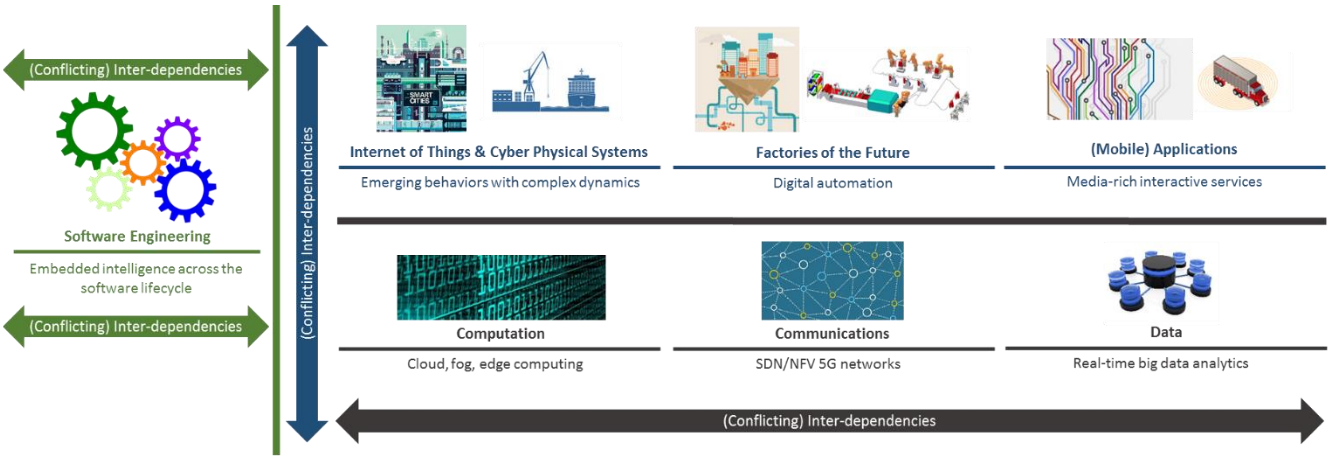
Beyond Clouds: Complete Computing

Keith Jeffery, Dimosthenis Kyriazis, Lutz Schubert

Rationale

Cloud computing matures and is increasingly being considered a baseline technology / enabler / utility, while being "extended" to address the requirements of emerging applications (e.g. mobility through fog computing) and incorporate the advancements of other technologies (e.g. computations on the network through edge computing). The same applies for *communications* - enabling SDN/NFV 5G networking - and *data-oriented technologies* - enabling real-time big data analytics. Nonetheless, the advancements and the corresponding optimizations achieved in the aforementioned cases are (in their majority) "isolated" in each technological domain (i.e. computation, communications, data) and mainly *utilize each other in a "black-box" way*. While they all aim at serving the applications, the tools and frameworks in each domain exploit the information from the application space (i.e. following a top-down approach) to self-optimize the operation and provision of the corresponding underlying services *without respecting the adaptations in or impact of the respective other domains*. Representative examples are orchestration / choreography services in clouds, SDNs in networks and incremental queries in data management, which all regard the application as a behavioural model isolated within their domain – disrespecting thereby that e.g. changes on the network layer will affect the data availability and therefore expected locality. Moreover, the future application landscape is expected to be comprised of IoT, CPS as well as FoF and mobile applications, thus posing even more requirements on computation, communication and data spaces: IoT and CPS are mainly characterised by complex dynamics and emerging behaviors in response to real-world events; FoF aim at digital automation incorporating unconventional technologies; while mobile applications provide media-rich interactive services. Are the highly-advanced but yet isolated computation, communication and data technologies able to address the requirements of the application plane in an efficient way?



As highlighted in the figure above, there are 4 types of inter-dependencies (covering both horizontal and vertical cases) not addressed by current environments: (i) between computation, communications and data technologies, (ii) between applications and computation/communications/data, (iii) software engineering techniques and applications, and (iv) between software engineering techniques and computation/communications/data. The vision of *Complete Computing is to close the control loops at all scales*, thus addressing all of the 4 cases of inter-dependencies and enabling high degrees of efficiency, quality and automation. It should be noted that *identifying, understanding and controlling these dependencies and their impact,* includes also the case of conflicting ones - in terms of configuration, provision and optimization in each of the corresponding domains. Given that these dependencies affect both the *functional and the non-functional properties* of these domains, the following paragraphs provide representative examples to highlight the importance of the dependencies consideration and the realization of the complete computing vision.

Representative functional challenge: Soft and hard real-time

Future applications in the cloud and generally on the internet – no matter whether Internet of Things, Factories of the Future or (media) services will demonstrate an increasing demand for meeting time barriers as part of their Quality of Service. In particular the Internet of Things is designated by a large amount of media content, be that audio and video, or different types of sensor data streams, ranging from temperature to earthquake detection. Most of these applications fall in the context of *soft* real time, where a barrier can, but should not be missed. However, cases such as disaster supervision (e.g. earthquake sensors, fire alarms) and in general Factories of the Future classify as *hard* real time, as missing the time barrier may be critical and can lead to immense cost and even loss of life.

The real-time capability of an application is not only defined by the algorithm (i.e. software engineering dependency), but in fact by multiple, strongly correlated factors: (i) location, replication and scale of the application directly impacts on latency and performance of the algorithm (i.e. computation dependency), this is directly dependent on (ii) network and communication characteristics, including reservation of bandwidth, congestion etc (i.e. communications dependency) which obviously is influenced by (iii) the size of communication, i.e. the amount of data (i.e. data dependency). The amount and structure of data furthermore has a direct impact on the algorithmic complexity and therefore on the processing requirements in the first instance (i.e. software engineering dependency).

The hard challenge thereby is to define applications in a fashion that allow the execution and configuration of (i) the computation instances, (ii) the communications / network and (iii) the data to be communicated (including prefetching and distribution of data). These aspects must be easily expressible in a common application model that is closely linked to the actual code to allow adaptation of quality, control of the network etc - which goes beyond current software engineering models.

Representative non-functional challenge: Security and privacy

The demands and requirements for any future application scenario are clear. The number 1 requirement is security and privacy. This will persist into Complete Computing unless addressed. The problem comes in several parts and is a quite presentative aspect of software, applications, computation, communications and data conflicting inter-dependencies: (i) Legislation is heterogeneous therefore providing a homogeneous end-user and system administrator experience across all domains is challenging, (ii) Multi-factor authentication and dynamic role-based access control are still nascent, (iii) There exist several well-known AAAI solutions – all heterogeneous – and systems to make them transparent to the end-user or application are only just appearing, (iv) Encryption is well-known and mature - however the potential use of more powerful decryption (especially brute force techniques using e.g. quantum computing or Biocomputing) is on the horizon, (v) The use of dynamic data partitioning for security and privacy is nascent, (vi) Sociological attacks (like phishing) are increasingly common and sophisticated leading to compromises, (vii) The sheer complexity of modern systems with multiple security/privacy regimes opens potential compromises, (viii) Isolated systems development methods and workflows may lead to inbuilt weaknesses such as “back door” entry.

The research required is in end-to-end security from end-user and system administrator (including sociological aspects) through user interfaces, APIs, applications and on through middleware to platforms and the underlying networks, data storage and computing hardware. The need is for dynamically reconfigurable (by command or irregular time triggers) but vertically integrated security and privacy appropriate to the perceived risk at all components of the complete computing. Related research concerns provision of security and privacy when an end-user utilises single-sign-on across a range of applications distributed over multiple middleware and platform stacks.