УДК 004.021

ПРИМЕНЕНИЕ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ И АЛГОРИТМОВ КЛАССИФИКАЦИИ ТЕКСТА В ЗАДАЧАХ ОБНАРУЖЕНИЯ СБОЕВ СИСТЕМ ХРАНЕНИЯ ДАННЫХ

© 2020 М.Б. Успенский

Санкт-Петербургский Политехнический университет Петра Великого

Статья поступила в редакцию 17.12.2019

В статье рассматривается построение диагностической модели системы хранения данных с использованием аппарата онтологического моделирования и методов машинного обучения и её применение для анализа системных журналов вычислительных узлов системы с целью обнаружения неисправностей. Для описания неочевидных или тяжело формализуемых связей между параметрами и состояниями компонентов систем хранения данных, предлагается механизм, основанный на использовании внешних объектов, таких как предварительно обученные алгоритмы машинного обучения. В качестве примера реализации такого типа связей описывается применение алгоритма классификации текстов для определения неисправностей на основании результатов анализа журналов программного обеспечения.

Ключевые слова: онтологическое моделирование, обнаружение сбоев, машинное обучение, классификация текстов.

ВВЕДЕНИЕ

Вопрос своевременного обнаружения неисправностей в корпоративных системах хранения данных (СХД) разного уровня в настоящее время приобретает большую важность, так как такого типа системы всё чаще используются для хранения важной информации (персональных данных, финансовой отчётности, конфиденциальных и секретных данных в гражданской и военной промышленности и т.д.). Безопасность такой информации обеспечивается в том числе и за счёт обнаружения программных или аппаратных неисправностей системы хранения данных, вызванных естественными причинами, такими как повреждение носителей информации, контроллеров хранения, сетевого оборудования и т.п., или какими-либоумышленными действиями.

Так как современные СХД по сути представляют собой сложные распределенные структуры со встроенными вычислительными узлами и сетевым оборудованием, классические подходы, направленные в основном на контроль состояния носителей информации, постепенно утрачивают свою актуальность, вызывая потребность в выработке комплексных подходов, позволяющих диагностировать не только неисправности отдельных компонентов, но и ошибки межкомпонентного взаимодействия.

1. ОПИСАНИЕ ОБЪЕКТА ИССЛЕДОВАНИЯ

Объектом исследованиянастоящей публикации является корпоративная СХД среднего уровня (mid-range), архитектурно представляющая собой Сеть хранения данных (SAN)[1]. В данной архитектуре (см. рисунок 1) выделяют три ключевых уровня:

Уровень контроллеров хранения представляет собой набор вычислительных узлов, объединенных в схему резервирования, предназначен для организации доступа к SAN.

Уровень фабрик предназначен для организации маршрутизации трафика в рамках SAN и представляет собой набор сетевого оборудования, такого как маршрутизаторы, коммутаторы, шлюзы и кабели.

Уровень носителей информации предназначен для фактического хранения данных и содержит носители информации различного типа и размера.

Исходя из вышесказанного, в разрабатываемую диагностическую модель включен набор программных и аппаратных компонентов, иерархически связанный между собой и включающий в себя: контроллеры хранения (серверные ЭВМ, в свою очередь состоящие из набора аппаратных компонентов), сетевое оборудование, носители информации, а кроме того – программные и логические сущности, такие как пулы хранения, объединяющие физические диски в единое пространство (storagepool) [2], логические тома, различные сервисы, предоставляемые системным программным обеспечением СХД (СПО СХД).

