Комплексный анализ данных киберфизических систем

Д. П. Плахотников[1], Е. Е. Котова[2]
Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
[1]dimapl21@yandex.ru, [2]apu_kotova@mail.ru

***Аннотация*. В процессе работы киберфизических систем образуется достаточно большой и разнородный набор данных – данные о коммерческом учёте, данные от датчиков, данные о состоянии оборудования и т. д. Для обработки таких данных необходимо разработать правила обработки «сырой» информации, соотнести информацию из разных систем, найти закономерности и проанализировать полученную информацию. Для данной задачи можно использовать системы класса аналитических платформ, иначе называемые средствами интеллектуального анализа данных (Business Intelligence).**

Ключевые слова: киберфизические системы; анализ данных; аналитические платформы; бизнес-аналитика

I. Введение

С появлением концепции киберфизических систем и возникновением действующих систем подобного рода связывают переход к четвертой промышленной революции. Она заключается к массовому внедрению киберфизических систем в производство и обслуживание человека. Особенностью четвертой промышленной революции является конвергенция транспортных технологий, новых информационных технологий и технологий искусственного интеллекта. Происходит интеллектуализация промышленности и промышленных изделий, появляется полностью автономный транспорт и транспортные инфраструктуры [1].

Одной из особенностей киберфизических систем является генерация системами такого рода большого объёма разнородных данных. Эти данные позволяют процессам в физических системах производить влияние на вычисления. Метод и способы обработки таких данных сейчас являются ключевой проблематикой области киберфизических систем. Анализ таких данных должен учитывать, как и разные методы получения информации – ручной ввод или сырые данные, так и возможные ошибки как технического рода, так и антропогенного.

II. Что представляет собой киберфизическая система?

Киберфизическая система – это система, интегрирующая в себе оборудование, датчики, вычислительные ресурсы и информационные системы, на протяжении всей цепочки создания стоимости, как правило, выходящей за рамки одного предприятия или бизнеса. В идеальной киберфизической системе, она сама настраивается на выполнение новых задач, сама себя обслуживает, анализирует и сама изменяет технологический процесс в зависимости от поставленных им задач.

Киберфизические системы бывают разных масштабов – это может быть, как беспилотный летательный аппарат, так и система умного города, включающая в себя десятки, а иногда и сотни тысяч различных устройств.

В качестве киберфизической системы будет рассмотрена сеть газозаправочных станций и газозаправочные колонки на них. Сеть газозаправочных станций представляет с собой разрозненные станции, которые расположены по территории страны. На каждой из станций располагается несколько колонок. На каждой из колонок расположены разнообразные датчики контроля.

Эти станции через виртуальную частную сеть (VPN) подключены к главному офису. VPN позволяет защищенно обмениваться информацией, используя открытые каналы связи (такие как интернет) [2]. Схематично это изображено на рис. 1.

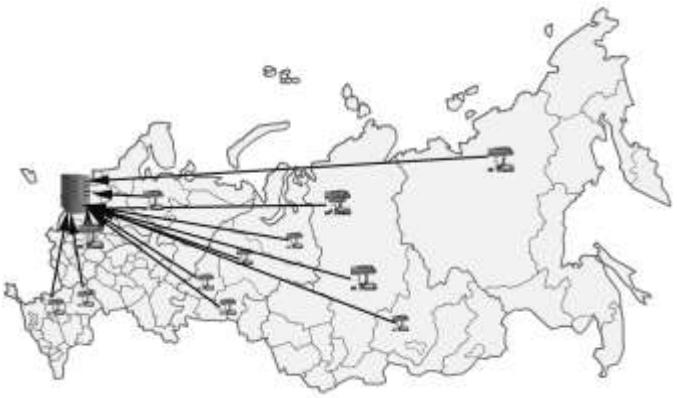

Рис. 1. Схематичное представление сети газозаправочных станций

В главном офисе настроена аналитическая система и рабочее место аналитика. Основная информация со станции передаётся в системную базу данных, но полный набор информации (данные датчиков, действия оператора, отчеты работы программы) остается на рабочем месте оператора. Это связанно с тем, что в отдалённых местах

имеется слабый пропускной канал (например, используется мобильная связь для передачи данных). Данная схема подключения станций к главному офису показана на рис. 2.

Рис. 2. Подключение станции к главному офису

В свою очередь, колонка, через которую отпускается газомоторное топливо, тоже представляет собой киберфизическую систему, поскольку к ней подключены различные датчики – датчик плотности, температуры, утечки и объема. Схематично это представлено на рис. 3.

