The concepts and technologies for network-centric computing and content evolved through the years

and led to several large-scale distributed system developments:

• The Web and the semantic Web are expected to support composition of services (not necessarily

computational services) available on the Web.1

• The Grid, initiated in the early 1990s by National Laboratories and Universities, is used primarily

for applications in the area of science and engineering.

• Computer clouds, promoted since 2005 as a form of service-oriented computing by large IT companies,

are used for enterprise computing, high-performance computing, Web hosting, and storage

for network-centric content.

The need to share data from high-energy physics experiments motivated Sir Tim Berners-Lee, who

worked at the European Organization for Nuclear Research (CERN) in the late 1980s, to put together

the two major components of the World Wide Web: HyperText Markup Language (HTML) for data

description and HyperText Transfer Protocol (HTTP) for data transfer. The Web opened a new era in

data sharing and ultimately led to the concept of network-centric content.

The semantic Web2 is an effort to enable laypeople to more easily find, share, and combine information

available on the Web. In this vision, the information can be readily interpreted by machines,

so machines can perform more of the tedious work involved in finding, combining, and acting upon

information on theWeb. Several technologies are necessary to provide a formal description of concepts,

terms, and relationships within a given knowledge domain; they include the Resource Description

Framework (RDF), a variety of data interchange formats, and notations such as RDF Schema (RDFS)

and the Web Ontology Language (OWL). Gradually, the need to make computing more affordable and to liberate users from the concerns

regarding system and software maintenance reinforced the idea of concentrating computing resources

in data centers. Initially, these centers were specialized, each running a limited palette of software

systems as well as applications developed by the users of these systems. In the early 1980s major

research organizations such as the National Laboratories and large companies had powerful computing

centers supporting large user populations scattered throughout wide geographic areas. Then the idea

to link such centers in an infrastructure resembling the power grid was born; the model known as

network-centric computing was taking shape.

A computing grid is a distributed system consisting of a large number of loosely coupled, heterogeneous,

and geographically dispersed systems in different administrative domains. The term computing

grid is a metaphor for accessing computer power with similar ease as we access power provided by the

electric grid. Software libraries known as middleware have been furiously developed since the early

1990s to facilitate access to grid services.

The vision of the grid movement was to give a user the illusion of a very large virtual supercomputer.

The autonomy of the individual systems and the fact that these systems were connected by wide-area

networks with latency higher than the latency of the interconnection network of a supercomputer posed

serious challenges to this vision. Nevertheless, several “Grand Challenge” problems, such as protein

folding, financial modeling, earthquake simulation, and climate and weather modeling, run successfully

on specialized grids. The Enabling Grids for Escience project is arguably the largest computing grid;

along with the LHC Computing Grid (LCG), the Escience project aims to support the experiments using

the Large Hadron Collider (LHC) at CERN which generate several gigabytes of data per second, or

10 PB (petabytes) per year.

In retrospect, two basic assumptions about the infrastructure prevented the grid movement from

having the impact its supporters were hoping for. The first is the heterogeneity of the individual systems

interconnected by the grid; the second is that systems in different administrative domains are expected to

cooperate seamlessly. Indeed, the heterogeneity of the hardware and of system software poses significant

challenges for application development and for application mobility. At the same time, critical areas

of system management, including scheduling, optimization of resource allocation, load balancing, and

fault tolerance, are extremely difficult in a heterogeneous system. The fact that resources are in different

administrative domains further complicates many already difficult problems related to security and

resource management. Although very popular in the science and engineering communities, the grid

movement did not address the major concerns of the enterprise computing communities and did not

make a noticeable impact on the IT industry.

Cloud computing is a technology largely viewed as the next big step in the development and deployment

of an increasing number of distributed applications. The companies promoting cloud computing

seem to have learned themost important lessons from the gridmovement. Computer clouds are typically

homogeneous. An entire cloud shares the same security, resource management, cost and other policies,

and last but not least, it targets enterprise computing. These are some of the reasons that several agencies

of the US Government, including Health and Human Services (HHS), the Centers for Disease Control

(CDC), the National Aeronautics and Space Administration (NASA), the Navy’s Next Generation

Enterprise Network (NGEN), and the Defense Information Systems Agency (DISA), have launched

cloud computing initiatives and conduct actual system development intended to improve the efficiency

and effectiveness of their information processing needs.

The term content refers to any type or volume of media, be it static or dynamic, monolithic or

modular, live or stored, produced by aggregation, or mixed. Information is the result of functions

applied to content. The creation and consumption of audio and visual content are likely to transform

the Internet to support increased quality in terms of resolution, frame rate, color depth, and stereoscopic

information, and it seems reasonable to assume that the Future Internet3 will be content-centric. The

content should be treated as having meaningful semantic connotations rather than a string of bytes;

the focus will be the information that can be extracted by content mining when users request named

data and content providers publish data objects. Content-centric routing will allow users to fetch the

desired data from the most suitable location in terms of network latency or download time. There are

also some challenges, such as providing secure services for content manipulation, ensuring global rights

management, control over unsuitable content, and reputation management.