Для описания состояния СХД может использоваться широкий спектр параметров, прямо или косвенно характеризующий состояние отдельных компонентов СХД или СХД в целом:

- параметры производительности (в том числе параметры нагрузки контроллеров хране-

Успенский Михаил Борисович, ведущий программист лаборатории промышленных систем потоковой обработки данных. E-mail: mikhail.uspenskiy@spbpu.com

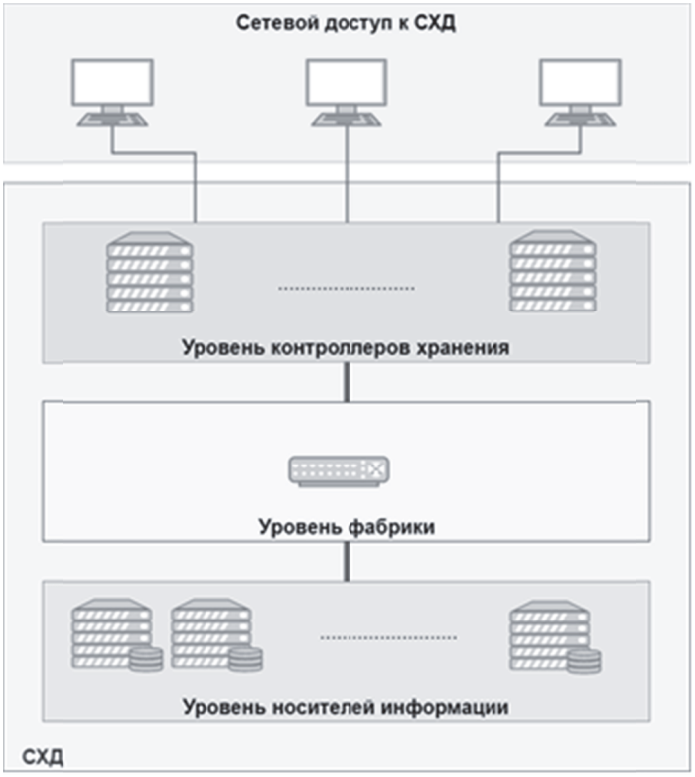

Рис. 1. Структурная схема системы хранения данных

ния, параметры потоков данных, и т.д.);

- параметры здоровья системы (SMART, диагностические данные сетевого оборудования и т.д.);

- параметры емкости (размеченная/неразмеченная емкость, занятое/незанятое место);

- параметры взаимодействия программного обеспечения (доступность сервисов, ошибки межсервисного взаимодействия).

- параметры окружающей среды (температуры, токи, напряжения и т.д.)

Источником получения значений параметров служат сервисы предоставления диагностической информации СПО СХД, журналы СПО СХД, различные датчики, установленные на оборудовании.

2. ОПИСАНИЕ ПОДХОДА К ОБНАРУЖЕНИЮ НЕИСПРАВНОСТЕЙ В СХД

Предлагаемый в настоящей публикации подход к обнаружению неисправностей в СХД основывается на распространенном в настоящее методе диагностики, подразумевающем объединение методов по обнаружению неисправностей на основании построения диагностических моделей и обнаружению неисправностей на основании анализа данных мониторинга [3,4,5].

Для применения такого подхода к обнаружению неисправностей строится диагностическая модель СХД путем сбора, систематизации и агрегации экспертных знаний о режимах работы объекта диагностики, симптомах, описывающих возникновение неисправностей, а также накопленных исторических данных о сбоях и неисправностях, возникавших в объекте диагностики в процессе эксплуатации. Для решения проблемы разнородности накопленной информации из разных источников, было принято решение использовать онтологический подход при построении диагностической модели, так как одним из важных свойств онтологии является возможность описания взаимосвязей между гетерогенными данными [6].

Основным назначением диагностической модели, построенной на основании онтологического подхода является установление соответствия между значениями диагностических параметров, характеризующими элементы объекта диагностики, состояниями элементов диагностики разного уровня иерархии и состоянием объекта диагностики в [7].

Модель топологии СХД представляет собой иерархическую структуру, состоящую из нескольких уровней вложенности, соответствующих подсистемам СХД, компонентам СХД и параметрам СХД. При этом, подсистема может состоять из подсистем, включая подсистемы такого же класса – например, пул хранения может включать пулы хранения. На вход модели подаётся вектор наблюдаемых значений параметров $\{P_i\}$, результатом моделирования является набор состояний (из списка работоспособное, предотказное, неисправность, полный отказ) компонентов СХД, подсистем СХД и СХД в целом $\{S_c, S_s, S\}$ и обнаруженных внештатных ситуаций .

