Defining Data-intensive computing

B. Ramamurthy

Topics for Discussion

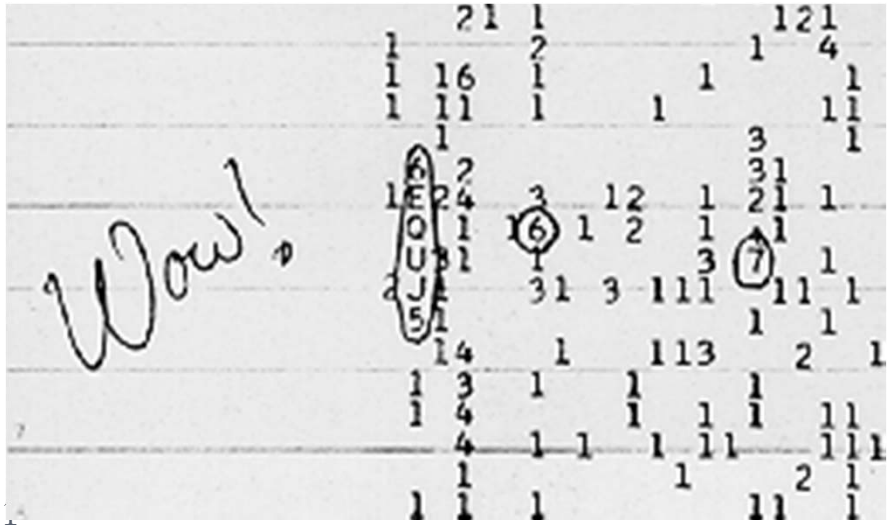
- Intelligence and scale of data
- Examples of data-intensive applications
- Basic elements of data-intensive applications
- Designing and building data-intensive applications
- Enabling technologies: Internet, Web, XML, web services creation and consumption
- Large-scale computing constraints and solutions
- Fourth paradigm: Data collection, curation and analysis
- Project 1: Web-services based three-tier application: stand alone and deployed on the cloud

Intelligence and Scale of Data

- Intelligence is a set of discoveries made by federating/processing information collected from diverse sources.
- Information is a cleansed form of raw data.
- For statistically significant information we need reasonable amount of data.
- For gathering good intelligence we need large amount of information.
- As the Fourth Paradigm (FP) book points out enormous amount of data is generated by the millions of experiments and applications.
- Thus intelligence applications are invariably data-heavy, data-driven and data-intensive.
- Very often the data is gathered from the web (public or private, covert or overt).

INTELLIGENCE

- Search for Extra Terrestrial Intelligence (seti@home project)
- http://setiathome.berkeley.edu/
- The Wow signal http://www.bigear.org/wow.htm



