

БАЗЫ ДАННЫХ DATABASE

УДК 004.5

А. С. Коровин, студент, e-mail: akorovin00@gmail.com,

И. П. Скирневский, ассистент, e-mail: igorskir@yandex.ru

Национальный Исследовательский Томский Политехнический Университет, Томск

Система динамической визуализации больших массивов данных сложных физических экспериментов

Описан один из подходов реализации инструмента, позволяющего отображать статус физических экспериментов с помощью различных видов визуализации. Сложность работы системы заключается в обработке большого объема разнородных данных в реальном времени. Для реализации предложено использовать стек веб-технологии, что позволяет применять инструмент удаленно среди широкого круга пользователей, имеющих разные типы устройств. Разработка такой системы была оправдана, так как на сегодняшний день отсутствуют проекты подобного вида, в то время как существующие приложения справляются с задачами неэффективно.

Ключевые слова: система визуализации, большие данные, мониторинг данных, клиент-серверная архитектура, гибкий интерфейс

Введение

В настоящее время сложность организации информационных систем переходит на уровень, позволяющий хранить и обрабатывать все большие объемы данных. В эру достижения огромной производительности и асинхронных вычислений увеличивается актуальность задачи обработки, интерпретации и предоставления конечному пользователю полученных данных.

Зачастую сложность обработки данных возрастает в силу их разнородности в рамках одного проектного решения и динамики обновления. За последнее время проблема больших данных оказалась в фокусе многих крупных компаний, большинство из которых инвестируют средства для проведения исследований проблем, связанных с большими объемами информации. [1]

В работе рассмотрена проблема визуализации данных, полученных в ходе сложных физических экспериментов. В исследовании [2] говорится, что эффективность визуализации данных оказывает прямое влияние на результаты работы конечных пользователей. Интенсивность работы исследовательской группы значительно увеличивается в силу использования актуальных, динамически обновляющихся, структурированных массивов данных с возможностью гибкого изменения визуальных элементов и вывода дополнительной статистики.

Здесь описан один из подходов реализации инструмента, позволяющего создавать и отображать статус физических экспериментов с помощью различных типов визуализации, использующих не только реальные расчетные данные, изменяющиеся с течением времени, но и архивные данные. Решение, приведенное в работе, позволяет в наглядной форме отобразить существенные для исследователя аспекты изучаемых процессов в эксперименте.

Актуальность разработки системы характеризуется несоответствием существующих приложений к требованиям, описанным в данной работе. Все существующие решения не позволяют выполнять работу удаленно, так как являются настольными приложениями, что не соответствует требованию кроссплатформенности и удаленного доступа с любой аппаратной платформы. В свою очередь, веб-приложения, выполняющие схожие функции, не адаптированы под мобильные платформы. Гибкий интерфейс системы визуализации обеспечивает быстрое преобразование (добавление, изменение) экранов отображения, что позволяет добавлять новые модули отображения данных в динамике, адаптируясь к меняющимся условиям физических экспериментов. Таким образом, описанный в работе подход может быть легко масштабируем для визуализации данных экспериментов любой сложности, что делает его универсальным, а возможность гибкой настройки адаптирует графический интерфейс под конечного пользователя.

1. Область применения системы

В технологическом институте Карлсруэ разрабатывают системы автоматизации для различных физических экспериментов, одним из которых является эксперимент KATRIN (Karlsruhe Tritium Neutrino Experiment) [3], проводимый в целях измерения массы частицы нейтрино, а также эксперимент KITSUBE [4], который направлен на изучение атмосферы, и другие эксперименты. На рис. 1, представлена секция спектрометров, где находится множество сенсоров, с которых собираются данные.

Каждая система автоматизации физического эксперимента имеет набор характеристик, отражающих ее состояние. Подобные характеристики в подавляющем большинстве случаев представляются в различных видах и формах. Данные от эксперимента KATRIN отображаются для конечного пользователя в виде временных рядов, а данные от эксперимента KITSUBE могут быть представлены в виде временных рядов, состоящих из двумерных погодных карт. На рис. 2 (см. четвертую сторону обложки) представлена погодная карта за пятидневный промежуток времени.