Внештатная ситуация ***{F}*** представляет собой совокупность значений параметров СХД, состояний компонентов или подсистем СХД, характеризующих наступление какой-либо неисправности. Каждая внештатная ситуация при этом классифицируется как соответствующая одному из возможных типов состояний в зависимости от уровня его критичности. При этом работоспособное состояние компонента СХД подразумевает отсутствие диагностированных неисправностей этого компонента, подсистемы – отсутствие диагностированных неисправностей во вложенных компонентах и подсистемах.

Определение состояния СХД в целом и его компонентов и подсистем выполняется снизу-вверх по топологии СХД, и начинается с определения наличия или отсутствия внештатных ситуаций в компонентах СХД на основании текущих значений параметров. Наблюдаемое значение параметра P_i идентифицируется как соответствующее внештатной ситуации при помощи пары [связь, значение] ***{Ld$_j$, V$_j$}***, для *j=1,N,* где *Ld*–связь, *V*–значение, N – число определенных в модели вариантов значений параметра. Данная пара фактически определяет правило, по которому наблюдаемое значение параметра должно оцениваться с точки зрения его штатному или нештатному состоянию. Например, наличие внештатной ситуации в компоненте «Жесткий диск» на основании значения параметра R-W-V-ErrCnt [8] определяется единственной парой *{«is_normal_below»*, 2}.

Состояние S_{ci} компонента C_i определяется при помощи вектора пар [связь, параметр] ***{Lo$_j$, P$_j$}***, для *j=1,…,N,* где *Lo*–связь, *P* – параметр, N – число параметров, определяющих состояние *i*-ого компонента. Связь определяет, каким именно образом состояние зависит от параметра, например, связь «show_in» на рисунке 2 просто обозначает, что состояние S_{c1} определяется как $P_1 \wedge P_N$.

Так как описать внештатные ситуации при помощи непересекающихся диапазонов (т.е., так чтобы можно было записать, что значению Pi от 0 до 10 соответствует одна внештатная ситуация, от 11 до 14 другая, от 15 до 17 третья и т.д.) значений параметров может быть либо невозможно, либо крайне трудоемко, в структуру онтологического описания был добавлен парный условный тип связи «solves_with/described_by», который представляет собой ссылку на внешнюю процедуру, обеспечивающую определение соответствия между вектором входных параметров ***{P}*** и вектором возможных внештатных ситуаций, см. рисунок 3.

