UpStage Team

Investigation of  
Data Storage

UpStage 2015 S2

# Researching Criteria for Storage of the Persistent Data

* Performance
  + How fast it is
  + How it suits to a web application
  + How much it requires the system resources

# XML

* Self-describing:  
  It can be used to define the common traits. XML contains not only the data, but also the necessary metadata.
* The data is simply written in such a way to allow machines to read it and make it accessible to a database.
* Hierarchical:  
  It contains the information about the relationship of data items to each other in the form of the hierarchy. It does not require any primary key or foreign key relationships.

Perspective for XML

* “If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes.

With XML, data can be stored in separate XML files. This way you can concentrate on using HTML/CSS for display and layout, and be sure that changes in the underlying data will not require any changes to the HTML.

With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.”

* “One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet.

Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.”

* “Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost.

XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.”

Possible negatives

* It can have a lot of overhead. It is not great for massive files.

# Relational Database

Database is the collection of schemes, tables, queries, etc. Well-known database management systems include MySQL, Microsoft SQL Server, Oracle and so on.

Pros:

* Reduced data redundancy
* Greater data integrity and independence from applications programs
* Improved data security

Cons:

* Complex, difficult and time consuming to design
* Damage to database affects virtually all applications programs

# Plain Text file

The main *advantage* of plain text formats is their simplicity: we do not require complex software to create or view a text file and we do not need esoteric skills beyond being able to type on a keyboard, which means that it is easy for people to view and modify the data. The simplicity of plain text formats means that virtually all software packages can read and write text files and plain text files are portable across different computer platforms.

The main *disadvantage* of plain text formats is also their simplicity. The basic conceptual structure of rows of values can be very inefficient and inappropriate for data sets with any sort of complex structure. The low-level format of storing everything as characters, with one byte per character, can also be very inefficient in terms of the amount of computer memory required.

# Rationale for choosing XML

Any performance comparison of data storages depends heavily on the data that will be used. UpStage does not have to handle large volumes of data within the system. This considered, XML is the best option to store the persistent data.

Relational databases are better for handling large volumes of data within a system while XML is not.

Also, since XML does a good job of delivering self-describing data feeds it has become a key standard in Service Oriented Architectures and Web Services.