive resource optimization that can take into account dynamically changing network conditions, availability of utilization of compute platforms as well as application requirements and adaptation boundaries, thus enabling more responsive and better-performing applications.

Several interesting research challenges have been identified that should be addressed in order to realize the CFN vision: How should the different levels of programmability in todays system be integrated into a consistent approach? How would programming and communication abstractions look like? How do orchestration systems need to evolve in order to be usable in these potentially large scale scenarios? How can be guarantee security and privacy properties of a distributed computing slice without having to rely on just location attributes? How would the special requirements and properties of relevant applications such as Distributed Machine best be mapped to CFN – or should distributed data processing for federated or split Machine Learning play a more prominent role in designing CFN abstractions?

This seminar was an important first step in identifying the potential and a first set of interesting new research challenges for re-imaging distributed computing through CFN – an exciting new topic for networking and distributed computing research.

2 Table of Contents

3 Overview of Talks

3.1 In-network telemetry

Gianni Antichi (Queen Mary University of London, GB)

Abstract

The possibility to easily add new functionality to network data planes has lately opened new exciting research directions towards understanding how such a programmability can impact the design of networks as well as their services. This talk focussed on network monitoring and analyzed how both flow-based and packet-level telemetry can be efficiently implemented on state-of-the-art programmable switches. Starting from the limitations of current sketch-based solutions and in-band network telemetry (INT), the presentation discussed new ways to provide faster flow level analysis to network collectors and lighter in-band telemetry through the use of network coding techniques. The talk concluded with considerations on what are the next steps in the field.

Discussion Summary

In-network telemetry was discussed as an example for a dataplane mechanism that could enable several useful measurement and management services in CFN networks such as exposing dynamic load/utilization information for joint optimization and execution profiling. Depending on the use case there may be different requirements with respect to the fidelity and completeness of the observed information. The current in network telemetry concepts are interesting due to their "end-to-end" nature, not requiring state on network elements. Several references to related concepts were made including [1], [2], [3], [4], [5].

References

1 CLOUD COMPUTING TECHNOLOGY LAW AND POLICY, Microsoft. `http://www.mccrc.org/`
2 Identifying and Supporting Financially Vulnerable Consumers in a Privacy-Preserving Manner: A Use Case Using Decentralised Identifiers and Verifiable Credentials, Tasos Spiliotopoulos et al, 2021. `https://arxiv.org/abs/2106.06053`
3 Securing software by enforcing data-flow integrity, Miguel Castro et al, OSDI 2006. `https://www.microsoft.com/en-us/research/publication/securing-software-by-enforcing-data-flow-integrity/`
4 NSA, Andreas Eschbach, 2020. `https://www.luebbe.de/luebbe-belletristik/buecher/sonstige-belletristik/nsa-nationales-sicherheits-amt/id_7539196`
5 Split Learning: Distributed deep learning without sharing raw data, MIT project page. `https://www.media.mit.edu/projects/distributed-learning-and-collaborative-\learning-1/overview/`

3.2 CFN for Health

Jon Crowcroft (University of Cambridge, GB)

Abstract

Sensing well-being and health via mobile devices is becoming mainstream. Logging the data centrally doesn't scale, and causes privacy (and ethical and legal) problems. Local processing, and in-network aggregation can support the individual (life logging, diagnostics etc) and public health (e.g. epidemic, e.g. avoiding infection, and environmental risk, e.g. air quality). As a use case for CFN, and an instance of federated machine learning and a challenge for privacy designs, the system architecture for such apps make an informative study.

Discussion Summary

We discussed how covid apps for contact tracing use a compute-first networking approach. In general, aggregated information is good enough for many applications. Linking back to the previous talk, we discussed the need to be able to explain the privacy and security properties of a system, and how it is easy to imagine dystopian futures.

3.3 New Network for Distributed Systems

Jianfei He (City University – Hong Kong, HK)

Abstract

An infrastructure like Networks should be designed with stable functions which are decoupled from the semantics of specific applications. This talk proposes to address the common requirements of various distributed systems, including collective communication and consistency. It is also emphasised that new network diagrams should be justified not only by its improvement on end-to-end performance, but also by its superiority over potential solutions at upper layers.

Discussion Summary

This talk led to a discussion of generality and modularity, i.e., which are the right application-agnostic building blocks? Would it make sense to have a taxonomy of different distributed application types and their requirements for in-network and distributed computing support functions? Whereas some fundamentals of distributed computing are well-understood and will be relevant in the future, it is not always useful to predict how certain mechanisms will be used and what optimization techniques might emerge, which seems to suggest a rather flexible approach to layering and re-use. In this context, Scaling Replicated State Machines with Compartmentalization [1] and CoFlow (`https://amplab.cs.berkeley.edu/tag/coflow/`) were mentioned.

References

1 Secure Scuttlebutt, project pages. `https://scuttlebutt.nz/`

3.4 Networking matters for storage systems

Micchio Honda (University of Edinburgh, GB)

Abstract

Although networks have been programmable and able to perform computation, the trend of end-to-end encryption and stream-oriented TCP protocol make it difficult to benefit from such programmable networks. This talk presents Prism as an opportunity of significantly improving cluster resource utilization and reliability with the aid of programmable networks although the communication occurs over regular TCP and TLS. We instantiate Prism as an scale-out object storage system that communicates with the client using widely used S3 protocol.

Discussion Summary

We discussed comparison to other approaches which use custom UDP-based network protocols in terms of achieving compute-first networking and usability. We also discussed other system-level approaches such as VM live migration.