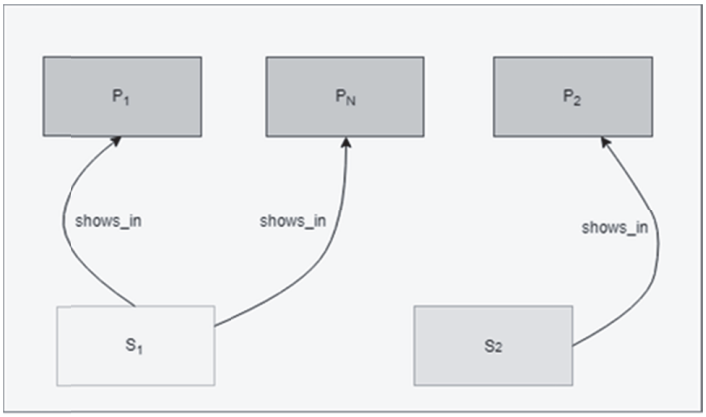

Рис. 2. Связь между параметрами и состояниями системы хранения данных

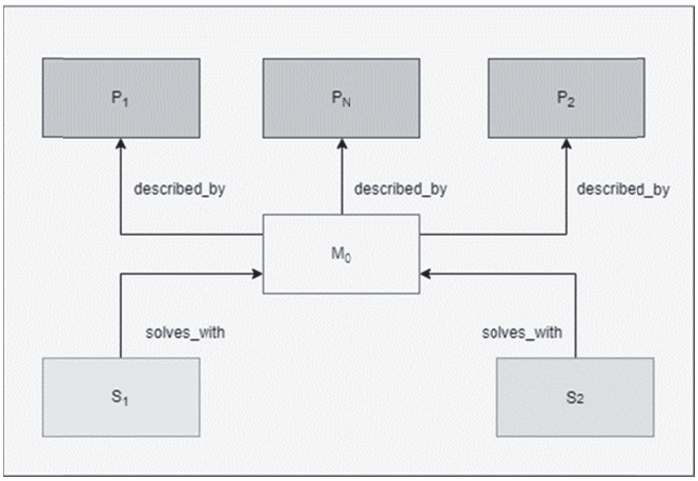

Рис. 3. Связь между параметрами и состояниями СХД

В настоящей публикации рассматривается случай, когда в качестве параметров СХД применяются журналы программного обеспечения СХД, в таком случае в качестве внешнего объекта M_0 для определения связи ***{P}*** и ***{S}*** используется заранее обученный на массиве исторических данных классификатор, основанный на использовании алгоритмов машинного обучения.

С учетом того, что определение состояний подсистем и СХД в целом выполняется аналогичным образом, функцию построения состояния СХД в целом можно сформулировать как $S = \{La, \{Lb, \{Lc...\{Lz,P\}\}\}$, где $L[a,b,c..z]$ – типы связей, а P – диагностические параметры, то есть внештатные ситуации и соответствующие им состояния фиксируются на всех уровнях иерархии элементов объекта диагностики.

3. ПРИМЕНЕНИЕ МОДЕЛИ В ПРОЦЕССЕ ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ

Для разработки онтологической модели, описанной в предыдущем разделе, применяется программный продукт Stanford Protégé[9], являющийся на текущий момент наиболее популярным средством для решения такого рода задач и создающий онтологии в формате OWL[10].

Иерархическая структура классов и объектов онтологиив данном формате, полезная в процессе систематизации и ввода данных, делает неудобным использование онтологической модели непосредственно в процессе обнаружения неисправностей из-за чрезмерно громоздкой структуры данных. Так, например, для определения какой-либо связи в онтологии создаётся подкласс «Связь» класса «Ограничение», причём для связи конкретного типасвязи может быть несколько уровней вложенности(см. пример в таблице 1). Обращение к такой структуре может потребовать значительных вычислительных ресурсов.

Для решения этой проблемы перед применением такойонтологической модели для выполнения автоматического обнаружения неисправностей, предлагается преобразовывать её в упрощенный граф в формате, пригодном для помещения в графовую базу данных[11]. Для этого разработан набор правил по преобразованиюклассов и именованных сущностей онтологии в узлы графа, а отношений между ними – во взвешенные дуги, с помещением уровней вложенности отношений, значений данных и т.п. в весовую характеристику дуги.

Таким образом, всё описание модели конвертируется в формат rdf-nquad[12]: [узел] <связь> [узел] [контекст]. При этом, узлы соответствуют классам или объектам онтологии, а контекст содержит дополнительное описание связи. Пример соотношения объектов исходной онтологии и её упрощенного графового представления приведен в таблице 1.

Таким образом диагностическая модель имеет два представления, одно из которых, он-

Таблица 1. Преобразование онтологии в формате OWL в упрощенное графовое представление

Онтология	<owl:DatatypeProperty rdf:about="is_normal_when"> <rdfs:subPropertyOf rdf:resource="is_normal"/> <rdfs:range> <rdfs:Datatype> <owl:unionOf rdf:parseType="Collection"> <rdf:Description rdf:about="integer"/> <rdf:Description rdf:about="string"/> </owl:unionOf> </rdfs:Datatype> </rdfs:range> <rdfs:label>is_normal_when</rdfs:label> </owl:DatatypeProperty> <owl:Class rdf:about="Health"> <rdfs:subClassOf rdf:resource="REST"/> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="is_normal_when"/> <owl:hasValue rdf:datatype="string">OK</owl:hasValue> </owl:Restriction> </rdfs:subClassOf> </owl:Class>
Упрощенный граф	Health <is_normal > “OK” [class=”when”] Health <subClassOf> REST .