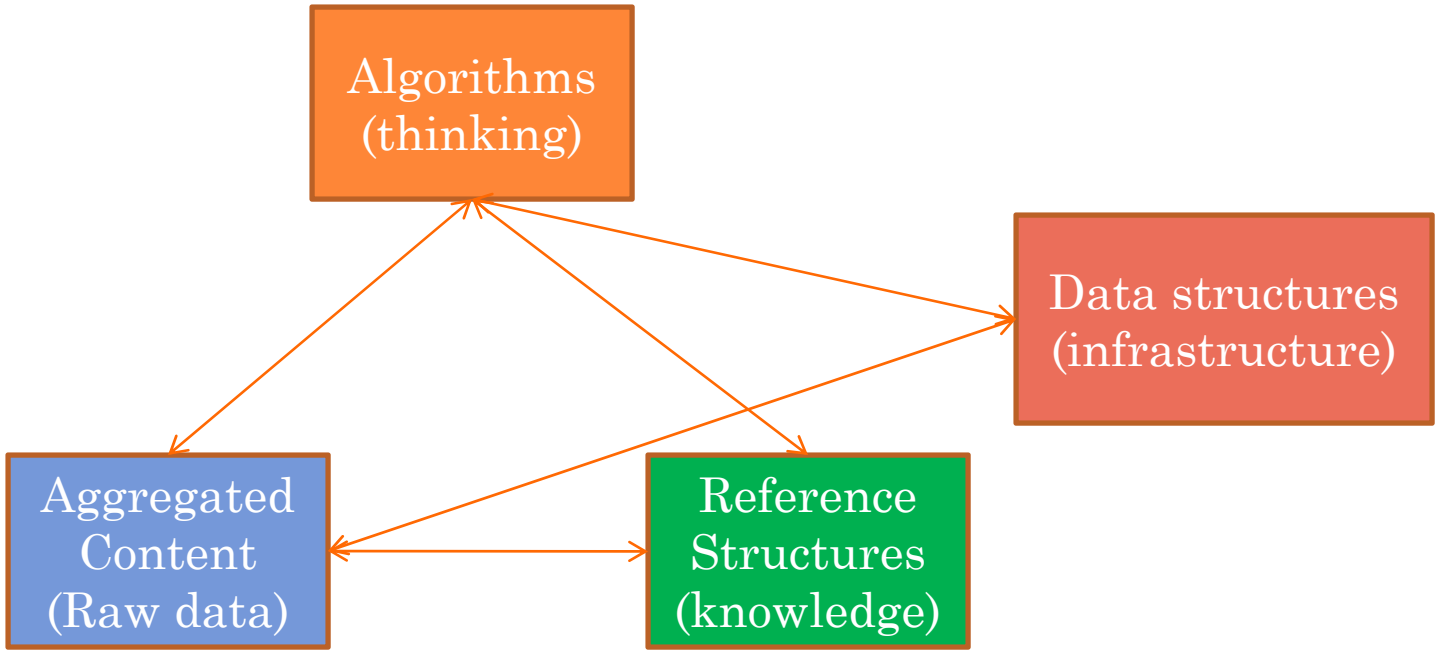
CHARACTERISTICS OF INTELLIGENT APPLICATIONS

- Google search: How is different from regular search in existence before it?
  - It took advantage of the fact the hyperlinks within web pages form an underlying structure that can be mined to determine the importance of various pages.
- Restaurant and Menu suggestions: instead of "Where would you like to go?" "Would you like to go to CityGrille"?
  - Learning capacity from previous data of habits, profiles, and other information gathered over time.
- Collaborative and interconnected world inference capable: facebook friend suggestion
- Large scale data requiring indexing
- …

EXAMPLES OF DATA-INTENSIVE APPLICATIONS

- Search engines
- Recommendation systems:
  - CineMatch of Netflix Inc. movie recommendations
  - Amazon.com: book/product recommendations
- Biological systems: high throughput sequences (HTS)
  - Analysis: disease-gene match
  - Query/search for gene sequences
- Space exploration
- Financial analysis

Data-intensive application characteristics



Basic Elements

- **Aggregated content**: large amount of data pertinent to the specific application; each piece of information is typically connected to many other pieces. Ex:
- **Reference structures**: Structures that provide one or more structural and semantic interpretations of the content. Reference structure about specific domain of knowledge come in three flavors: dictionaries, knowledge bases, and ontologies
- **Algorithms**: modules that allows the application to harness the information which is hidden in the data. Applied on aggregated content and some times require reference structure Ex: MapReduce
- **Data Structures**: newer data structures to leverage the scale and the WORM characteristics; ex: MS Azure, Apache Hadoop, Google BigTable

More intelligent data-intensive applications

- Social networking sites
- Mashups : applications that draw upon content retrieved from external sources to create entirely new innovative services.
  - http://www.ibm.com/developerworks/spaces/mashups
- Portals
- Wikis: content aggregators; linked data; excellent data and fertile ground for applying concepts discussed in the text
- Media-sharing sites
- Online gaming
- Biological analysis
- Space exploration

BUILDING YOUR OWN APPLICATION

- Determine your functionality: UML model use case diagram is a very nice tool to use at this stage
- Determine the source of your internal and external data
- Examine the data and its utilization in the application
- Methods for enhancing the application
  - Web data
  - Crawling and screen scraping
  - RSS feeds
  - RESTful services
  - Web services

FUNCTIONALITY

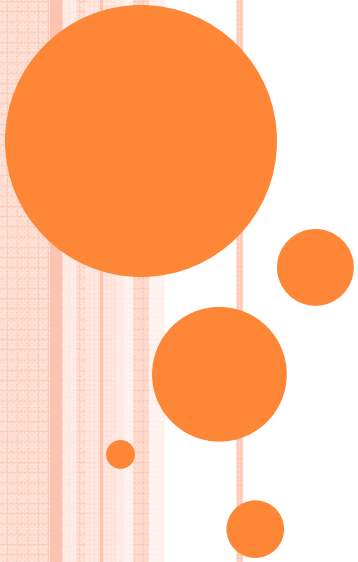
- Use case diagram is a good tool to discover/define the functionality of your applications
- Questions to ask:
  - What are the main functions?
  - What kind of data? Structured? Unstructured?
  - Where will be stored?
  - Will it be shared?
  - What are the sources of data?
  - Does it deal with geographic locations (maps)?
  - Does it share content?
  - Does it have search?
  - Any automatic decisions to be made based on rules?
  - What is the security model?