3.5 Privacy Preservation in Edge Video Processing

Jag Minhas (Sensing Feeling – London, GB)

Abstract

In today's world of virtual computers, high speed networking and algorithms powered by deep neural networks, application developers have a wide choice of architectures and arrangements at their disposal to implement working solutions. For those implementing vision processing systems in the area of human surveillance and behaviour these architectural choices will often have an impact on how end-user rights to privacy are upheld. This talk summarises an established "privacy affirming" framework for distributed vision processing systems and discusses the implications of this approach as our architectural choices widen further in a "Compute First Networking" future.

Discussion Summary

Privacy is agreed as a critical topic for compute-first networking. Distribution of processing offers the potential for improved privacy protection, for example by distributing machine learning inference to multiple nodes, but there are many subtle issues (for example, the assumption that you can trust that the system is doing what it's supposed to). There can be privacy implications even with no (other) human in the loop, for example the use of captured video for ad tracking. A full solution needs to consider the legal and policy framework as well as the technical design.

3.6 Compute Centric Networking for Connected and Automated Driving

Naresh Nayak (Robert Bosch GmbH – Stuttgart, DE)

Abstract

In-vehicular networks are rapidly evolving to support the future requirements of the automotive industry stemming, e.g., from highly autonomous driving. The introduction of microprocessor platforms (aka vehicle computers) networked over high bandwidth Ethernet backbone coupled with wireless connectivity has transformed the modern day vehicle into an IoT device on wheels. The continually increasing amount of computing capability within a vehicle, however, comes at a cost (monetary, energy, etc.). In-network computing is a promising avenue to include the computing capabilities from the infrastructure, and thus, reduce the corresponding costs. This talk presented different use cases where driving functions could be offloaded from a vehicle to the infrastructure and discussed the associated research challenges.

Discussion Summary

A vehicle now contains many IoT devices and produces a lot of data (raw data at 10s TB/hour). So there can be a compute-first network within the car, as well as in the nearby roadside infrastructure. Some of the use cases have critical timing requirements (for safety reasons) – so may be a use case for deterministic computing – whilst others are more tolerant of latency. Security and trust are critical issues, especially if cars use information from other cars as part of automated driving.

3.7 EVEn Harder Challenges

Erik Nordmark (Zededa – San Jose, US)

3.7.0.1 Abstract

The LF Edge EVE project targets IoT workloads at the smart device edge whilst LF Edge has a suite of other projects for other parts of the edge-cloud continuum. EVE's approach to control issues is that there is a single controller per edge node with a "phone home" function. EVE's approach and assumptions to issues such as trust, placement and security were outlined.

3.7.0.2 Discussion Summary

Discussion was about detailed issues concerning the LF Edge EVE project.

3.8 Networking and Computing – the great confluence

David Oran (Network Systems Research & Design – Cambridge, US)

Abstract

Two contemporaneous trends are merging the two separate yet interdependent technologies of computing and networking. Hardware and software historically associated with computing complexes (General-purpose CPUs running virtual machines, conventional operating systems, and languages) are being used more and more to host networking functions at all but the highest speed tiers. Networking devices such as switches, routers and NICs are becoming programmable in ways that allow general purpose computing to be done "in the network". This talk examines these trends, presents some of the salient research illuminating the advantages and limitations, and speculates on where this merging of technologies might take us.

Discussion Summary

Comparing the capabilities of servers and switches, for example in terms of cycles/bit, memory and rate of feature evolution: server capabilities are superior (providing only a limited number of ports are needed) and programming is easier.

Many topics were discussed.

One issue associated with joint optimisations (e.g, considering networking, compute, storage and possibly energy resources) is if the coordinated allocation of resources is allowed across boundaries. We should learn from the experiences with multi-provider QoS which has been a non-starter for decades (for commercial and policy reasons – we do have the necessary protocols).

3.9 Split Learning Opportunities with CFN

David Oran (Network Systems Research & Design – Cambridge, US)

Basics of Split Learning Summary of potential advantages for combining split learning with edge computing 3 Possible use cases, focused on IoT (mostly)

3.10 (Some Random Thoughts on) In-Network Compute Architecture

Jörg Ott (TU München, DE)

Abstract

In-network computing, the execution of program logic on behalf of an application "inside" the network, may be interpreted in many ways, yielding rich history: from the early work on active networking to flavors of mobile offloading in the context of pervasive computing to

programmable data planes to edge computing to recent extensions to information-centric networking in support of computations, among others. These interpretations differ in the complexity of the application structure exposed to "the network", the code complexity, where and on what kind of devices program logic is executed, and how and when the location(s) for execution are determined and routed to. In this talk, we explore some architectural considerations on in-network computing. We start out from basic networking/computing primitives and explore basic units of computation and interaction, look into different application models, where the "main" control thread (if any) could reside, and how traditional Internet design principles, especially best effort operation and the end-to-end principle, could translate to in-network computing. Another important question is how much abstraction from (as opposed to insight into) the execution scheduling and orchestration a network should provide. We discuss the importance of naming and scoping concepts for in-network functions (and state) that make up the program logic: for invoking them from applications outside the program logic as well as for the possible interactions among multiple (distributing) functions comprising this very logic. We look at name resolution and routing mechanisms and how transport layer mechanisms relate to in-network computing. Overall, there seem to be many feasible approaches, and many design decisions likely depend on how such in-network compute infrastructure would eventually be realized: augmenting IP, running as overlays, or defining a novel clean-slate network architecture.