тологическое, предназначено для её наполнения и редактирования, а второе, упрощенное, для использования в рамках процедуры обнаружения сбоев. Это позволяет организовать совместную работу по формированию и наполнению модели экспертами и разработчиками в рамках привычной среды редактирования онтологий, при этом имея все преимущества работы с графовыми базами данных.

4. ИСПОЛЬЗОВАНИЕ ЖУРНАЛОВ СПО СХД ДЛЯ ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ

При решении задач диагностики вычислительных систем широко используется подход, основанный на использовании данных, помещаемых в журналы СПО СХД. При этом, чаще всего такая диагностика основывается либо на обнаружении сообщений с определенным шаблоном, либо преобразование неструктурированного текста в цепочки последовательных событий [13,14,15] И в том, и в другом случае подразумевается, что структура сообщений системного журнала детально анализируется для его использования для обнаружения сбоев.

Альтернативный вариант, более уместный с учетом описанного ранее способа внедрения в модель методов машинного обучения, предполагает работу с журналом, как неструктурированным текстом. Преимуществом такого подхода является отсутствие необходимости анализировать структуру журналов и отдельных сообщений, причины и последовательность их появления и корреляцию сообщений друг с другом. Это важно в условиях работы современных вычислительных систем, когда объем и количество типов хранимых журналов весьма значительно (в рамках настоящего исследования рассматривались пакеты журналов общим объемом более 350 Гбайт и количеством типов журналов от 33 до 105, что делает невозможным их детальный анализ). Предлагаемый в настоящей работе подход, за счёт применения алгоритмов машинного обучения, выполняющих классификацию на основании неочевидных для человека закономерностей, позволяет использовать в процессе диагностики журналы практически любого типа, независимо от объема и структурированности.

Процесс обнаружения неисправностей, предлагаемый в настоящей публикации, предполагает следующие допущения:

1. На уровне онтологии выбираются журналы, изменение которых влияет на состояние анализируемого компонента.

2. Текущим значением параметра *Pi* в случае журнала является блок сообщений, попадающий в заданный временной интервал. В частности, для рассматриваемого объекта диагностики экспериментальным образом был определен интервал в 1 минуту, как вмещающий наименьшее возможное количество сообщений, позволяющее решить задачу обнаружения неисправности.

3. Перед применением алгоритма классификации временные окна каждого журнала проходят процесс приведения к нормальному виду, построение векторного представления[16] и присвоения каждому слову весовых коэффициентов TF-IDF[17].

4. Для обучения классификатора используется размеченная выборка, полученная по результатам поиска причин возникновения внештатных ситуаций в работе СХД. Для каждого типа неисправностей компонентов выделены фрагменты журналов, попадающие во временной промежуток, равный временному окну, заканчивающийся моментом наступления неисправности. Применение алгоритмов машинного обучения позволяет неявно выделить скрытые закономерности, идентифицирующие порядок и состав сообщений, характеризующие неисправность компонентов.

Для экспериментальной проверки эффективности применения такого метода был использован набор из ~1600 пакетов журналов СПО СХД, содержащих сведения о 41 типе неисправностей, связанных с конфигурированием СХД, каждый из которых встречался не менее 15 раз. Для балансировки использовались ещё ~ 4000 пакетов журналов СПО СХД, соответствующих штатному режиму работы СХД. Суммарный объем размеченной выборки – порядка 350 Гб.

Указанный 41 тип неисправностей можно сгруппировать для наглядности по характеру происходящих в указанный промежуток времени процессов в системе – см. рисунок 4.

Для случая применения текстового классификатора RandomForest[18]с оптимизированными параметрами и усреднением по 10 прогонам были получены следующие результаты, см таблица №2

ЗАКЛЮЧЕНИЕ

Рассмотренный в настоящей публикации подход к обнаружению сбоев в системах хранения данных, основанный на применении онтологической модели и применении алгоритмов машинного обучения для анализа журналов СПО СХД позволяет совместить преимущества подходов к диагностике, как использующих применение моделей, так и ориентированных на анализ исторических данных о возникновении неисправностей.

