BITIRME TEZ SUNUMU

DATA FILTERING AND TEXT CLEANING

Log Nedir?

- Log bir operasyonun kalıcı kayıt altına alınmasıdır.
- Runtime anında uygulamaların durumunun saklanmasıdır.
- Uygulamanın durumunu yansıtır.
- Developlent, test ve debug süreçlerinde kullanılır.

Log Yapısı

- Loglama yapısı sistematik, kontrol edilebilir ve okunabilir olmalıdır.
- Log'lanacak olan bilgiler doğru belirlenmelidir.
- Log strajesi belirlenmelidir.
- Keyfi log'dan uzak durulmalıdır.
- Kötü yazılmış log, okunamayacağı gibi uygulama performansını da düşürür.
- Amaç; sistemin anlık durumunu yansıtmaktır.

Loglama Avantajı

- Kolay bakım.
- Hızlı debug.
- Geri bildirim/Analiz.
- Maliyet ve zaman kazancı.
- Geçmiş.

Loglama Dezavantajı

- Kötü yazılmış loglama yapısı uygulama performansını düşürür.
- Verimli/Yararlı olmayan log kafa karışıklığına sebep olur.
- Log yazmak için ek kod.
- Kötü log yazılmışsa zaman ve maliyet kaybı.

Log4J Logging Framework

- Open Source ve Apache Licence.
- Thread Safe.
- Different Log Level.
- Java ile yazılmıştır.
- Flexible.
- Katmanlı mimari.
- Top Layer, Middle Layer, Bottom Layer.
- Diğer Programlama dillerine import edilebilir(C,C++,C#)

Log4J Logging Framework

- Core Object and Support Object.
- Core object is mandatory, Support Object optional.
- Core Object;
- Logger, Loglama bilgilerini alır. Top Layer.
- Appender, Log bilgilerini ilgili hedefe/dizine yollar.
- Layout, Log bilgisinin formatlanmasını sağlar.(Human Readable)

Log4J Logging Framework

- Support Object;
- Level, seven levels of logging.

OFF, DEBUG, INFO, ERROR, WARN, FATAL, and ALL.

- Filter, before publis log info analyze logging.
- ObjectRenderer, string represantation of diffrent object.
- LogManager, initalizing and managing.

Log4J Logging Framework

- Appender Object;
- File,Console, DB, JMS, JPA, SMTP, Socket, SysLog appender.
- Multiple Logging at same time.
- Log4j configuration;

  log4j.xml and log4j.properties

Log4J Logging Framework

- Logging Layout;
- Csv, Json, PatternLayout, XmlLayout.
- Different destination, different layout format.
- Csv Log;

```
logger.info("Ignored", value1, value2, value3);
// value1, value2, value3
```

- Json Log;

```
 {
        "logger":"com.wora.hadoop.Analys",
        "timestamp":"1376681196470",
         "level":"INFO",
        "thread":"main",
        "message":"Received text data"
}
```

Log Analys

- Log içerdiği veri itibariyle önemlidir.
- Log analizi/incelenmesi çoğu zaman gereklidir.
- Kötü yazılmış log = maliyet + zaman kaybı.
- Kötü yazılmış log = performans kaybı.
- Sektörde log framework'leri bolca kullanılır.
- Toplam 3 uygulama;
  1 - CreateLog => İşlenecek olan Log dosyasını oluşturmaktan sorumludur.
  2 - LogAnalys => Log dosyalarını verilen template'e göre işlemekten sorumludur.
  3 - LogAnalysMonitor => LogAnalys uygulamasından oluşan output'u göstermekten sorumludur.