Discussion Summary

The presentation raised many questions that prompted much discussion and further questions, including: whether one needs to think about specific use case examples and the business models; what the API should be like; what do we name (nodes, data, functions, processes etc) – do we explicitly invoke a function, or do we address an object and the function happens consequently; "ownership" of a function's execution and the associated resource; partial updating is normal; how to take account of the unreliability of memory and so also how to degrade gracefully; and whether all computation should be explicit (so it is all "end to end", with no implicit or "hidden" computation).

3.11 Deconstructing and Reimagining Orchestration

Andy Reid (BT – Ipswich, GB)

Abstract

Orchestrators are a general concept, including SDN controllers, cloud controllers and package managers, which are layered (one form of orchestrator feeds another). Each has an interface through which its user can inject policy. Each includes the same set of activities (such as parser, translator, configuration manager and optimiser). Intent-based networks have the same general approach. Today's orchestrators are focussed on the management plane. The proposal is that it would be beneficial to re-style orchestrators as primarily part of the control plane, which would imply a greater degree of automated operation, with security built into the protocol. One implication suggested is that functions should be called by name, not by address.

Discussion Summary

One topic of discussion was about optimisation: whether this is easier with hard or soft constraints, given the overall system is likely to be non-linear and a "soft" answer is wanted. Since compute resource has no real equivalent of the "bit" as a unit of networking resource, it is harder to optimise. Another topic was how to evolve from the current state to the desired state (as recommended by the optimiser), without putting the network /system through an unsafe intermediate state (to avoid this problem, routing policy changes have to go through many intermediate steps). A further evolution issue is that feedback is needed about whether the resources required in the desired state are actually available (especially as there may be inaccuracies), whereas today's control planes typically just "try now" – the potential role of digital twins was mentioned.

3.12 Heterogenous Networks and Laying out the Compute Graph

Peer Stritzinger (Peer Stritzinger GmbH – Maisach, DE)

Abstract

For use cases in edge computing and IoT placing compute functionality in a heterogeneous not fully connected network is an open challenge. Focussing on a computational model consisting of lightweight processes and message passing approaches to modelling distributed algorithms and compute and network resources have been explored. Process and messages are a model to which all distributed computational models can be mapped easily. Relatively static vs. highly dynamic mapping is necessary for different use cases. Properties of processes vs. nodes and messages vs. links need to be combined with topology of communication and network. Orchestration needs to be implementable in a distributed manner without central control.

Discussion Summary

Discussion included Erlang approaches to achieving this distributed computation model. Other issues raised included: the potential role of time sensitive networking and deterministic networking; energy and green energy sensitivity; 'interference' from other traffic or computations; and how to scale to distributed IoT systems that could have billions of processes.

3.13 After the Fun the Hard Stuff

Christian Tschudin (Universität Basel, CH)

Abstract

A core assumption of traditional distributed computing is that processes send messages among themselves to coordinate their work while in this talk we suggest exploring memory-mediated coordination. The Internet is process-centric (packet switching among protocol entities)

which is typically made available to applications in the form of interprocess communication (IPC). Decentralized systems like Secure Scuttlebutt, however, have no notion of IPC and instead rely on any form of content propagation (id-centric append-only logs). Such a content-centric venue may become important when we envisage highly volatile in-network computing environments where processes are not stable enough to receive and answer messages, or because they follow probabilistic computing models, or they simply are not concurrently online. We exclude approaches based on a request-reply pattern à la CCN/NDN and Named-Function Networking (which was fun, but too perfect) because they imply some notion of processes. While one could try to harden the network fabric to provide message buffering beyond process lifetime (mailbox systems for actors), we suggest to focus on bare in-network memory and force processes to coordinate via "DIstributed Memory Side Effects" (DIMSE) only. Such an architecture would have immediate benefits beyond crash-resilience, namely for delay tolerance, transport agnosticity and even for data-center networks (fast remote memory access for HPC).

Discussion Summary

The talk prompted a very lively discussion. The idea is for push-only data distribution, with social forwarding-and-filter (meaning that filtering is based on the social graph). The role of "scent marks" to enable this was discussed, with each node in memory storing a collection of scent marks. Questions were raised about garbage collection, contracts and motivations (and tit-for-tat issues), and lessons from earlier systems such as Linda, SRM and floating content.

3.14 In-network computing challenges and opportunities

Noa Zilberman (University of Oxford, GB)

Abstract

Programmable network devices have changed the way we perceive the network, allowing us to rethink where, when and how we process data. The emergence of in-network computing, the execution of standard host applications within network devices, has been especially promising. However, the rise of in network computing has brought with it multiple challenges, ranging all the way from architecture challenges, through micro-architecture, design and validation, to multi-tenancy. This talk explored the many challenges in designing functional, high performance, and robust in-network applications. These challenges propose new research opportunities, allowing to turn in-network applications from research-ideas into commercially-viable solutions.

Discussion Summary

The discussion focused on the applicability of P4 to distributed computing. Whereas P4 dataplane programming is sometimes considered as the most promising in-network computing approach, this talk highlighted many of the practical difficulties. Whereas some point solutions for distributed computing applications and optimization have been developed, such as [1], the lack of Turing completeness and the vast heterogeneity in platform capabilities seems to suggest that P4 dataplane programming would be suitable

for certain packet-level optimization while possibly integrated into a larger, heterogeneous distributed computing framework. In that sense, P4 dataplane programming could represent a certain class of enabling platforms, and a compelling research challenge might be to productively leverage such special environments. Similar concerns could arise for other powerful, but really specialized environments such as Multi-GPU communication (e.g., `https://www.nvidia.com/en-gb/data-center/nvlink/`).