Применение такого подхода позволяет:

- использовать разнородные данные, поступающие из различных источников для иденти-

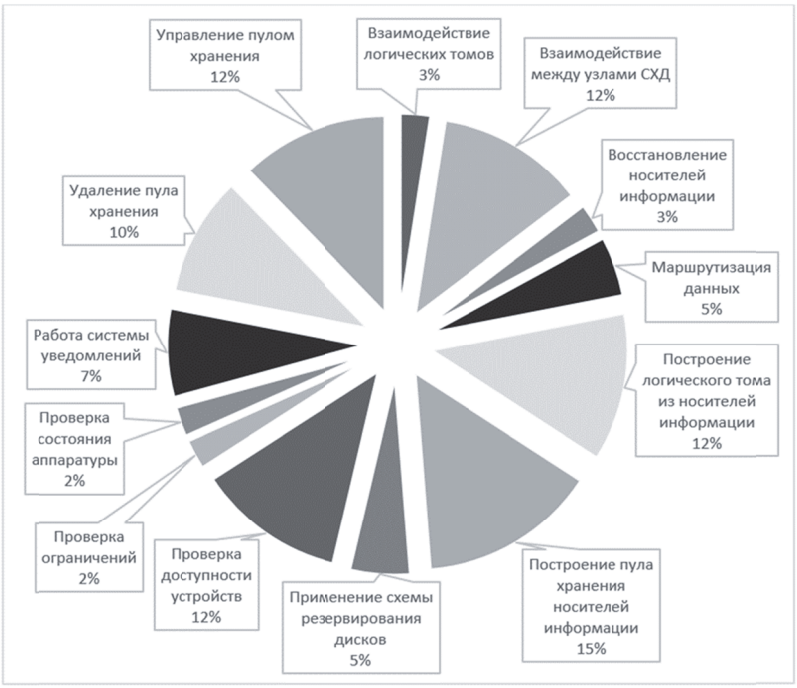

Рис. 4. Распределение количества типов неисправностей по группам

Таблица №2. Результаты классификации

Показатель	**Значение**
Средняя точность	0,74
Средняя полнота	0.71
Средняя f-мера	0.71

фикации состояния отдельных компонентов СХД и СХД в целом;

- оптимизировать организацию как процесса построения, так и применения диагностической модели;

- увеличить число идентифицируемых состояний компонентов за счёт возможности обнаруживать как множествонеисправностей, описываемых при помощи детерминированных правил, так и множество неисправностей, выявляемых на основании вероятностной оценки при помощи алгоритмов машинного обучения;

- выявлять неисправности на основании результатов анализа журналов ПО СХД путем применения алгоритмов классификации текстов с использованием машинного обучения.

Приведение онтологической модели к упрощенному графовому виду позволяет решить проблему структурной сложности иерархии наследований классов и свойств онтологии в формате OWL с сохранением иерархии классов компонентов, подсистем СХД и СХД в целом, что позволяет сократить сложность модели для использования её в рамках диагностической процедуры.

Экспериментальное исследование показало удовлетворительную точность классификации журналов ПО СХД по типам обнаруживаемых неисправностейс использованием алгоритма RandomForestи процедуры предварительной обработки текста. Это подтверждает возможность практического применения такого типа алгоритмов для решения диагностических задач как совместно с диагностической моделью, так, в более частных случаях, и в качестве отдельной диагностической процедуры.

