



Chesapeake Chapter INCOSE

International Council on Systems Engineering

[Print Article](#)
[Sign up for the Newsletter](#)
[Go to Back Issues](#)

May
2011

Feature Article

Cloud Computing What's the hype all about?

by Kurt Christensen



Figure 1. I'm not expendable, I'm not Stupid, and I'm not going.

Like most terms with *buzz*, the term **Cloud Computing** has been overloaded to mean many things. Generally, it means the application of replicated commodity hardware (or virtual machines) working together to, more or less, appear as a single service.

Doing a web search on '[cloud computing companies](#)' yields a long list. Check out the *Business Innovations* article [90 Cloud Computing Companies to Watch in 2011](#). Here is a short list of the more recognizable cloud computing service providers: [Amazon](#); [Google](#); [Microsoft](#); and [Rackspace](#).

If some sort of extensive computing capability is needed, but you don't want to provide in-house support, these "services"

*may be leased as a **utility** from vendors. From the point of view leasing cloud services, you pay only for what you use.*

- Need more compute power? Provision more machines.
- Need more storage? Provision more disks.
- Need more load-handling capability? Provision more servers.
- Need to save expense? Provision less. Pay only for what you use.

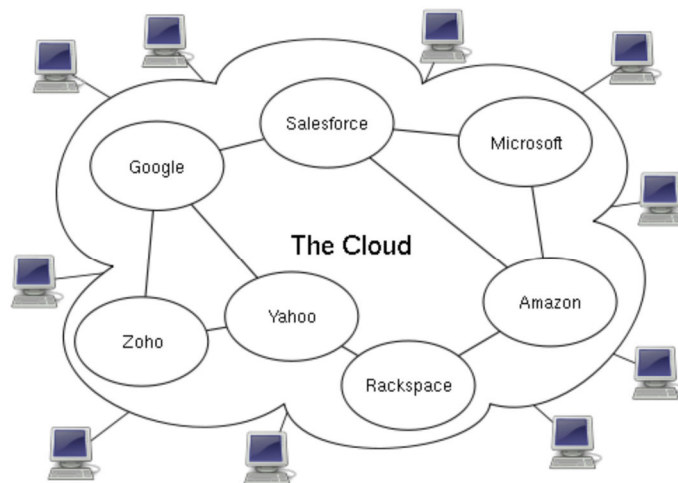


Figure 2. Cloud Computing Conceptual Diagram

Quoting Peter Mell and Tim Grance from NIST

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

[Return to Newsletter](#)



Mark your Calendars with these upcoming Chapter events:

Wednesday 18 May 2011

The Challenge of Cloud Computing

Harry J. Foxwell, Ph.D;
Principal Consultant; Oracle Corp

- New Location: CyberCore Inc.
- Free Dinner

[>>download May flyer HERE<<](#)

The Chesapeake Chapter is always looking for volunteers to speak at our upcoming meetings! Please contact our Programs Director, [Mr. Donald York](#), if you would like the opportunity to speak or can recommend someone.

Up Coming Conferences

May 01 - 05, 2011
Detroit, MI

[Department of Defense Intelligence Information Systems \(DoDIIS\) Worldwide Conference](#)

Hosted annually by the DIA Directorate for Information Management (DS) and Chief Information Officer, DoDIIS Worldwide provides a unique opportunity for defense intelligence community members and industry peers to come together for knowledge

Cloud computing, extends the notion of [Software as a Service \(SaaS\)](#), where applications are removed from the desktop and moved to remote, virtual hardware, typically in a cloud. These services implement various functions like word processors, spreadsheets, photo archives and the like. Some of these services provide for collaboration among many users. This notion can be extended to [Infrastructure as a Service \(IaaS\)](#) allowing deployment of almost all business functions into a cloud.

The utility definition is rather broad. Some purists restrict the definition of "cloud computing" to distributed applications and storage. This implies single programs being distributed to multiple machines and a single file system that has been distributed.

In computing, a [distributed file system](#) or network file system is any file system that allows access to [files](#) from multiple hosts [sharing](#) via a [computer network](#). [Silberschatz, Galvin (1994). *Operating System concepts*, chapter 17 Distributed file systems. Addison-Wesley Publishing Company. ISBN 0-201-59292-4.] This makes it possible for multiple users on multiple machines to share files and storage resources.