ACQUIRING THE DATA

- Example: Get the houses available from Craigslist and post it on Google maps
- Enabling technologies for acquiring data:
- Crawler: spiders, start with a URL and visit the links in the URL collecting data, depth of crawling is parameter
- Screen scrappers: extract information that is contained in html pages.
- Biological sciences: High throughput sequencers
- Web services: APIs that facilitate the communication between applications. Organizations make available the relevant information as services
  - REST and SOAP are two underlying pipes for WS

ALGORITHMS

- Machine learning is the capability of the software system to generalize based on past experience and the use of these generalization to provide answers to questions related old, new and future data.
- Data mining
- Soft computing
- We also need algorithms that are specially designed for the emerging storage models and data characteristics.



WEB SERVICES

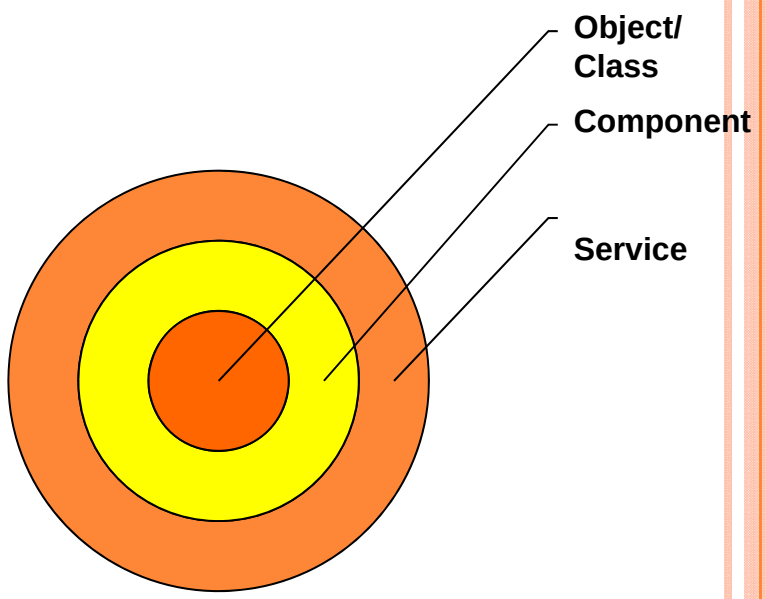
B. Ramamurthy

Topics

- What is a web service?
- From OO to WS
- WS and the cloud
- WS code

EVOLUTION OF THE SERVICE CONCEPT

- A service is a meaningful activity that a computer program performs on request of another computer program.
- Technical definition: A service a remotely accessible, self-contained application module.
- From IBM,



Class, Component and Service

- Class is a core concept is object-oriented architectures. An object is instantiated form a class.
  - Focus on client side, single address space programs.
- Then came the component/container concept to improve scalability and deployability. Ex: EJBs.
  - Focus on server side business objects and separation of resources from code.
- Service came into use when publishing, discoverability, on-demand operation among interacting enterprise became necessity.
  - Focus of enterprise level activities, contracts, negotiations, reservations, audits, etc.