CreateLog

- Log dosyası oluşturma.
- İşlenecek olan veri'nin oluşturulması.
  <STX><SUB>2 Jan 2016 06:36:21<SUB>contextInitialized<SUB>Log4j and Scheduler job initialized<SUB>167<SUB><ETX>
- İşlenecek olan mesajları özel hale getirmek.
  Mesaj başına, ortasına ve sonuna özel karekterler eklemek.
  Bkz: <STX> : Start Of Text , <ETX> : End Of Text
- Variable declaration : <STX><ETB>02 Jan 2016 06:36:21<ETB>createDealerEmployee<ETB>212<ETB><ETX>
  Mesela burada ETB ile ayrılmış, tarih, variable adı ve değişken boyutu değerleri kullanılmıştır.
- Context İnitialize : <STX><SUB>2 Jan 2016 06:36:21<SUB>contextInitialized<SUB>Log4j and Scheduler job initialized<SUB>167<SUB><ETX>

CreateLog

- Context Destroyed : <STX><ESC>02 Jan 2016 06:36:2<ESC>contextDestroyed<ESC>Cleaning log4j and scheduler job<ESC>23<ESC><ETX>
- Loop declaration : <STX><CAN>02 Jan 2016 06:36:22<CAN>getChangeDealerEmployee<CAN>70000000 <CAN>19<CAN><ETX>

LogAnalys

- Log Processor katmanı ilk olarak template'de tanımlanmış ise begin ve end karekterlerinin olup olmadığını kontrol eder
  Örn: <STX> ve <ETX> arasındaki mesaj kullanılmak istensin.
- Eğer begin ve end karekterleri tanımlı ve log datasında bulunuyor ise bu iki karekter arasındaki alt string dizisi elde edilir.
- Log Processor input : 2016-01-02 18:36:21 DEBUG VarietyOperationDaoImpl - LOOP : <STX><CAN>02 Jan 2016 06:36:22<CAN>getChangeDealerEmployee<CAN>70000000<CAN>19<CAN><ETX>
- Log Processor output : <CAN>02 Jan 2016 06:36:22<CAN>getChangeDealerEmployee<CAN>70000000<CAN>19<CAN>

LogAnalys

- Log Processor'den gelen yapılı veri Validation'a tabi tutulur.
- Validation Processor input : <STX><CAN>02 Jan 2016 06:36:22<CAN>getChangeDealerEmployee<CAN>70000000<CAN>19<CAN><ETX>
- Validation Processor aşaması;

  02 Jan 2016 06:36:22 -Tipi date -> pattern'a uyuyor mu?
  Örneğin; Pattern'da "greater than %01 Jan 2016 06:36:22%" ifadesi var ise valid bir data olarak değerlendirilecektir.
  19 - Tipi numeric -> pattern'da "less than %20%" ifadesi var, uygun bir datatır.

Satır üzerindeki tüm alanlar validasyondan geçer ise processor tarafından memory database'e eklenmek üzere insert scripti oluşturulur ve template'e atanan adaptor'lere bu mesaj iletilir.

Log Monitor

- Log Processor output >>> Log Monitoring App input
- Java Web ile MVC pattern ile yazılmıştır.
  Model : Log Line
  Controller : Log Action
  View : Jsp

Log Monitor

- Monitor uygulaması, LogAnalys uygulamasının işlemini bitirmesi sonucu internal db'ye yazılan verilerin sunulmasından sorumludur.
- Uygulama bir Java web uygulamasıdır ve Tomcat üzerinde çalışmaktadır.
- Ayrıca, LogAnalys uygulaması içerisinde yer alan Embedded Tomcat modülü ile birlikte iki uygulama ayrı ayrı çalışacağı gibi birleştirilerek de çalışabilmektedir.
- Embed Tomcat üzerinde, properties dosyası üzerinden ayarlamar yapılır ve uygulama WAR olarak export edilerek otomatik olarak deployment yapılarak istenilen port'dan uygulama istenilen context name ile ayağa kalkarak hizmet vermeye başlamaktadır.

Log Monitor

- Monitor ile kullanıcı, tanımladığı template'lerin bilgisini görebilmektedir.
- Tanımlanan template'lere hangi verilerin, ne kadar verinin uyduğunu görebilecektir. Örneğin; Performans ölçütü olarak server startup süresinin 100ms'den büyük olunan durumlar irdelenebilecektir.
- Toplam template sayısı, bu template'lere uyan veriler, export edilen veriler, export durumu vb gibi tüm detaylar ekran üzerinden izlenebilecektir anlık olarak.