References

1 Raft does not Guarantee Liveness in the face of Network Faults, Heidi Howard et al, 2020. `https://decentralizedthoughts.github.io/2020-12-12-raft-liveness-full-omission/`

4 Open problems

4.1 Potential Research Problems

All Participants

The seminar held one session on discussing research opportunities and potential PhD topics in the CFN (Compute First Networking) field to develop an understanding of what original and substantial research activities could look like. Here is a summary of the ideas discussed:

- Dynamic compilation and placement of CFN instances – network-wide, cross-platform, just-in-time compilation
- Profile-Guided Optimization as CFN Optimizer – Profile-guided optimization is an existing technique that uses profiling test runs of a program to identify parts of the code that are executed more frequently and should be optimised. The idea would be to use data plane telemetry to do the profiling.
- Layer 3 Protocol for In-Network Computing – a clean slate protocol is suggested for routing to compute, possibly building on some of the ideas from ICN and NFN
- In-Network Computing for Network Management – with a logically centralised control plane but policy distributed to each node
- Analysis of CFN – the research topic would be to demonstrate the "provable" gain for CFN
- In-Network Edge Computing for Smart Transportation – the goal would be fast, safe and improved coordination of all sorts of transport users (cars, lorries, trams, cycles, pedestrians, buses, and so on). Questions include the role of road side units and direct coordination between vehicles, and privacy-preserving techniques and data ownership.
- Climate change – what role can CFN play in the goal of net zero?
- Faster packet processing after hitting specialization limits – packet processing speeds are now improving only slowly, and specialisation for a given domain works once, so what are the options to improve network packet processing speed? The potential roles of reconfigurable hardware, specialist hardware, specialist channel, and quantum computing.
- Distribution Usage Descriptions to deny denial of service – the idea would be to infer about flows and to police them in a decentralised manner.
- Gracefully Degradable Turing Machines – faster hardware is hitting reliability limits, so algorithms are needed that cope with memory faults

- New Network for Distributed Systems – routers as we know them will decrease / disappear, instead we will have compute and inter-computing – in that world, what are the basic functions?
- Planarian Distributed Computing – the notion of a 'breaking point' in distributed computing – partitioning and merging. Is there an interesting space between consensus based and eventual consistency partition tolerance?
- Towards a Formalism for CFN – Start with Misra and Chandy's UNITY formalism, which has distributed computations as collections of iterated assignment statements, and extend into fault tolerance and different models for the comms/compute/storage
- Split Learning Opportunities with CFN – apply Deep Neural Networks, but split vertically across layers rather than horizontally across samples, and smashing the output data at the boundary to preserve only relevant excitation coefficients. This can help with privacy, by confining data to the device only, or in secure enclaves in edge computing nodes; it also can provide failure resiliency, as there's degraded operation (rather than none) when the cloud back-end can't be reached.

5 Findings and Recommendations for Compute First Networking

All Participants

In this section we present the organisers' conclusions from the seminar. Undoubtedly not all participants will agree with all of them.

The field of Compute First Networking (CFN) is at an exciting moment. We speculate that compute-first networking is at the same stage as cloud computing 20 years ago. At that moment, we had some insight about both the commercial drivers (the use cases for utility computing) and the core technical ideas (data centres, virtualisation and fairly fast networks). But we didn't understand how so many industries would be transformed, nor all the technical work required in order to be able to deploy and orchestrate services (such as map-reduce, stream processing, load balancers, migration, containers, and distributed ledgers).

We believe that Compute First Networking has similar potential to transform the business world and the way people live. Today, we have the first ideas and partial understandings about the use cases, commercial drivers and technical directions, but there is a huge amount still to do.

In the absence of new service models, it seems likely that the hyperscalers will extend their dominance of cloud computing into CFN. For those who would like to see a more diverse ecosystem, now is therefore an opportune moment to invest in researching the key components needed to disrupt the status quo. We recommend that government and industry fund extensive research and development in order to help bring CFN to fruition, and allow fresh innovation by a decentralised gallery of players.

5.1 Use Cases and Business Drivers for CFN

The continued consolidation in data centre clouds has made other deployment models appear as an exception – for example, we talk about edge computing when referring to non-cloud-only applications (where most of the artefacts and compute resources are concentrated in the cloud).

Non-cloud-only applications are becoming increasingly important, due to mainstream consumer-focused applications (scalability of content distribution and low-latency requirements for interactive applications) but specifically due to new applications that leverage the availability of Internet-connected physical resources such as sensors and video cameras, and the potential advantages of integrating these into larger distributed applications.

For example, from a mere data logistics perspective, (semi)-autonomous vehicles produce large amounts of data, mostly destined for local consumption (video and various sensors), but increasingly also for external processing and storage – for tele-driving, diagnostics, forensics and other applications. Such applications fundamentally require a robust data offloading and processing infrastructure – to some extent the reverse functionality of today's downstream CDNs (Content Distribution Networks). Processing should enable filtering, aggregation, and other forms of transformation, and must thus be fully programmable.

Not all interactions are upstream-data-based – for example, car and factory networks have rigid reliability and low-latency requirements. Corresponding control loops may be driven by local data as well as by remote information and instructions. Enabling such robust control in the presence of dynamic and heterogeneous sub-system properties calls for infrastructure support that enables operating distributed applications across different failure and time domains .