Таким образом, выходные массивы данных от подобного рода экспериментов представляют собой трудночитаемые человеком структуры большого объема, поэтому следует выдавать параметры в сжатом виде. Задача визуализации осложняется постоянным изменением данных с течением времени. Неточность в показаниях может привести к неверным вычислениям при дальнейшей аналитике, следовательно, возникает необходимость в постоянной актуализации состояния системы для конечного пользователя.

Доступ к результатам визуализации играет большую роль, так как разные пользователи и группы могут удаленно анализировать данные. Таким образом, необходимо обеспечить доступ к инструменту визуализации всем членам группы: ученым, участвующим в эксперименте, операторам, следящим за ходом работ, и другим участникам команды, в том числе и разработчикам инструмента. Системы с подобным уровнем доступа на сегодняшний день

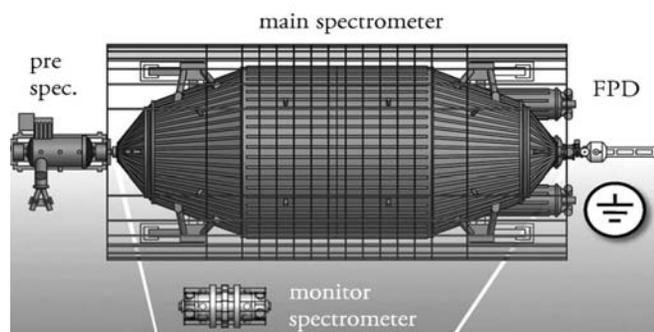


Рис. 1. Секция спектрометров эксперимента KATRIN

реализуются на базе веб-технологий и позволяют удаленно работать с данными с различных платформ, как настольных, так и мобильных.

Растущее число обладателей мобильных устройств порождает спрос на мобильные приложения, в том числе и уже существующих программных продуктов. Специфика таких устройств диктует особые условия визуализации данных, связанные с дизайном, эргономичностью, контентом и другими характеристиками отображения контента. Следовательно, приложение, обеспечивающее визуализацию данных, должно отвечать таким требованиям, как:

- кроссплатформенность;
- оптимизация под мобильные экраны;
- упрощенная и удобная навигация;
- возможность настройки графического интерфейса;
- низкая загрузка передающего канала.

Использование стека веб-технологий для передачи результатов и представления последних пользователям имеет несколько недостатков, относящихся к проблемам производительности, возникающим вследствие ограниченности пропускной способности и задержек в сети. Подробно данная проблема описана в работе [5]. Второй проблемой, связанной с использованием клиент-серверной архитектуры, является уязвимость конфиденциальных данных, хранящихся на серверной стороне, а также возможность перехвата пакетов злоумышленниками во время передачи между клиентом и сервером. Большинство существующих решений, использующихся в Технологическом институте Карлсруэ, не соответствуют требованиям, предъявляемым к гибким системам визуализации данных физических экспериментов, приведенных выше. Следовательно, существующие системы визуализации не способны показывать данные в динамике и не имеют универсальных интерфейсов для адаптации к новым условиям экспериментов. Такие системы недостаточно гибки, что делает процесс внедрения в другие эксперименты ресурсозатратным, а компоненты и визуальные элементы таких систем не оптимизированными к мобильным устройствам.

Следовательно, существует потребность в инструменте, способном хранить, регистрировать и отображать данные с разных экспериментов в различных — как графических, так и текстовых — формах.

Для решения поставленных задач возникает потребность в создании большой библиотеки визуальных элементов отображения. Изменения экранов визуализации и быстрая адаптация к новым условиям экспериментов должны обеспечиваться редактором с графическим интерфейсом.

