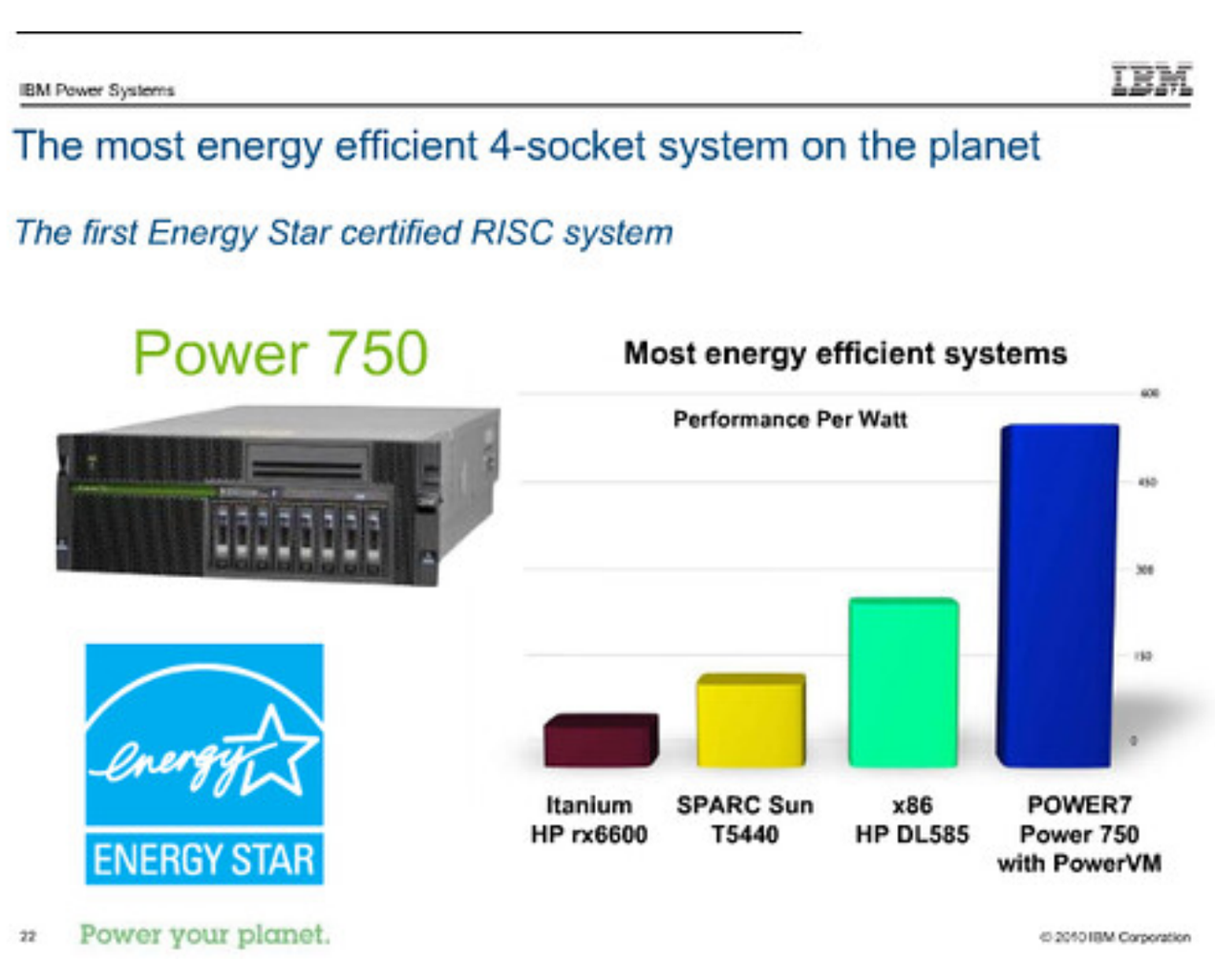
IBM Watson -- How to replicate Watson hardware and systems design for your own use in your basement

| Feb 18 2011 | Comments (12) | Visits (288441)

For the longest time, people thought that humans could not run a mile in less than four minutes. Then, in 1954, [Sir Roger Bannister] beat that perception, and shortly thereafter, once he showed it was possible, many other runners were able to achieve this also. The same is being said now about the IBM Watson computer which appeared this week against two human contestants on *Jeopardy!*

(**2014 Update:** A lot has happened since I originally wrote this blog post! I intended this as a fun project for college students to work on during their summer break. However, IBM is concerned that some businesses might be led to believe they could simply stand up their own systems based entirely on open source and internally developed code for business use. IBM recommends instead the [IBM InfoSphere BigInsights] which packages much of the software described below. IBM has also launched a new "Watson Group" that has [Watson-as-a-Service] capabilities in the Cloud. To raise awareness to these developments, IBM has asked me to rename this post from **IBM Watson - How to build your own "Watson Jr." in your basement** to the new title **IBM Watson -- How to replicate Watson hardware and systems design for your own use in your basement**. I also took this opportunity to improve the formatting layout.)