Infrastructure programmability and resource multiplexing raise the question of multi-tenancy, i.e., the ability to run several workloads over shared computing and network infrastructure with adequate isolation . Fundamental building blocks such as suitable encryption algorithms, trusted execution, attestation etc. exist and are emerging. They fuel the design of more reliable and trustworthy distributed applications over shared infrastructure, enabling new use cases.

When edge sensor data contains personally identifiable information (PII) or can be used to construct PII by combining multiple inputs, additional concerns regarding privacy preservation arise. We reject the suggestion that privacy is over-rated – there are already too many dystopian abuse scenarios – and we reject the 'counsel of despair' that it is already too late (because the hyperscalers and some governments, service providers, and/or credit card companies collect so much information today). On the other hand, the Covid-19 pandemic has shown importance of finding a 'middle way', where health data can be leveraged for tracking, trend analysis, and understanding transmission patterns. We believe that a future system should preserve privacy by design , meaning that it technically prevents leakage of PII (such as raw video data). This could enable a whole new class of applications, such as the use of medical data in a productive and ethical manner. In-network computing actually offers some promise for new approaches to privacy preservation, because computation is intrinsically distributed, and end users might be able to control which functions/analytics are run, where they run (in the local network), for whom, and on what data.

The economics of compute first networking merits research work. What types of compute function are likely to be of most value? Are there new use cases where compute-first networking offers particular benefit? What applications are distributed and multi-party? What are the externality costs like power, real estate, right-of-way, regulation of physical infrastructure?

Also worthwhile is consideration of the legal aspects of compute first networking. At a fundamental level, should there be regulation to achieve "in-network compute neutrality"? Today, some countries have regulations about "net neutrality" for the networking, but not about the cloud compute, which has (probably) been key to enable the development of a plethora of applications, but has disintermediated the telecoms operators from the value

chain. What is the preferred societal outcome for CFN? There are also detailed questions to be resolved, such as the appropriate legal frameworks when it may be necessary to reveal some of the data or its analysis. Examples include accident data for cars, medical malpractice, and criminal conspiracy.

5.2 Technical challenges for CFN

While distributed computing functionality in the network is an essential design feature in many mainstream applications today, current architectural models can only provide a limited form of in-network distributed computing due to a lack of appropriate programming models and run-time visibility.

Many applications today are becoming increasingly multi-party and distributed, e.g., user front-end applications talking to a microservice backend, possibly mediated by proxies, load balancers etc. However, from an application developer perspective, it is difficult and possibly undesirable to make the network fully transparent to the programmer: there are performance inhomogeneities in both throughput and delay, complex partial failures that require some form of response, depending on the distributed system's run-time behaviour. With current systems, except for rather static directed acyclic graph structures (such as Dataflow systems), traditional programming models today only easily exploit localized parallelism, so it is still not possible to reason about wider system parallelization without requiring a lot of orchestrator intelligence.

DevOps-based systems can theoretically enable incremental partial deployment and dynamic module updates, however coordination with network underlays is not trivial and not always possible due to strict separation into organizational and protocol realms.

In addition, computing and communications are on different cost/performance trajectories , which results in additional platform heterogeneity and different typical functions that one would run on compute servers vs. network switches. From a processing and memory perspective, compute servers can afford to spend many cycles per processed bit and to perform memory intensive operations, whereas network switches are concerned with line-rate processing on optimized hardware, allowing only a few cycles per bit with only small or moderate memory consumption. Correspondingly, the workloads are also quite different, especially considering the need for multi-tenancy and tenant isolation (requiring cryptographically enabled isolation on servers, which is typically not available on network switches). Thus, we believe that there will be "puddles and ponds" of compute, connected naturally via networks (rather than every device including networking and compute). For scalability reasons these resources are more likely to congregate.

Programmable data planes are likely to be an important component, but today's technologies, such as P4, are not fully formed as yet, as their capabilities are too limited and there are many tricky issues remaining to be solved for generic in-network computation. In addition to generic ('universal') processors, there will be a need for heterogeneous hardware specialised to jobs such as sensor processing and filtering (e.g. sparse FFT), or query minimization for differential privacy.

On privacy , a first (but not sufficient) technical step could be the careful selection of processing loci in the system, i.e., perform data analytics close to the data source and then only share processed, anonymized result data. Federated machine learning could be one such pattern that enables the controlled sharing of the results of analytics, for example featuring individual health data logging and public health data sharing via user-controlled analytics

functions. Other approaches, such as split learning could map the different deep learning layers onto different nodes/domains in the network, thus enabling a privacy-preserving distributed training process, where each layer can only look at its own raw data.

Provenance tracking is also important, as insights derived from data are dependent on data trustworthiness. Also, on a compute node, isolation is needed between tenants, with some kind of trusted computing.

A key research topic is about the placement of functions, distributed across the network (the processing graph), and the related optimisation of usage of resources . This requires joint consideration of, and trading off between, different types of resource capabilities – compute, storage, networking, (green) energy – as well as other constraints, such as time-sensitivity of the application, and the users' appetite for risk and cost. To take a simple example: it may be better to move the function to be near the data, rather than shift all the data to a function running in the central cloud or an edge cloud, particularly if the volume of data produced makes it impossible to move in its original form. Since the optimisation is an NP-complete problem, it is not about finding a unique global optimum, but a reasonably good enough solution. The optimisation also has to take account of some additional challenges. One is how to include resilience as a key criteria (since producing densely connected edge networks is probably prohibitively expensive) — including solutions that are stable for some degree of failures, and the ability to safely transition from one solution to another without violating service level criteria, or via a risky, unstable intermediate point.