Рис. 3. Датчики колонки

Чтобы подключить датчики к компьютеру оператора используется специальный контроллер. Такой контроллер имеет аналоговые и цифровые входы и порт Ethernet. Через него осуществляется контроль оборудования [3].

III. Проектирование аналитической системы

Аналитическая система может располагаться на одном из нескольких виртуальных серверов. Один из них действует как центральный узел и служит точкой управления всей системой, а иные сервера выполняют определенные роли – хранение данных, обработка данных, отображение приложений другим пользователям и так далее.

Данная архитектура представлена на рис. 4. Аналитика построена на данных, которые поступают на сервер из различных источников. Исходные данные могут быть загружены на сервер как по расписанию, так и по запросу администратора сервера.

Рис. 4. Архитектура аналитической системы

Источниками данных для аналитической системы, помимо системной базы данных (представленной Oracle SQL), так же являются плоские файлы (формат * .xlsx, * .csv, *.txt), реляционные базы данных рабочих мест оператора станции (MS SQL) и репозитории доменных служб MS Active Directory (MS AD).

В плоских файлах формата Excel (*.xlsx) хранится дополнительная информация, которая не представлена в базе данных, такая как ФИО начальника станции, время работы станции, координаты станции и иное.

В текстовом формате табличных данных (*.csv) представлена информация о времени загрузки данных, для инкрементальной (пошаговой) загрузки данных. Хранение времени загрузки позволяет уменьшить нагрузку на базу и увеличить скорость обработки новой информации.

В текстовом формате данных (*.txt) хранятся необработанные данные датчиков со счётчиков в различном формате. Формат обычно строго задан, что позволяет использовать данные для парсинга (процесса сопоставления различных лексем формального или естественного языка с его определённой формальной грамматикой) [5].

В реляционной базе данных MS SQL сохраняются отчеты работы программного обеспечения на станции.

В репозитории доменных служб MS AD хранится информациях о всех пользователях и компьютерах в локальной сети.

Загрузка и обработка данных происходит в выделенных процессах ETL / ETR, которые выполняются до этапа анализа данных, загруженных на сервер аналитики [4]. Данные считываются из различных источников, затем выборочно обрабатываются и сохраняются в аналитической системе. На основе этих полученных данных разрабатывается аналитическое приложение позволяющие сопоставлять данные из различных систем.

IV. Практическая часть

В процессе своей работы на газозаправочной станции образуется много данных. В качестве данных для анализа были выбраны данные отпуска по счётчику. Была поставлена задача сверки отпуска через системную базу данных и счётчика газораспределительной колонки.

A. Получение информации о станциях

Для того, чтобы собрать информацию со станций необходимо их вначале определить их расположение в локальной сети.

Все компьютеры в локальной сети хранятся в Active Directory поэтому чтобы получить информацию необходимо воспользоваться следующей командой: *powershell.exe /C "Get-ADComputer –LDAPFilter '(Name=*STATION*)' – Properties "DistinguishedName", "DNSHostName", "Enabled", "Name", "ObjectGUID", "SamAccountName" | Export-Csv -NoTypeInformation GNStation.csv -Encoding Unicode"*

В ней, с помощью фильтра (LDAPFilter) удалось отсеять иные компьютеры в сети, а также ускорить получение информации с помощью необходимых характеристик (*Properties*). Так, например, характеристика «Enabled» (включена) позволяет понять работает ли станция на данный момент.

После получения информации о станциях был сформирован скрипт параллельного подключения к станции для забора информации.

B. Распаковка данных со станции

После получения списка станций предстоит получить информацию с датчиков. Она хранится в архивах по определённому пути. Каждый архив содержит информацию за неопределённый период времени. Был написан скрипт формирования списка файлов, представленный на рис. 5.

```
For tmp=0 to $(vNows)   //перебор станций
Let vNewFolderPath=Peek('Folder', $(tmp), 'Folders');   //путь до папки
Let vNewFolder=SubField('$(vNewFolderPath)','/',-1);

Let _Last_FileModify_max=ApplyMap('Map_UnpackPeriod','$(vNewFolder)',0);
Let Last_FileModify_max=Alt(_Last_FileModify_max,0);
Let _FileModify_max=Last_FileModify_max;

FOR Each vFile in FileList('$(vNewFolderPath)/*.7z')
Let vFileFullName=SubField('$(vFile)','/',-1);
Let vFileModify=Timestamp(FileTime('$(vFile)'));

If (vFileModify>Last_FileModify_max) then
Let _FileModify_max=Timestamp(RangeMax('$(_FileModify_max)','$(vFileModify)'));
UNPACK:
LOAD
    Replace(Replace('$(vFile)','$(vRootLogs)','$(vRootLogsFull)'),'/','\') as FilePath,
    Replace(Replace('$(vNewFolderPath)','$(vRootLogs)','$(vUnpackPath)'),'/','\') as UnpackPath,
    '$(vFileModify)' as FileModify
AutoGenerate 1;
End If;
NEXT vFile

UnpackPeriod:
Load
    '$(vNewFolder)' as NewFolder,
    '$(_FileModify_max)' as MaxFileModify
AutoGenerate(1);

Next tmp
```

Рис. 5. Скрипт формирования списка файлов для распаковки

С помощью таблицы UnpackPeriod запоминается дата последнего распакованного файла, а с помощью таблицы UNPACK формируется список файлов для распаковки.