Often, when a company demonstrates new techology, these are prototypes not yet ready for commercial deployment until several years later. IBM Watson, however, was made mostly from commercially available hardware, software and information resources. As several have noted, the 1TB of data used to search for answers could fit on a single USB drive that you buy at your local computer store.

But could you fit an entire Watson in your basement? The IBM Power 750 servers used in IBM Watson earned the [EPA Energy Star] rating, and is substantially [more energy-efficient than comparable 4-socket x86, Itanium, or SPARC servers]. However, having *ninety* of them in your basement would drive up your energy bill.

That got me thinking, would it be possible to build your own question-answering system, something less fancy, less sophisticated, scaled-down for personal use? John Pultorak explained [how to build your own Apollo Guidance Computer (AGC) in your basement]. Jay Shafer explains [how to build your own house for $20K]. And a 17-year-old George Hotz figured out a [hack to unlock your Apple iPhone] over the summer in his basement.

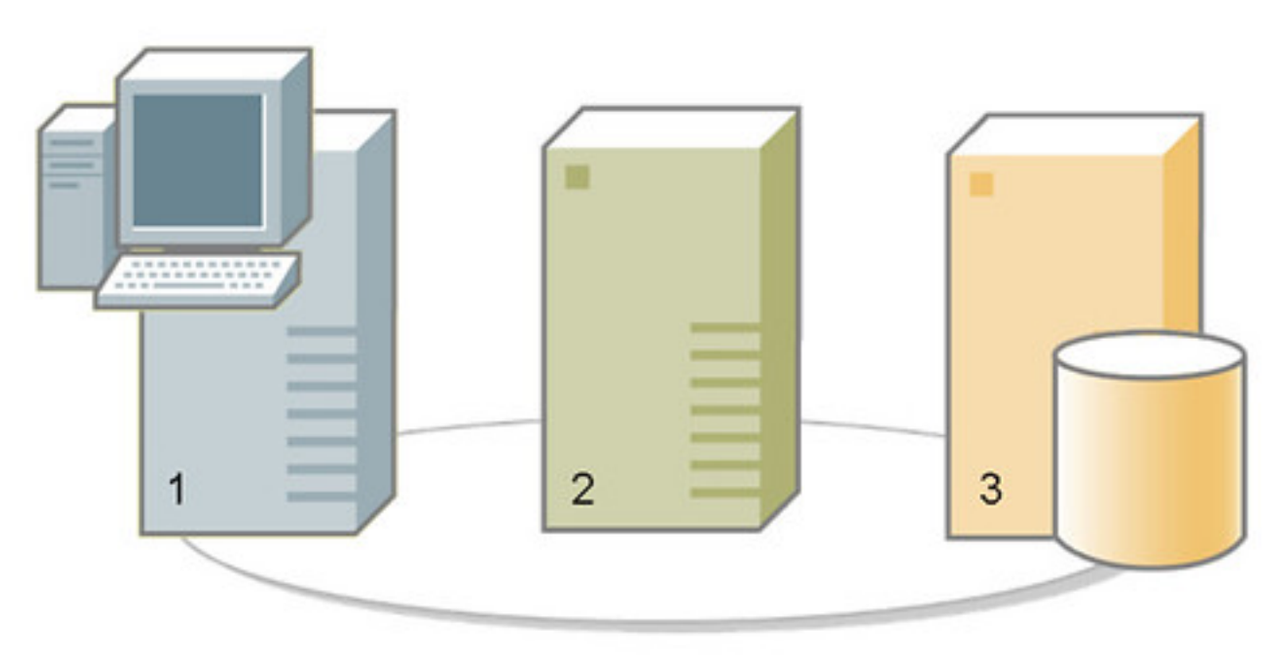
It turns out that much of the inner workings of IBM Watson were written in a series of articles in [IBM Systems Journal, Vol. 43, No. 3]. You can also read the [Wikipedia article]. Eric Brown from IBM Research will be presenting "Jeopardy: Under the Hood of IBM Watson Supercomputer" at next month's [The Linux Foundation End User Summit].