Another challenge is how to include the cost of change — including the time to compute the solution vs. the time before the configuration has to change (due to changes in the workload and/or the resource availability), the cost of spinning resources up and down, and – on a longer timescale – the optimisation of value (rather than cost) through time. A third issue is the inclusion of policy preferences and constraints of the CFN operator and the users. Finally, it is possible that the resources may be contributed by different owners (for example, owners of the RAN, access network, distributed compute, compute in a POP), whilst spot price manipulation should be prevented.

Another topic fruitful for research could be to re-consider the balance of the management and control planes . Traditionally the control plane operates automatically and autonomously, driven by protocols and algorithms, whilst the management plane operates with manual intervention and specialisation. Can orchestration, in its broadest sense, be re-architected so that it is a control plane set of functionality that automatically converts the "intent" of the users (and networks) into components instantiated on the hosting infrastructure?

Finally it is worthwhile to re-consider earlier approaches that were abandoned, in light of compute-first networking. For example, I3, active networking, and network coding.

In summary, today the building blocks and concepts for CFN are only partially understood. There is great potential and significant challenges to achieve CFN as a platform that can enable new use cases and business opportunities.

References

1 An Overview of Privacy in Machine Learning Emiliano De Cristofaro, March 2020. https://emilianodc.com/PAPERS/privacyML.pdf
2 Rearchitecting Kubernetes for the Edge, Andrew Jeffery et al, EdgeSys 2021. https://arxiv.org/abs/2104.02423
3 The ephemerizer: Making data disappear, Radia Perlman, 2005. https://www.researchgate.net/publication/228360589_The_ephemerizer_Making_data_disappear

4 Chaffing and winnowing, Wikipedia. https://en.wikipedia.org/wiki/Chaffing_and_winnowing
5 How mass testing helped limit the spread of COVID-19 at the University of Cambridge, Craig Brierley, 2020. https://www.cam.ac.uk/stories/screeningprogramme
6 On the Utilisation of Persistent Programming Environments, Richard Cooper, PhD thesis, 1989. http://theses.gla.ac.uk/77963/
7 Membrane computing, Wikipedia. https://en.wikipedia.org/wiki/Membrane_computing
8 Opportunistic content sharing applications, Ott et al, NoM12. http://www.netlab.tkk.fi/~jo/papers/2012-nom-floating-api.pdf
9 Efficient Publish/Subscribe-Based Multicast for Opportunistic Networking with Self-Organized Resource Utilization, Greifenberg et al, AINAW08. https://dl.acm.org/doi/10.1109/WAINA.2008.255
10 A reliable multicast framework for light-weight sessions and application level framing, Floyd et al, ACM CCR 1995. https://dl.acm.org/doi/10.1145/217391.217470
11 Manufacturer Usage Description Specification, IETF, 2019. https://datatracker.ietf.org/doc/html/rfc8520
12 Automated valet parking video:https://www.youtube.com/watch?v=0q-kLX5ISZA
13 An Advanced Teleoperation Testbed, oss et al, 2008. https://link.springer.com/chapter/10.1007/978-3-540-75404-6_28
14 Autocloud, or how to use your car battery to save energy! Jon Crowcroft. https://www.cl.cam.ac.uk/~jac22/talks/autocloud.pdf
15 How Internet Concepts and Technologies Can Help Green and Smarten the Electrical Grid, Keshav et al, 2010. http://svr-sk818-web.cl.cam.ac.uk/keshav/papers/10/greennet.pdf
16 Modelling Incentives for Collaboration in Mobile Ad Hoc Networks, Crowcroft et al, WiOpt'03. https://hal.archives-ouvertes.fr/inria-00466747/
17 In-situ OAM IPv6 Options, Bhandari et al, work in progress. https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-ipv6-options
18 IOAM (IPv6) in Linux kernel, Iurman. https://datatracker.ietf.org/meeting/105/materials/slides-105-ippm-5-ioam-implementation-00
19 Scalable Network Management Using Lightweight Programmable Network Services, Calvert, 2006. https://link.springer.com/article/10.1007/s10922-005-9013-6
20 Pervasive Debugging, Hand et al. https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/pdb/papers/2004.escience.pdf
21 Generic Multicast Transport Services: Router Support for Multicast Applications, Cain et al. 2000. https://www.semanticscholar.org/paper/Generic-Multicast-Transport-Services%3A-Router-for-Cain-Towsley/aecc9f22d0b0dfd7b416a1837d051bfb92a4a27a
22 Tiny Packet Programs for low-latency network control and Monitoring, Jeyakumar et al, Hotnets 13. https://conferences.sigcomm.org/hotnets/2013/papers/hotnets-final47.pdf
23 Probabilistic Synchronous Parallel, Wang et al, 2017. https://arxiv.org/abs/1709.07772
24 NaaS: Application-Specific Network Services, project page. https://lsds.doc.ic.ac.uk/projects/naas
25 The Case For In-Network Computing On Demand, Tokusashi et al, Eurosys 19. https://dl.acm.org/doi/10.1145/3302424.3303979
26 P4xos: Consensus as a Network Service, Dang et al, ToN 2020. https://ieeexplore.ieee.org/abstract/document/9095258