In support of distributed applications, MapReduce is a distributed computing paradigm developed by Google. It takes advantage of computing resources that are collocated with distributed data. The term refers to two steps in processing. The map step takes an initial request and distributes data the resulting selectors and operations to machines that are likely to contain appropriate data. These machines may distribute more specific versions of the request to other machines. Selection and primary operations are performed locally on that data. The reduce steps takes the result of the map step and combines them into produce a reduced set, hence the term. All that is need within the framework is the definition of a map function and a reduce function. The framework handles distribution for you.

From [Wikipedia on MapReduce](#)

Logical view

The Map and Reduce functions of **MapReduce** are both defined with respect to data structured in (key, value) pairs. Map takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain:

$$\text{Map}(k1, v1) \rightarrow \text{list}(k2, v2)$$

The Map function is applied in parallel to every item in the input dataset. This produces a list of (k2,v2) pairs for each call. After that, the MapReduce framework collects all pairs with the same key from all lists and groups them together, thus creating one group for each one of the different generated keys. The Reduce function is then applied in parallel to each group, which in turn produces a collection of values in the same domain:

$$\text{Reduce}(k2, \text{list}(v2)) \rightarrow \text{list}(v3)$$

Each Reduce call typically produces either one value v3 or an empty return, though one call is allowed to return more than one value. The returns of all calls are collected as the desired result list.

Thus the MapReduce framework transforms a list of (key, value) pairs into a list of values. This behavior is different from the functional programming map and reduce combination, which accepts a list of arbitrary values and returns one single value that combines all the values returned by map.

It is necessary but not sufficient to have implementations of the map and reduce abstractions in order to implement MapReduce. Distributed implementations of MapReduce require a means of connecting the processes performing the Map and Reduce phases. This may be a distributed file system. Other options are possible, such as direct streaming from mappers to reducers, or for the mapping processors to serve up their results to reducers that query them.

How hard is this? An example

sharing, training and discussion of current and future information technology (IT) challenges and requirements. The conference's 2011 theme, Secure and Collaborative Intelligence in Defense of the Nation, highlights the DIA Chief Information Officer's commitment to developing and maintaining secure and reliable networks for Department of Defense personnel, services and information technology customers.

May 02 - 04, 2011
Canberra, Australia
[SETE2011 The Systems Engineering Test and Evaluation Conference](#)
SETE2011 (Systems Engineering Test and Evaluation) is a conference co-organized by The Systems Engineering Society of Australia (SESA), The Southern Cross Chapter of The International Test and Evaluation Association (ITEA) and INCOSE Australia. The theme of SETE2011 is 'Systems Engineering in the Next Decade'

June 20 - 23, 2011
[21st Annual International Symposium](#)
The INCOSE International Symposium is the premier international forum for Systems Engineering. Participants network, share ideas, knowledge and practices, and learn more about the most recent innovations, trends, experiences and issues in Systems Engineering. Paper authors, panelists and tutorial presenters are encouraged to address ways in which Systems Engineering principles and perspectives are performed today and how Systems Engineering may influence our future. Topics of value include technology insertion, process improvements, and organizational governance of the systems we make, manage, operate and maintain over their life cycle, to the benefit of mankind.

The Chesapeake Chapter of INCOSE is proud to recognize the following organizations for sponsoring our endeavors to expanding the understanding and appreciation of Systems Engineering in the local area:

The canonical example application of MapReduce is a process to count the appearances of each different word in a set of documents:

```
void map(String name, String document):
  // name: document name
  // document: document contents
  for each word w in document:
    EmitIntermediate(w, "1");

void reduce(String word, Iterator partialCounts):
  // word: a word
  // partialCounts: a list of aggregated partial counts
  int result = 0;
  for each pc in partialCounts:
    result += ParseInt(pc);
  Emit(AsString(result));
```

Here, each document is split into words, and each word is counted initially with a "1" value by the Map function, using the word as the result key. The framework puts together all the pairs with the same key and feeds them to the same call to Reduce, thus this function just needs to sum all of its input values to find the total appearances of that word.

Inspired by Google's papers on GFS and MapReduce, the Apache Software Foundation provides a software framework called Hadoop along with an associated distributed file system that supports data-intensive distributed applications. It is written in Java and is available under a [free license](#).