Take a look at the [IBM Research Team] to determine how the project was organized. Let's decide what we need, and what we don't in our *version for personal use*:

Role:	Do we need it for *personal use*?
Team Lead	Yes, That's you. Assuming this is a one-person project, you will act as Team Lead.
Algorithms	Yes, I hope you know computer programming!
Game Strategy	No, since *this version for personal use* won't be appearing on Jeopardy, we won't need strategy on wager amounts for the Daily Double, or what clues to pick next. Let's focus merely on a computer that can accept a question in text, and provide an answer back, in text.
Systems	Yes, this team focused on how to wire all the hardware together. We need to do that, although *this version for personal use* will have fewer components.
Speech Synthesis	Optional. For now, let's have *this version for personal use* just return its answer in plain text. Consider this *Extra Credit* after you get the rest of the system working. Consider using [eSpeak], [FreeTTS], or the Modular Architecture for Research on speech sYnthesis [MARY] Text-to-Speech synthesizers.
Annotations	Yes, I will explain what this is, and why you need it.
Information Sources	Yes, we will need to get information for *personal use* to process
Question Parsing	Yes, this team developed a system for parsing the question being asked, and to attach meaning to the different words involved.
Search Optimization	No, this team focused on making IBM Watson optimized to answer in 3 seconds or less. We can accept a slower response, so we can skip this.
Project Management	Yes, even for a one-person project, having a little "project management" never hurt anyone. I highly recommend the book [Getting Things Done: The Art of Stress-Free Productivity] by [David Allen].

(**Disclaimer:** As with any Do-It-Yourself (DIY) project, I am not responsible if you are not happy with your *version for personal use* I am basing the approach on what I read from publicly available sources, and my work in Linux, supercomputers, XIV, and SONAS. For our purposes, *this version for personal use* is based entirely on commodity hardware, open source software, and publicly available sources of information. Your *implementation* will certainly not be as fast or as clever as the IBM Watson you saw on television.)

Step 1: Buy the Hardware



Supercomputers are built as a cluster of identical compute servers lashed together by a network. You will be installing Linux on them, so if you can avoid paying extra for Microsoft Windows, that would save you some money. Here is your shopping list:

- Three x86 hosts, with the following:
  - 64-bit quad-core processor, either Intel-VT or AMD-V capable,
  - 8GB of DRAM, or larger
  - 300GB of hard disk, or larger
  - CD or DVD Read/Write drive
  - 1GbE Ethernet
- Computer Monitor, mouse and keyboard
- Ethernet 1GbE 4-port hub, and appropriate RJ45 cables
- Surge protector and Power strip
- Local Console Monitor (LCM) 4-port switch (formerly known as a KVM switch) and appropriate cables. This is optional, but will make it easier during the development. Once your *implementation* is operational, you will only need the monitor and keyboard attached to one machine. The other two machines can remain "headless" servers.

Step 2: Establish Networking

IBM Watson used Juniper switches running at 10Gbps Ethernet (10GbE) speeds, but was not connected to the Internet while playing *Jeopardy!* Instead, these Ethernet links were for the POWER7 servers to talk to each other, and to access files over the Network File System (NFS) protocol to the internal customized SONAS storage I/O nodes.

The *implementation* will be able to run "disconnected from the Internet" as well. However, you will need Internet access to download the code and information sources. For our purposes, 1GbE should be sufficient. Connect your Ethernet hub to your DSL or Cable modem. Connect all three hosts to the Ethernet switch. Connect your keyboard, video monitor and mouse to the LCM, and connect the LCM to the three hosts.

Step 3: Install Linux and Middleware

To say I use Linux on a daily basis is an understatement. Linux runs on my Android-based cell phone, my laptop at work, my personal computers at home, most of our IBM storage devices from SAN Volume Controller to XIV to SONAS, and even on my Tivo at home which recorded my televised episodes of *Jeopardy!*

For this project, you can use any modern Linux distribution that supports KVM. IBM Watson used Novel SUSE Linux Enterprise Server [SLES 11]. Alternatively, I can also recommend either Red Hat Enterprise Linux [RHEL 6] or Canonical [Ubuntu v10]. Each distribution of Linux comes in different orientations. Download the the 64-bit "ISO" files for each version, and burn them to CDs.