27 Partitioned Paxos via the Network Data Plane, Dang et al, 2019. https://arxiv.org/abs/1901.08806
28 From photons to big-data applications: terminating terabits, Zilberman et al, 2016. https://royalsocietypublishing.org/doi/full/10.1098/rsta.2014.0445
29 Taurus: An Intelligent Data Plane, Swamy et al, 2020. https://arxiv.org/abs/2002.08987
30 CHANGING CONDITIONS FOR NEURAL NETWORK PROCESSING, Hemsoth, 2020. https://www.nextplatform.com/2020/04/07/changing-conditions-for-neural-\network-processing/
31 IPU processors, webpaeg. https://www.graphcore.ai/products/ipu
32 NVLink and NVSwitch, webpage. https://www.nvidia.com/en-gb/data-center/nvlink/
33 Introduction to Actors, webpage. https://doc.akka.io/docs/akka/current/typed/actors.html
34 Fibbing: Central Control over Distributed Routing, Tilmans et al, 2016. https://datatracker.ietf.org/meeting/97/materials/slides-97-rtgarea-\fibbing-anrp-00
35 What is Autonomic Computing? Ghosh, 2020. https://thecustomizewindows.com/2020/04/what-is-autonomic-computing/
36 Rate of convergence of increasing path-vector routing protocols, Daggitt et al. https://www.cl.cam.ac.uk/~tgg22/publications/icnp_2018.pdf
37 Prism: Proxies without the Pain, Hayakawa et al, NSDI 21. https://www.usenix.org/conference/nsdi21/presentation/hayakawa
38 PASTE: A Networking Programming INterface for Non-Volatile Main Memory, Honda et al, NDSI 18. https://www.usenix.org/conference/nsdi18/presentation/honda
39 Bigraphs and Their Algebra, Milner, 2008. https://core.ac.uk/download/pdf/82274932.pdf
40 Modelling and verification of large-scale sensor network infrastructures, Sevegnani et al, ICECCS 2018 https://ieeexplore.ieee.org/document/8595061
41 Compute First Networking: Distributed Computing meets ICN, Krol et al, ICN 2019. https://dl.acm.org/doi/10.1145/3357150.3357395
42 XenoSearch: Distributed Resource Discovery in the XenoServer Open Platform, Spence et al. https://www.cl.cam.ac.uk/research/srg/netos/papers/2003-xenosearch.pdf
43 Resolution strategies for serverless computing in information centric networking, Scherb, PhD thesis 2020. https://edoc.unibas.ch/77823/
44 AntidoteDB, website. https://www.antidotedb.eu
45 MobilityFirst Future Internet Architecture Project, Seskar et al, 2011. https://www.researchgate.net/publication/220785899_MobilityFirst_Future_Internet_Architecture_Project
46 Collective communication, Message Passing Interface Forum, 1995. https://www.mpi-forum.org/docs/mpi-1.1/mpi-11-html/node64.html#Node64
47 Coflow project website. https://amplab.cs.berkeley.edu/tag/coflow/
48 Scaling Replicated State Machines with Compartmentalization, Whittaker et al, 2020. https://arxiv.org/abs/2012.15762
49 Marc Brooker's blog. https://brooker.co.za/blog/
50 Amazon S3 Object Lambda website. https://aws.amazon.com/s3/features/object-lambda/
51 Development and management of collective network and cloud computing infrastructures, Viñas, PhD 2019. https://dsg.ac.upc.edu/rogerb-phd

52 Age of Information: An Introduction and Survey, Yates et al, JSAC 2021. https://ieeexplore.ieee.org/document/9380899

53 The PGM Reliable Multicast Protocol, Gemmell et al. https://www.cl.cam.ac.uk/teaching/1213/R02/papers/pgm_ieee_network.pdf

54 MLIR: A Compiler Infrastructure for the End of Moore's Law, Lattner et al, 2020. https://arxiv.org/abs/2002.11054

55 In-Network Computing for App-Centric Micro-Services, Trossen et al, 2021. Work in progress. https://datatracker.ietf.org/doc/html/draft-sarathchandra-coin-appcentres

56 Milking the Cache Cow With Fairness in Mind, Wang et al, IEEE Transactions on Networking, 2017. https://ieeexplore.ieee.org/document/7945473

57 What the Fitbit Saw During That 4.4 Quake, Miller, KQED 2018. https://www.kqed.org/futureofyou/438270/what-your-fitbit-and-smartphone-saw\-during-that-4-5-quake

58 On the duality of resilience and privacy, Crowcroft, Proceedings of Royal Society, 2014. https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2014.0862

59 Critical regions and region-disjoint paths in a network, Trajanovski et al, IFIP 2013. https://ieeexplore.ieee.org/document/6663507

60 Climate change: science and solutions, Royal Society, 2021. https://royalsociety.org/topics-policy/projects/climate-change-science-solutions/

61 Programmable Network Data Planes, Antichi et al, Report from Dagstuhl Seminar 19141, 2019. https://drops.dagstuhl.de/opus/volltexte/2019/11295/pdf/dagrep_v009_i003_p178_19141.pdf

62 A Foundation of Parallel Programming, Misra, 1988. https://www.cs.utexas.edu/users/misra/psp.dir/Marktoberdorf-88.pdf

63 On chemical and self-healing networking protocols, Meyer. PhD thesis, 2011. http://edoc.unibas.ch/diss/DissB_9383

64 Perspective: Energy Landscapes for Machine Learning, Ballard et al, 2017. https://arxiv.org/pdf/1703.07915.pdf

65 ACM FAccT conference (Fairness, Accountability and Transparency) https://facctconference.org/index.html

66 Counterfactual Fairness, Kusner et al, 2018 https://arxiv.org/abs/1703.06856

67 Path-Specific Counterfactual Fairness, Chiappa et al, 2018. https://deepmind.com/research/publications/path-specific-counterfactual-fairness