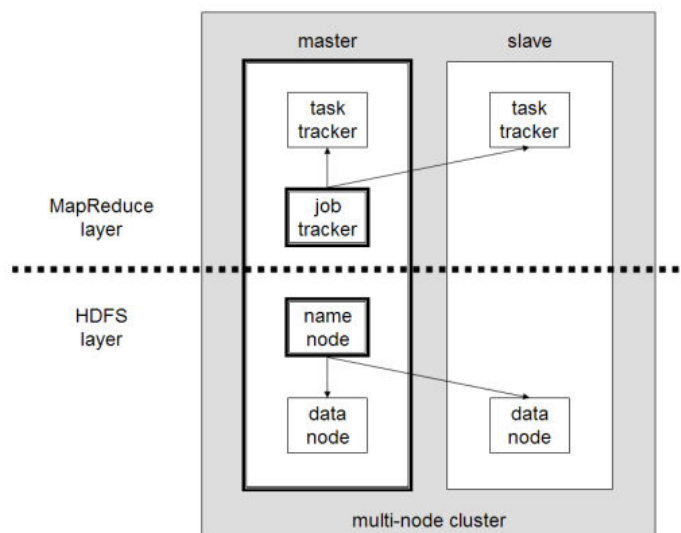


Figure 3. A multi-node Hadoop Cluster

Cloud Computing - Pros and Cons

The upside:

- Utility providers will provide redundancy and backups.
- Commodity hardware implementations are typically cheaper than mainframe/supercomputer solutions.
- Pay for only what you need, without the headaches of having to support your infrastructure.

The downside:

- Critical and private data ends up being located in places outside your control. Perhaps, overseas.
- Poorly conceived distributed applications can choke network bandwidth within the cloud.

Recommended Reading



[Cloud Computing: Fact versus Fog](#) - Grail Research

Related Technologies Prior Art

The concept of software as a service goes way back to the early 1960s. Only recently has the global network infrastructure been substantial enough to bring it to everyday life. The following are technologies that predate the implementation of modern cloud computing.

- [Distributed Computer](#)
- [Computer Cluster](#)
- [Massively Parallel Processor](#)
- [Grid Computing](#)
- [Distributed Databases, Federated Query](#)
- [Remote Procedure Calls](#)
- [Web Services](#)

A **distributed computer** (also known as a distributed memory multiprocessor) is a distributed memory computer system in which the processing elements are connected by a network. Distributed computers are highly scalable.

A **computer cluster** (see [Cluster Monkey](#)) is a group of loosely coupled computers that work together closely, so that in some respects they can be regarded as a single computer. Clusters are composed of multiple standalone machines connected by a network. While machines in a cluster do not have to be symmetric, load balancing is more difficult if they are not. The most common type of cluster is the Beowulf cluster (see also [Beowulf cluster](#)), which is a cluster implemented on multiple identical commercial off-the-shelf computers connected with a TCP/IP Ethernet local area network. Beowulf technology was originally developed by Thomas Sterling and Donald Becker. The vast majority of the [TOP500 supercomputers are clusters](#). [TOP500 Supercomputing Sites. Clusters make up 74.60% of the machines on the list. Retrieved on November 7, 2007.]



Figure 4. The Silicon Graphics Cluster-SGI; an example of a cluster computer

A **massively parallel processor (MPP)** is a single computer with many networked processors. MPPs have many of the same characteristics as clusters, but MPPs have specialized interconnect networks (whereas clusters use commodity hardware for networking). MPPs also tend to be larger than clusters, typically having "far more" than 100 processors.[29] In an MPP, "each CPU contains its own memory and copy of the operating system and application. Each subsystem communicates with the others via a high-speed interconnect." [30] Blue Gene/L, the fifth fastest supercomputer in the world according to the June 2009 TOP500 ranking, is an MPP.

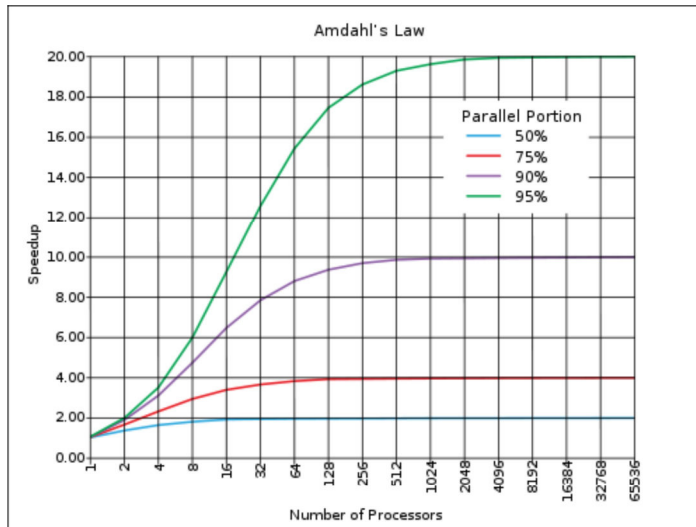


Figure 5. A graphical representation of Amdahl's law. The speed-up of a program from parallelization is limited by how much of the program can be parallelized. For example, if 90% of the program can be parallelized, the theoretical maximum speed-up using parallel computing would be 10x no matter how many processors are used.