- Graphical User Interface (GUI) oriented, often referred to as "Desktop" or "HPC-Head"
- Command Line Interface (CLI) oriented, often referred to as "Server" or "HPC-Compute"
- Guest OS oriented, to run in a Hypervisor such as KVM, Xen, or VMware. Novell calls theirs "Just Enough Operating System" [JeOS].

For *this version for personal use*, I have chosen a [multitier architecture], sometimes referred to as an "n-tier" or "client/server" architecture.

Host 1 - Presentation Server

For the Human-Computer Interface [HCI], the IBM Watson received categories and clues as text files via TCP/IP, had a [beautiful avatar] representing a planet with 42 circles streaking across in orbit, and text-to-speech synthesizer to respond in a computerized voice. Your *implementation* will not be this sophisticated. Instead, we will have a simple text-based Query Panel web interface accessible from a browser like Mozilla Firefox.

Host 1 will be your Presentation Server, the connection to your keyboard, video monitor and mouse. Install the "Desktop" or "HPC Head Node" version of Linux. Install [Apache Web Server and Tomcat] to run the Query Panel. Host 1 will also be your "programming" host. Install the [Java SDK] and the [Eclipse IDE for Java Developers]. If you always wanted to learn Java, now is your chance. There are plenty of books on Java if that is not the language you normally write code.

While three little systems doesn't constitute an "Extreme Cloud" environment, you might like to try out the "Extreme Cloud Administration Tool", called [xCat], which was used to manage the many servers in IBM Watson.

Host 2 - Business Logic Server

Host 2 will be driving most of the "thinking". Install the "Server" or "HPC Compute Node" version of Linux. This will be running a server virtualization Hypervisor. I recommend KVM, but you can probably run Xen or VMware instead if you like.

Host 3 - File and Database Server

Host 3 will hold your information sources, indices, and databases. Install the "Server" or "HPC Compute Node" version of Linux. This will be your NFS server, which might come up as a question during the installation process.

Technically, you could run different Linux distributions on different machines. For example, you could run "Ubuntu Desktop" for host 1, "RHEL 6 Server" for host 2, and "SLES 11" for host 3. In general, Red Hat tries to be the best "Server" platform, and Novell tries to make SLES be the best "Guest OS".

My advice is to pick a single distribution and use it for everything, Desktop, Server, and Guest OS. If you are new to Linux, choose Ubuntu. There are plenty of books on Linux in general, and Ubuntu in particular, and Ubuntu has a helpful community of volunteers to answer your questions.

Step 4: Download Information Sources

You will need some documents for *your implementation* to process.

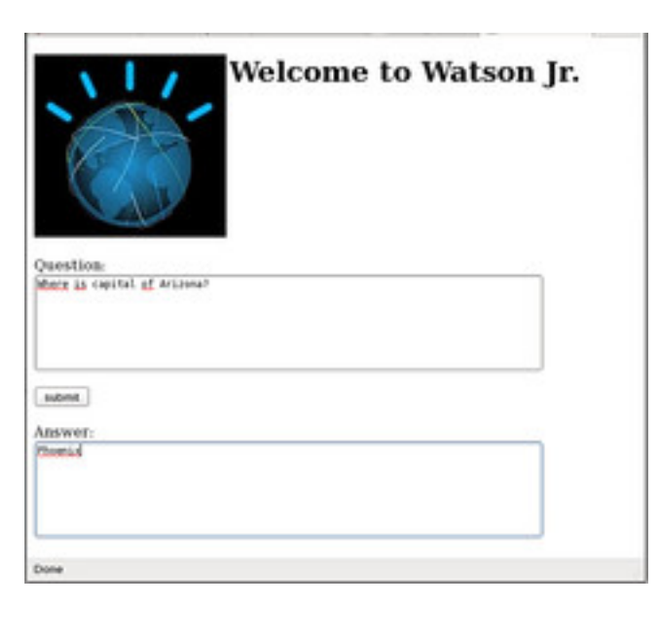
IBM Watson used a modified SONAS to provide a highly-available clustered NFS server. For *this version*, we won't need that level of sophistication. Configure Host 3 as the NFS server, and Hosts 1 and 2 as NFS clients. See the [Linux-NFS-HOWTO] for details. To optimize performance, host 3 will be the "official master copy", but we will use a Linux utility called *rsync* to copy the information sources over to the hosts 1 and 2. This allows the task engines on those hosts to access local disk resources during question-answer processing.