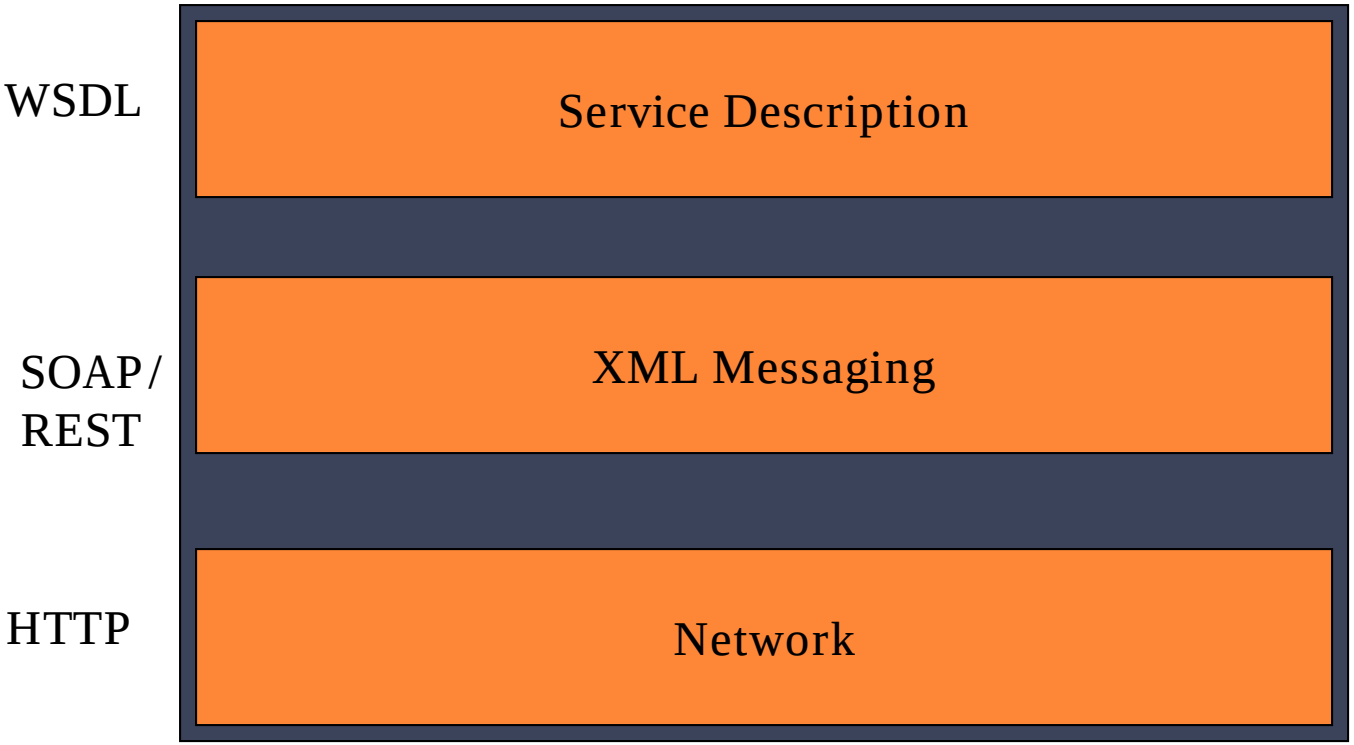
Object-oriented programming

- Object-oriented programming
  - Encapsulation of data and function in a class, instances of a class is called an object
  - Objects communicate through messages (invoking methods)
  - Class represents a type from which another type can be derived resulting inheritance hierarchy.
  - Problem: level of abstraction and granularity exposed is fine to enable reuse.
  - Data and functions are tightly coupled.
  - The concept of interface
- Service-orientation assumes that data and functionality are separated.

Web Services and the Cloud

- *Web Service* is a technology that allows for applications to communicate with each other in a standard format.
- A *Web Service* exposes an interface that can be accessed through XML messaging.
- A Web service uses XML based protocol to describe an operation or the data exchange with another web service. Ex: SOAP
- A group of web services collaborating accomplish the tasks of an application. The architecture of such an application is called Service-Oriented Architecture (SOA).
- Web service is an important enabling technology of cloud computing: software-as-a-service (SaaS), platform-as-a-service(PaaS), infrastructure-as-a-service (IaaS)

WS Interoperability Infrastructure



Do you see any platform or language dependencies here?

XML TO SOAP

- Simple xml can facilitate sending message to receive information.
- The message could be operations to be performed on objects.
- Simple Object Access Protocol (SOAP) or REST

SOAP Request

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productId>827635</productId>
    </getProductDetails>
  </soap:Body>
 </soap:Envelope>
```


SOAP Reply

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productId>827635</productId>
        <description>3-Piece luggage set.  Black Polyester.</description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP→Web Services (WS)

- Read this paper:

http://www.w3.org/DesignIssues/WebServices.html

- Lets look at some WScode:

REST-BASED WEB SERVICES

- Representational State Transfer (REST)
- Ph.D. thesis by Roy Fielding, who was the chairman of the Apache Software Foundation (not anymore)
- We will use REST-based WS for our projects.
- We will discuss basics of REST next classes.
- 8/30/2011: Reading Assignment: Front material, and up to page 44 of the Fourth Paradigm text.
- Review your Java skills, install your favorite IDE (Eclipse, netbeans etc.), J2EE (helios), any design tool.