СПИСОК ЛИТЕРАТУРЫ

1. *Troppens U., Erkens R., Muller-Friedt W.* Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FCoE // Wiley – 2019.
2. Storage pools [Электронныйресурс]. – Режим доступа: https://docs.net app.com/ontap-9/index.jsp?topic=%2Fcom.netapp.doc.onc-sm-help-960%2FGUID-31FCCF2B-6BFE-44E9-90B8-4724E4B59A77.html (дата обращения 20.10.2019)
3. *Slimani A., Ribot P., Chanthery E., Rachedi N.* Fusion of Model-based and Data-based Fault Diagnosis Approaches. //IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS. – 2018. – №24-51. – P. 1205-1211
4. *Luo, J., Namburu, M., Pattipati, K.R., Qiao, L., Chigusa, S.* Integrated model-based and data-driven diagnosis

of automotive antilock braking systems //IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. – 2010. – №2-40. – P. 321-336
5. *Jung, D., Sundström, C.A* Combined Data-Driven and Model-Based Residual SelectionAlgorithm for Fault Detection and Isolation //IEEE Transactions on Control Systems Technology. – 2017. – №1. – P.1-15
6. *Mercier D., Cheong H., Tapaswi C.* Unified Access to Heterogeneous Data Sources Using an Ontology //Semantic technology: 8th Joint International Conference. – 2018. – P.104-118
7. *Mamoutova O., Shirokova S., Uspenskij M., Loginova A.* The ontology-based approach to data storage systems technical diagnostics // E3S Web of Conferences. – 2019. – №91.
8. SCSI Commands Reference Manual – Seagate [Электронный ресурс]. – Режим доступа: https://www.seagate.com/files/staticfiles/ support/docs/manual/Interface%20manuals/100293068j.pdf (Дата обращения 20.10.2019)
9. OWL 2 Web Ontology LanguageStructural Specification and Functional-Style Syntax [Электронныйресурс]. – Режим доступа: https://www.w3.org/TR/owl2-syntax/ (Дата обращения 20.11.2019)
10. Dgraph Documentation [Электронный ресурс]. – Режим доступа: https://docs.dgraph.io/ (дата обращения 20.11.2019)
11. RDF 1.1 N-Quads[Электронный ресурс]. – Режим доступа: https://www.w3.org/TR/n-quads/ (дата обращения 20.10.2019)
12. StanfordProtégé [Электронный ресурс]. – Режим доступа: https:// https://protege.stanford.edu/ (дата обращения 20.10.2019)
13. *Nagaraj K., Killian C. E., and Neville J.* Structured comparative analysis of systems logs to diagnose performance problems // NSDI. – 2012. – №1. – P.353–366.
14. *He P., Zhu J., He S., Li J. and Lyu M. R.,* Towards automated logparsing for large-scale log data analysis. // IEEE Trans. Dependable Sec.Comput. (TDSC). – 2018. – №15. – P. 931–944.
15. *Hamooni H., Debnath B., Xu J., Zhang H., Jiang G. and Mueen A.* LogMine: fast pattern recognition for log analytics //CIKM. – 2016. – №1. – P. 1573–1582.
16. *Pande, A., Ahuja, V.* WEAC: Word embeddings for anomaly classification from event logs. //2017 IEEE International Conference on Big Data (Big Data). – 2017.
17. *Salton, G. and Buckley, C.* Term-weighting approaches in automatic text retrieval // Information Processing & Management. – 1988. – №24. – P. 513–523.
18. *Breiman L.* Random Forests //Machine Learning. – 2001. – №45. – P. 5–32.

APPLICATION OF ONTOLOGICAL MODELLING METHODS AND TEXT CLASSIFICATION ALGORITHMS FOR STORAGE SYSTEM FAULTS DETECTION

© 2020 M.B. Uspenskij

Peter the Great St. Petersburg Polytechnic University

This paper describes application of diagnostic model, created with ontological modelling methods and machine learning text classification algorithms, for fault detection, based on system log messages data, in enterprise-level storage system.

Proposed fault detection model uses external procedures for the description ofthe relations between parameters and states of storage systems, based on the implementation of the machine learning algorithms. As an example of such relation, author describes application of the text classification method for the task of software log analysis.

Keywords: ontological modelling, fault detection, machine learning, text classification.

Mikhail Uspenskij, Leading Programmer, Laboratory for Industrial Data Streaming Systems
E-mail: mikhail.uspenskiy@spbpu.com