We will also need a relational database. You won't need a high-powered IBM DB2. *Your implementation* can do fine with something like [Apache Derby] which is the open source version of IBM CloudScape from its Informix acquisition. Set up Host 3 as the Derby Network Server, and Hosts 1 and 2 as Derby Network Clients. For more about structured content in relational databases, see my post [IBM Watson - Business Intelligence, Data Retrieval and Text Mining].

Linux includes a utility called *wget* which allows you to download content from the Internet to your system. What documents you decide to download is up to you, based on what types of questions you want answered. For example, if you like Literature, check out the vast resources at [FullBooks.com]. You can automate the download by writing a shell script or program to invoke *wget* to all the places you want to fetch data from. Rename the downloaded files to something unique, as often they are just "index.html". For more on *wget* utility, see [IBM Developerworks].

Step 5: The Query Panel - Parsing the Question

Next, we need to parse the question and have some sense of what is being asked for. For this we will use [OpenNLP] for Natural Language Processing, and [OpenCyc] for the conceptual logic reasoning. See Doug Lenat presenting this 75-minute video [Computers versus Common Sense]. To learn more, see the [CYC 101 Tutorial].



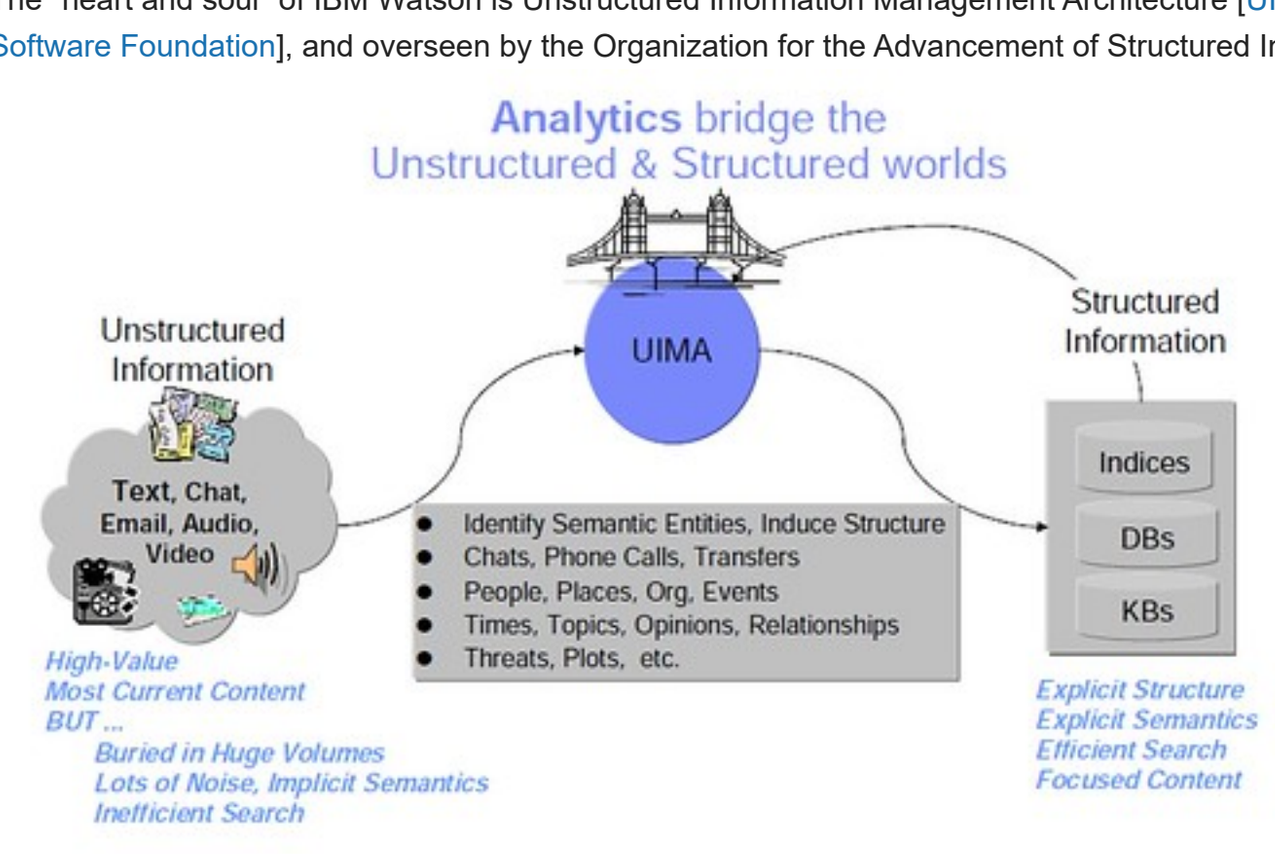
Unlike *Jeopardy!* where Alex Trebek provides the answer and contestants must respond with the correct question, we will do normal Question-and-Answer processing. To keep things simple, we will limit questions to the following formats:

- Who is ...?
- Where is ...?
- When did ... happen?
- What is ...?
- Which ...?

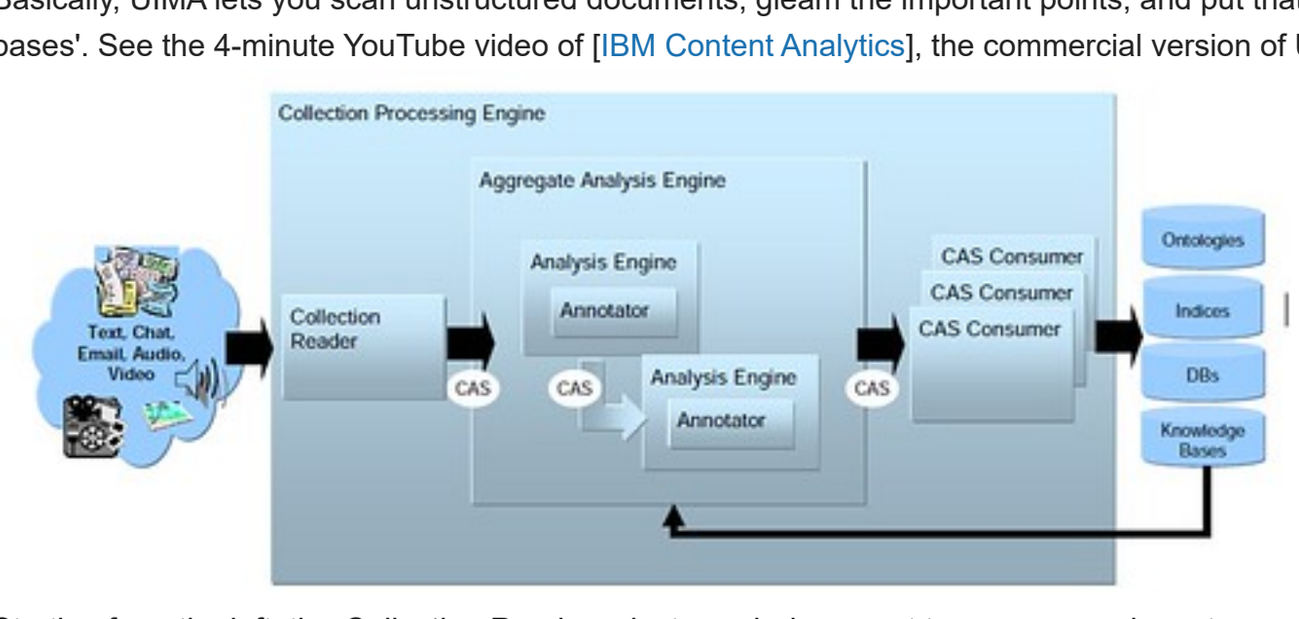
Host 1 will have a simple Query Panel web interface. At the top, a place to enter your question, and a "submit" button, and a place at the bottom for the answer to be shown. When "submit" is pressed, this will pass the question to "main.jsp", the Java servlet program that will start the Question-answering analysis. Limiting the types of questions that can be posed will simplify hypothesis generation, reduce the candidate set and evidence evaluation, allowing the analytics processing to continue in reasonable time.

Step 6: Unstructured Information Management Architecture

The "heart and soul" of IBM Watson is Unstructured Information Management Architecture [UIMA]. IBM developed this, then made it available to the world as open source. It is maintained by the [Apache Software Foundation], and overseen by the Organization for the Advancement of Structured Information Standards [OASIS].



Basically, UIMA lets you scan unstructured documents, gleam the important points, and put that into a database for later retrieval. In the graph above, DBs means 'databases' and KBs means 'knowledge bases'. See the 4-minute YouTube video of [IBM Content Analytics], the commercial version of UIMA.