Распаковка происходит по сформированную списку с помощью заданного архиватора $(vArchivator) в путь $(vUnpackPath). $(vMask) позволяет указать только нужные файлы, а $(vKey) ключ для перезаписи. Скрипт распаковки представлен на рис. 6.

```
Let vUnpackRows=Alt(NoOfRows('UNPACK')-1,0);

If ($(vUnpackRows)>0) then
For tmp=0 to $(vUnpackRows)   //перебираем
Let vFilePath=Peek('FilePath', $(tmp), 'UNPACK');
Let vUnpackPath=Peek('UnpackPath', $(tmp), 'UNPACK');
    Execute $(vArchivator) e $(vFilePath) -o$(vUnpackPath) $(vMask) $(vKey);
Next tmp
End If;
STORE UnpackPeriod into [$(vCsvPath)_UnpackPeriod.csv] (txt);
End Sub;
```

Рис. 6. Скрипт распаковки информации с датчиков

После получения данных со станций их необходимо их изучить и обработать. Пример полученных исходных данных с контроллеров представлен на рис. 7.

Рис. 7. Исходные данные с контроллеров счетчиков

C. Обработка данных с контроллеров

В ходе изучения исходных данных для анализа были выбраны события ReadPumpCounters – события получения состояния счётчика колонки. Скрипт обработки таких данных представлен на рис. 8.

```
EXIST:
Load * inline [@4,
ReadPumpCounters
];

[_Dons]:
LOAD
    RowNo() as RowNo,
    '$(vStationId)' as StationId,
    Date#([@1]) as Date,
    Time(Time#(Trim([@2]),'hh:mm:ss:fff')) as Time,
    Timestamp(Date#([@1])&' '&Time(Time#(Trim([@2]),'hh:mm:ss:fff'),'h:mm:ss.fff')) as LogTS,
    [@6] as PumpNum,
    num(SubField([@11],'Counter=',2),'0,000') as Counter
FROM [$(vFile)]
(txt, codepage is 28592, no labels, delimiter is spaces, msq)
where Exists(@4);
```

Рис. 8. Скрипт обработки данных со счетчиков

После этого были получены сырые данные. Выяснилось, что счётчики периодически сбрасывают показания в 0, ведут свой отчёт до 1000000, после чего начинают с начала, появляются «фантомные» значения (краткосрочное изменение на некорректные значения). Для фильтрации таких данных был разработан скрипт фильтрации данных, представленный на рис. 9.

В первую очередь он фильтрует пустые данные (0) и фантомные изменения. Затем заполняются предыдущие значения, и происходит очищение данных от различного рода ошибок. Данные действия позволили добиться минимальных расхождений от реальной картины.

```
NoConcatenate
_Doms_filter:
Load
*
Resident _Doms_sort
Where Counter<>Previous(Counter) and Counter<>Previous(Previous(Counter))
and Counter<>0;
Drop Table _Doms_sort;

NoConcatenate
_Doms_prev:
Load
*,
Previous(PumpNum) as Prev_PumpNum,
Previous(Counter) as Prev_Counter
Resident _Doms_filter;
Drop Table _Doms_filter;

NoConcatenate
_Doms_Volume:
Load
*,
if(Counter-Prev_Counter<-99000 and Counter-Prev_Counter>=-100000,
Round(Counter-Prev_Counter+100000,0.001)*10,
if(Counter<Prev_Counter and Counter<999,Round(Counter,0.001)*10,
if(fAbs((Counter+1)/(Prev_Counter+1))<999,//было 9999
if(Counter>Prev_Counter,//избавление от отрицательных значений
Round(Counter-Prev_Counter,0.001)*10)))) as Volume
Resident _Doms_prev
Where PumpNum=Prev_PumpNum and Counter<>Previous(Prev_Counter);
Drop Table _Doms_prev;
```