Grid computing is the most distributed form of parallel computing. It makes use of computers communicating over the Internet to work on a given problem. Because of the low bandwidth and extremely high latency available on the Internet, grid computing typically deals only with embarrassingly parallel problems. Many grid computing applications have been created, of which SETI@home and Folding@Home are the best-known examples.[31] Most grid computing applications use middleware, software that sits between the operating system and the application to manage network resources and standardize the software interface. The most common grid computing middleware is the Berkeley Open Infrastructure for Network Computing (BOINC). Often, grid computing software makes use of "spare cycles", performing computations at times when a computer is idling.



Figure 6. A beowulf cluster is something you could do at home.

A **distributed database** is a [database](#) that is under the control of a [central database management system](#) (DBMS) in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers. Collections of data (e.g. in a database) can be distributed across multiple physical locations. A distributed database can reside on [network servers](#) on the [Internet](#), on corporate [intranets](#) or [extranets](#), or on other company [networks](#). The replication and distribution of databases improves database performance at end-user [worksites](#). [O'Brien, J. & Marakas, G.M. (2008) Management Information Systems (pp. 185-189). New York, NY: McGraw-Hill Irwin]

On the other hand a **Federated Query** mechanism provides a means to query database that is not under the control of a central database management system ([federated search](#)). Using federated search, in one query we can search multiple database at one time and arrange the informations in useful form and it returns the results to the user.

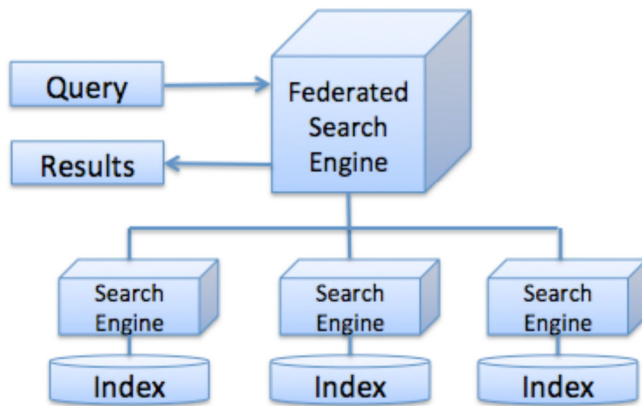


Figure 7. Topology of a Federated Query.

A [remote procedure call \(RPC\)](#) is an [inter-process communication](#) that allows a [computer program](#) to cause a [subroutine](#) or procedure to execute in another [address space](#) (commonly on another computer on a shared network) without the programmer explicitly coding the details for this remote interaction. That is, the programmer writes essentially the same code whether the subroutine is local to the executing program, or remote. When the software in question uses object-oriented principles, RPC is called [remote invocation](#) or [remote method invocation](#).

Note: there are many different (often incompatible) technologies commonly used to accomplish this. One technology, the XML-RPC was the predecessor to SOAP used in many web services.

The [W3C](#) defines a [Web service](#) as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." The W3C also states, "We can identify two major classes of Web services, REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations; and arbitrary Web services, in which the service may expose an arbitrary set of operations."



Figure 8. Web Services Architecture.

I'll stop here because the discussion of [Web Services](#) can, itself, be a deep subject.

[Return to Newsletter.](#)



Keep up with the latest news and events. Find out about our new Board of Directors. Explore our extensive library of previous lectures from our Monthly Dinner Meetings. Learn of the Benefits of Joining INCOSE. Check out Systems Engineering education in the local area. All this and more awaits you at our [INCOSE Chesapeake Chapter Website](#).

For any comments or suggestions about this newsletter please e-mail our [President, John Lewis](#) or our [Communications Officer, Paul Martin](#). We value your feedback.

Board of Director Officers, 2011
- President: Mr. John Lewis
- Past President: Mr. George Anderson
- President Elect: Gundars Osvald
- Treasurer: Mr. Glenn Gillaspay
- Secretary: Mr. Bob Berkovits

Directors at Large
- Communications: Mr. Paul Martin
- Programs: Mr. Donald York
- Membership Committee: Ms. Bhanumati Sunkara

Please use the Forward email link below so we can invite your friends to join our mailing list. Thanks in advance.

INCOSE Chesapeake Chapter © 2011