Starting from the left, the *Collection Reader* selects each document to process, and creates an empty *Common Analysis Structure (CAS)* which serves as a standardized container for information. This CAS is passed to *Analysis Engines* , composed of one or more *Annotators* which analyze the text and fill the CAS with the information found. The CAS are passed to *CAS Consumers* which do something with the information found, such as enter an entry into a database, update an index, or update a vote count.

(**Note:** This point requires, what we in the industry call a small matter of programming, or [SMOP]. If you've always wanted to learn Java programming, XML, and JDBC, you will get to do plenty here.)

If you are not familiar with UIMA, consider this [UIMA Tutorial].

Step 7: Parallel Processing

People have asked me why IBM Watson is so big. Did we really need 2,880 cores of processing power? As a supercomputer, the 80 TeraFLOPs of IBM Watson would place it only in 94th place on the [Top 500 Supercomputers]. While IBM Watson may be the [Smartest Machine on Earth], the most powerful supercomputer at this time is the Tianhe-1A with more than 186,000 cores, capable of 2,566 TeraFLOPs.

To determine how big IBM Watson needed to be, the IBM Research team ran the DeepQA algorithm on a single core. It took 2 hours to answer a single Jeopardy question! Let's look at the performance data:

Element	Number of cores	Time to answer one Jeopardy question
Single core	1	2 hours
Single IBM Power750 server	32	< 4 minutes
Single rack (10 servers)	320	< 30 seconds
IBM Watson (90 servers)	2,880	< 3 seconds

The old adage applies, [many hands make for light work]. The idea is to divide-and-conquer. For example, if you wanted to find a particular street address in the Manhattan phone book, you could dispatch fifty pages to each friend and they could all scan those pages at the same time. This is known as "Parallel Processing" and is how supercomputers are able to work so well. However, not all algorithms lend well to parallel processing, and the phrase [nine women can't have a baby in one month] is often used to remind us of this.

Fortuantely, UIMA is designed for parallel processing. You need to install UIMA-AS for Asynchronous Scale-out processing, an add-on to the base UIMA Java framework, supporting a very flexible scale-out capability based on JMS (Java Messaging Services) and ActiveMQ. We will also need Apache Hadoop, an open source implementation used by Yahoo Search engine. Hadoop has a "MapReduce" engine that allows you to divide the work, dispatch pieces to different "task engines", and the combine the results afterwards.

Host 2 will run Hadoop and drive the MapReduce process. Plan to have three KVM guests on Host 1, four on Host 2, and three on Host 3. That means you have 10 task engines to work with. These task engines can be deployed for *Content Readers*, *Analysis Engines*, and *CAS Consumers*. When all processing is done, the resulting votes will be tabulated and the top answer displayed on the Query Panel on Host 1.

Step 8: Testing

To simplify testing, use a batch processing approach. Rather than entering questions by hand in the Query Panel, generate a long list of questions in a file, and submit for processing. This will allow you to fine-tune the environment, optimize for performance, and validate the answers returned.

There you have it. By the time you get your *implementation* fully operational, you will have learned a lot of useful skills, including Linux administration, Ethernet networking, NFS file system configuration, Java programming, UIMA text mining analysis, and MapReduce parallel processing. Hopefully, you will also gain an appreciation for how difficult it was for the IBM Research team to accomplish what they had for the Grand Challenge on *Jeopardy!* Not surprisingly, IBM Watson is making IBM [as sexy to work for as Apple, Google or Facebook], all of which started their business in a garage or a basement with a system as small as *this version for personal use*.

technorati tags: IBM, Watson, Jeopardy, Challenge, POWER7, EPA, Energy Star, RHEL, SLES, Ubuntu, Linux, UIMA, Hadoop, MapReduce, KVM, DeepQA, Roger Bannister, John Pultorak, Jay Shafer, George Hotz
, Eric Brown

Tags: eric+brown sles epa watson uima rhel ibm ubuntu linux mapreduce power7 jeopardy kvm energy+star deepqa challenge hadoop

Add a Comment | More Actions ▾

Add a Comment | *More Actions* ▾

Comments (12)

Comments (12)

1 | 2 Previous | Next

TonyPearson commented Mar 13 2011

Amazon Web Services provides a complimentary blog post titled "Run SUSE Linux Enterprise Server on Cluster Compute Instances" based on my blog post here:
http://aws.typepad.com/aws/2011/03/run-suse-linux-enterprise-server-on-cluster-compute-instances.html