Network-centric computing and network-centric content share a number of characteristics:

• Most applications are data-intensive. Computer simulation becomes a powerful tool for scientific

research in virtually all areas of science, from physics, biology, and chemistry to archeology. Sophisticated

tools for computer-aided design, such as Catia (Computer Aided Three-dimensional Interactive

Application), are widely used in the aerospace and automotive industries. Thewidespread use of sensors

contributes to increases in the volume of data. Multimedia applications are increasingly popular;

the ever-larger media increase the load placed on storage, networking, and processing systems.

• Virtually all applications are network-intensive. Indeed, transferring large volumes of data requires

high-bandwidth networks; parallel computing, computation steering,4 and data streaming are examples

of applications that can only run efficiently on low-latency networks.

• The systems are accessed using thin clients running on systems with limited resources. In June 2011

Google released Google Chrome OS, designed to run on primitive devices and based on the browser

with the same name.

• The infrastructure supports some form of workflow management. Indeed, complex computational

tasks require coordination of several applications; composition of services is a basic tenet ofWeb 2.0.

The advantages of network-centric computing and network-centric content paradigms are, at the

same time, sources for concern; we discuss some of them:

• Computing and communication resources (CPU cycles, storage, network bandwidth) are shared and

resources can be aggregated to support data-intensive applications. Multiplexing leads to a higher

resource utilization; indeed, when multiple applications share a system, their peak demands for

resources are not synchronized and the average system utilization increases. On the other hand,

the management of large pools of resources poses new challenges as complex systems are subject

to phase transitions. New resource management strategies, such as self-organization, and decisions

based on approximate knowledge of the state of the system must be considered. Ensuring quality-ofservice

(QoS) guarantees is extremely challenging in such environments because total performance

isolation is elusive.

• Data sharing facilitates collaborative activities. Indeed, many applications in science, engineering,

and industrial, financial, and governmental applications require multiple types of analysis of shared

data sets and multiple decisions carried out by groups scattered around the globe. Open software

development sites are another example of such collaborative activities. Data sharing poses not only

security and privacy challenges but also requires mechanisms for access control by authorized users

and for detailed logs of the history of data changes.

• Cost reduction. Concentration of resources creates the opportunity to pay as you go for computing

and thus eliminates the initial investment and reduces significantly the maintenance and operation

costs of the local computing infrastructure.

• User convenience and elasticity, that is the ability to accommodate workloads with very large peakto-

average ratios.

It is very hard to point out a single technological or architectural development that triggered the

movement toward network-centric computing and network-centric content. This movement is the result

of a cumulative effect of developments inmicroprocessor, storage, and networking technologies coupled

with architectural advancements in all these areas and, last but not least, with advances in software

systems, tools, programming languages, and algorithms to support distributed and parallel computing.

Through the years we have witnessed the breathtaking evolution of solid-state technologies which

led to the development of multicore and many-core processors. Quad-core processors such as the AMD

Phenom II X4, the Intel i3, i5, and i7 and hexa-core processors such as the AMDPhenom II X6 and Intel

Core i7 Extreme Edition 980X are now used in the servers populating computer clouds. The proximity

of multiple cores on the same die allows the cache coherency circuitry to operate at a much higher clock

rate than would be possible if the signals were to travel off-chip.

Storage technology has also evolved dramatically. For example, solid-state disks such as RamSan-

440 allow systems to manage very high transaction volumes and larger numbers of concurrent users.

RamSan-440 uses DDR2 (double-data-rate) RAM to deliver 600,000 sustained random input/output

operations per second (IOPS) and over 4 GB/s of sustained random read or write bandwidth, with

latency of less than 15 microseconds, and it is available in 256 GB and 512 GB configurations. The

price of memory has dropped significantly; at the time of this writing the price of a 1 GB module for a

PC is approaching $10. Optical storage technologies and Flash memories are widely used nowadays.

The thinking in software engineering has also evolved and new models have emerged. The three-tier

model is a software architecture and a software design pattern. The presentation tier is the topmost

level of the application; typically, it runs on a desktop PC or workstation, uses a standard graphical user

interface (GUI) and displays information related to services such as browsing merchandise, purchasing

products, and managing shopping cart contents. The presentation tier communicates with other tiers by

sending the results to the browser/client tier and all other tiers in the network. The application/logic tier

controls the functionality of an application and may consist of one or more separate modules running

on a workstation or application server; it may be multitiered itself, in which case the architecture is

called an n-tier architecture. The data tier controls the servers where the information is stored; it runs

a relational database management system (RDBMS) on a database server or a mainframe and contains

the computer data storage logic. The data tier keeps data independent from application servers or

processing logic and improves scalability and performance. Any of the tiers can be replaced independently;

for example, a change of operating system in the presentation tier would only affect the user

interface code.