68 Liquid Data Networking, Byers et al, ACN 2021 https://dl.acm.org/doi/10.1145/3405656.3418710

69 Jayadev Misra; A Foundation of Parallel Programming; https://www.cs.utexas.edu/users/misra/psp.dir/Marktoberdorf-88.pdf

70 Vimalkumar Jeyakumar, Mohammad Alizadeh, Changhoon Kim, and David Mazières. 2013. Tiny packet programs for low-latency network control and monitoring. In Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII). Association for Computing Machinery, New York, NY, USA, Article 8, 1–7. https://doi.org/10.1145/2535771.2535780

71 Steven Hand, Tim Harris, Alex Ho, Rashid Mehmood; Pervasive Debugging https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/pdb/papers/2004.escience.pdf

72 Cain, B. and D. Towsley. "Generic Multicast Transport Services: Router Support for Multicast Applications." NETWORKING (2000).https://www.semanticscholar.org/paper/Generic-Multicast-Transport-Services%3A-Router-for-Cain-Towsley/aecc9f22d0b0dfd7b416a1837d051bfb92a4a27a

73 Liang Wang, Ben Catterall, Richard Mortier; Probabilistic Synchronous Parallel; 2017 https://arxiv.org/abs/1709.07772

74 Calvert, K.L., Griffioen, J. & Wen, S. Scalable Network Management Using Lightweight Programmable Network Services. J Netw Syst Manage 14, 15–47 (2006). https://doi.org/10.1007/s10922-005-9013-6

75 Yuta Tokusashi, Huynh Tu Dang, Fernando Pedone, Robert Soulé, and Noa Zilberman. 2019. The Case For In-Network Computing On Demand. In Proceedings of the Fourteenth EuroSys Conference 2019 (EuroSys '19). Association for Computing Machinery, New York, NY, USA, Article 21, 1–16. https://doi.org/10.1145/3302424.3303979

76 H. T. Dang et al., "P4xos: Consensus as a Network Service," in IEEE/ACM Transactions on Networking, vol. 28, no. 4, pp. 1726-1738, Aug. 2020, doi: 10.1109/TNET.2020.2992106.

77 Dang, Huynh Tu, Pietro Bressana, Han Wang, Ki-Suh Lee, Noa Zilberman, Hakim Weatherspoon, M. Canini, F. Pedone and R. Soulé. "Partitioned Paxos via the Network Data Plane." ArXiv abs/1901.08806 (2019). https://arxiv.org/abs/1901.08806

78 Zilberman Noa, Moore Andrew W. and Crowcroft Jon A. 2016; From photons to big-data applications: terminating terabits; Phil. Trans. R. Soc. A.3742014044520140445; http://doi.org/10.1098/rsta.2014.0445

79 Swamy, Tushar, Alexander Rucker, M. Shahbaz and K. Olukotun. "Taurus: An Intelligent Data Plane." ArXiv abs/2002.08987 (2020): https://arxiv.org/abs/2002.08987

80 Nicole Hemsoth; Changing Conditions For Neural Network Processing; April 2020 https://www.nextplatform.com/2020/04/07/changing-conditions-for-neural-\network-processing/

81 Whittaker, M., Ailijiang, A., Charapko, A., Demirbas, M., Giridharan, N., Hellerstein, J., Howard, H., Stoica, I., & Szekeres, A. (2020). Scaling Replicated State Machines with Compartmentalization Technical Report. ArXiv, abs/2012.15762 https://arxiv.org/abs/2012.15762

Remote Participants

- Chris Adeniyi-Jones
ARM Ltd. – Cambridge, GB
- Laura Al Wardani
FH Emden, DE
- Uthra Ambalavanan
Robert Bosch GmbH – Renningen, DE
- Gianni Antichi
Queen Mary University of London, GB
- Roberto Bifulco
NEC Laboratories Europe – Heidelberg, DE
- Olivier Bonaventure
UC Louvain, BE
- Kenneth L. Calvert
University of Kentucky – Lexington, US
- Jon Crowcroft
University of Cambridge, GB
- Philip Eardley
BT Applied Research – Ipswich, GB
- T M Rayhan Gias
FH Emden, DE
- Tim Harris
Microsoft Research – Cambridge, GB
- Jianfei He
City University – Hong Kong, HK
- Micchio Honda
University of Edinburgh, GB
- Teemu Kärkkäinen
TU München, DE
- Jussi Kangasharju
University of Helsinki, FI
- Namseok Ko
ETRI – Daejeon, KR
- Michal Król
City – University of London, GB
- Ike Kunze
RWTH Aachen, DE
- Dirk Kutscher
FH Emden, DE
- Julie McCann
Imperial College London, GB
- Jag Minhas
Sensing Feeling – London, GB
- Marie-Jose Montpetit
Concordia University – Montreal, CA
- Naresh Nayak
Robert Bosch GmbH – Stuttgart, DE
- Erik Nordmark
Zededa – San Jose, US
- David Oran
Network Systems Research & Design – Cambridge, US
- Jörg Ott
TU München, DE
- Andy Reid
BT – Ipswich, GB
- Eve M. Schooler
Intel – Santa Clara, US
- Peer Stritzinger
Peer Stritzinger GmbH – Maisach, DE
- Christian Tschudin
Universität Basel, CH
- Klaus Wehrle
RWTH Aachen, DE
- Cedric Westphal
Futurewei – Santa Clara, US
- Peter Willis
BT – Ipswich, GB
- Chenren Xu
Peking University, CN
- Noa Zilberman
University of Oxford, GB