Рис. 9. Скрипт фильтрации данных

D. *Получение данных из главного офиса*

Для получения данных из главного офиса было выполнено подключение к системной базе данных, загрузка таблицы транзакций и таблицы состояний счётчиков на каждую смену. Обработка данных не потребовалась, поскольку данные были подготовлены самой базой данных. Необходимо было лишь сопоставить формат на счётчике и в базе данных с помощью скрипта, представленного на рис. 10.

```
if(STARTREALCOUNTER>${vLimit},Round(STARTREALCOUNTER/10000,0.01),Round(STARTREALCOUNTER,0.01))
as StartRealCounter,
if(ENDREALCOUNTER>${vLimit},Round(ENDREALCOUNTER/10000,0.01),
if((ENDREALCOUNTER/STARTREALCOUNTER)>10 and (ENDREALCOUNTER/STARTREALCOUNTER)<15
and fAbs(ENDREALCOUNTER-STARTREALCOUNTER)>9999,Round(ENDREALCOUNTER/10,0.01),
Round(ENDREALCOUNTER,0.01))) as EndRealCounter,
```

Рис. 10.Скрипт обработки данных о счётчиках в базе данных

E. *Сопоставление данных*

Были объединены данные со счётчиков и транзакций, если отпуск был в рамках часа (+/- час) на данной колонке. Это было сделано с помощью следующих формул корректировки времени начала отпуска (*StartFillingDT*) и конца отпуска (*EndFillingDT*):

If(IsNull(StartFillingDT),Timestamp(TransTM-(63.5/1440)), Timestamp(StartFillingDT-(60/1440))) as StartFillingDT,

If(IsNull(EndFillingDT),Timestamp(TransTM+(62.5/1440)), Timestamp(EndFillingDT+(62/1440))) as EndFillingDT.

Далее, необходимо было отфильтровать полученные данные, так как в течение двух часов практически всегда было более одного отпуска. После произведенного анализа данных транзакций с системной базы данных и данных счётчиков выяснилось, что медианная разница между событием в базе и событии на счётчике составляет 28,8 секунды. Была добавлена следующая формула:

*Alt(fAbs(TransTM-LogTS-0.48/1440)*1440,999) as DifTS*

Она демонстрирует разницу между счётчиком и базой, сдвинутую на медианную разницу. Так же были добавлены два флага – флаг суммы (FlagSum) и флаг ошибки (ErrorFlag):

if(IsNull(OilSupplied),3,0)+if(fAbs(OilSupplied-Volume)>0.15, 1,0)+if(fAbs(OilSupplied-Volume)>0,1,0) as FlagSum,

if(Round(Volume/10,0.01)=Round(Counter,0.01) and (IsNull(OilSupplied) or fAbs(OilSupplied-Volume)>0.15),1,0) as ErrorFlag

Они помечают, отличается ли значение, причём существенно или нет, а также, если объём отличается ровно в 10 раз, возможно, произошло смещение десятичной части счетчика. На следующем шаге выбирались транзакции с минимальной разницей во времени со счётчиком (DifTS) и минимальным состоянием флага ошибки и флага суммы.

По получившимся данным была построена таблица сопоставления с данными счётчиков и базы данных. Специальным цветом были помечены отличающиеся значения и данные, отсутствующие в системной базе данных, но присутствующие по данным датчиков.

Рис. 11. Таблица сопоставления данных

V. Заключение

Комплексный анализ киберфизических систем позволяет выявить различного рода расхождения в работе оборудования и учётом. Это может сигнализировать как об ошибке в работе оборудования, так и в возможных недобросовестных действиях персонала.

Список литературы

[1] Евстафьев Д. Четвертая промышленная революция: пропагандистский миф или «знак беды»? [Электронный ресурс] URL: https://www.if24.ru/4-promyshlennaya-revolyutsiya-mif/ (дата обращения 15.03.2020).

[2] Наполова Е.И., Кожевников С.В. Защита компьютерных сетей на основе технологии Virtual Private Network // Экономика и качество систем связи. 2018. № 2 (8).

[3] PSS 5000. Technical Manual For systems with CPB50x [Электронный ресурс]. – Интернет-сайт. – URL: http://www. https://www.doms.com/sites/doms.com/files/assets/80304618.pdf (дата обращения: 15.03.2020).

[4] Plakhotnikov D.P. and Kotova E.E. "The Use of Artificial Intelligence in Cyber-Physical Systems," 2020 XXIII International Conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, 2020, pp. 238-241, doi: 10.1109/SCM50615.2020.9198749.

[5] Broucke S. Practical Web Scraping for Data Science / S. Broucke, B. Baesens. Apress, Berkeley, CA, USA, 2018. 306 p. ISBN 978-1-4842-3582-9.