Comment Permalink

TonyPearson commented Mar 4 2011

Gang Ji,
Yes, you can start with one machine, and expand to other machines later. I suggest you install the "Desktop" version of Linux on your one and only system, and then intall the applicatoins for KVM, Libvirt, xCat, and the rest of the stack. The guest images should be the "Server" or "Guest OS" optimized versions of Linux.
--- Tony

Comment Permalink

Ruggedman commented Mar 4 2011

Hi Tony, Appreciate your info. Watson jr is a very inspiring project and I am going to try it. It is possible to have the system running on single quad-core server with 12GB memory initially as I try it out in small scale before sinking more money for additional servers? Thanks,

Comment Permalink

TonyPearson commented Mar 2 2011

Michelle Castillo writes "Why I Would(n't) Want to Build My Own Watson Jr." here:
http://techland.time.com/2011/02/24/why-i-wouldnt-want-to-build-my-own-watson-jr/

Comment Permalink

TonyPearson commented Mar 2 2011

Paul,
Thanks for the comment. I have had great success with ASUS-based systems.

Chuck,
Glad to hear it. Please keep me informed on your progress!
-- Tony

Comment Permalink

Chuck_Byte_W commented Feb 28 2011

Call me mad, but I'm seriously going to try this.

Comment Permalink

ChavaRod commented Feb 28 2011

Belated congratulations on this great post and the CW article, Tony!

Comment Permalink

TonyPearson commented Feb 27 2011

IBM has created an animated infographic on "What Powers IBM Watson" here:
http://www-943.ibm.com/innovation/us/watson/watson-for-a-smarter-planet/watson-schematic.html

Comment Permalink

bobleah commented Feb 25 2011

Very impressive Tony!

Comment Permalink

polbel commented Feb 25 2011

Thank You Mr Pearson for this impressively dense introduction to Watson and deepqa. From what I can gather, your mind is more open than ibm's publication dept.. For example, [IBM Systems Journal, Vol. 43, No. 3] in this page's link is most directly accessible only through ieee xplore at a cost of 30$ / article, making that month's issue as expensive as this quad-core commodity: http://www.tigerdirect.ca/applications/SearchTools/item-details.asp?EdpNo=7275091&CatId=3508 So I will stick with the hardware and what data is available in the open and run the steps You suggest. I hope in the long run my efforts with Watson will be able to use the 500,000 ebooks in my library to make sure the 5th major life extinction on earth was the last. We have come a long way from Lehnert's doctoral thesis on computer modeling of human question answering and there is still a long way to go before earth's backup, not to mention the solar system's and the galaxy's. My heart-felt gratitude goes to You paul bellehumeur

Comment Permalink

Show 10 | 20 | 50 items Jump to page 1 of 2 Previous | Next

Previous Entry | Main | Next Entry

2018 IBM Systems Techni…
Issuer: IBM
Provided by Acclaim

IBM Redbooks Gold Autho…
Issuer: IBM
Provided by Acclaim

IBM developerWorks…
Issuer: IBM
Provided by Acclaim

2017 IBM Systems Techni…
Issuer: IBM
Provided by Acclaim

Cloud & Cognitive Pattern…
Issuer: IBM
Provided by Acclaim

IBM Expert Network

Related posts

More third-party sto…
Updated Dec 11 2019 0 0

Oncologia na mira do…
Updated Nov 8 2019 0 0

Everything Worth Kno…
Updated Oct 9 2019 0 1

Holiday Readiness: A…
Updated Oct 28 2019 0 0

Personality Insights…
Updated Sep 12 2019 0 0

Links to other blogs...

- Accelerate with ATS
- Alltop - Top Storage Blogs
- Anthony Vandewerdt
- Barry Whyte (IBM)
- Bob Sutor (IBM)
- Brad Johns Consulting
- Chris M. Evans
- Greg Schulz
- Hu Yoshida (HDS)
- Jim Kelly (IBM)
- Jon Toigo - DrunkenData
- Kirby Wadsworth (F5)
- Raj Sharma, IBM Storage and Te...
- Richard Swain (IBM)
- Roger Leuthy, Storage CH Blog
- Seb's SANblog
- Stephen Foskett, Pack Rat
- Storageneers
- Storagezilla
- Technology Blogs

Archive

Disclaimer