2. Архитектура приложения

Благодаря увеличению скорости и улучшению качества интернет-соединения, появлению стандарта HTML5 и расширению функциональных возмож-

ностей браузеров, современные веб-приложения по возможностям сравнимы с приложениями для настольных компьютеров. Таким образом, для решения поставленной задачи инструмент реализован с помощью стека веб-технологий.

Такое приложение может быть запущено в браузере любой операционной системы, это уменьшает требовательность к ресурсам аппаратной платформы. Помимо этого, отсутствуют сложности, связанные с обратной совместимостью и поддержкой прошлых версий программ. Пользователям часто приходится решать проблемы обновления установленных копий при появлении новой версия настольного приложения. При использовании веб-приложений такие проблемы отсутствуют, так как существует только одна версия, в которой работают все пользователи.

Другим преимуществом является то, что в веб-приложении используется централизованное хранилище данных. Один из важных аспектов хранения данных — защита от потерь. Это реализуемо только при регулярной репликации баз данных, что неудобно, когда данные распределены среди большого числа компьютеров, так как очень затруднительно создавать и поддерживать все резервные копии проекта. Хранение информации в одном месте позволяет решить эту проблему [6]. Особенно это выгодно в том случае, когда пользоваться приложением будет большое число людей.

Для взаимодействия компонентов распределенной программной системы использовали архитектурный стиль REST. Для его внедрения необходимо наличие клиент-серверной архитектуры, кэширование ответов и унифицированный программный интерфейс. Подробно данный стиль описан в работе [7]. REST-сервис не требует больших затрат при реализации, так как он базируется на уже существующих протоколах HTTP, URI, JSON и RDF. Данный стиль легко масштабируется ввиду возможности включения в каждый ресурс всех состояний, необходимых для обработки запроса. Хорошая производительность достигается за счет кэширования данных и страниц приложения [7].

Для построения интерактивных пользовательских интерфейсов был использован набор технологий AJAX. В работе [8] авторы пришли к выводу, что применение данной технологии приводит к существенной экономии трафика, уменьшению нагрузки на сервер и ускорению реакции интерфейса. Схема взаимодействия клиентской части с серверной с помощью технологии AJAX представлена на рис. 3.

Наличие широкого спектра экспериментов и множества разноплановых групп пользователей системы выдвигает на передний план задачу гибкой настройки инструментов визуализации системы. В рамках решения реализовано отдельное приложение с графическим интерфейсом — редактор страниц визуализации. Этот способ существенно

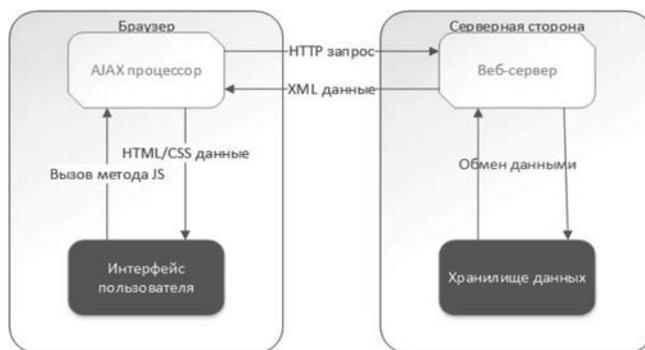


Рис. 3. Схема взаимодействия клиентской и серверной сторон с помощью AJAX

ускоряет разработку и изменение существующих конфигураций страниц без вмешательства специалиста и разработчика продукта.

3. Реализация веб-приложения

Веб-система визуализации реализована как "толстый клиент", где логика приложения перенесена на сторону клиента. В работе [9] говорится, что такой подход может эффективно снизить нагрузку на серверную часть и уменьшить вероятность возникновения задержек сети. Схема работы представлена на рис. 4.

Все сущности элементов визуализации в веб-приложении основаны на моделях и представлениях. Состояние интерфейса меняется с помощью компонента контроллера. Пример схемы работы с элементом визуализации показан на рис. 5, где каждая модель и представление элемента наследуются от базового класса для повышения уровня абстракции и уменьшения дублирования кода. Для хранения,

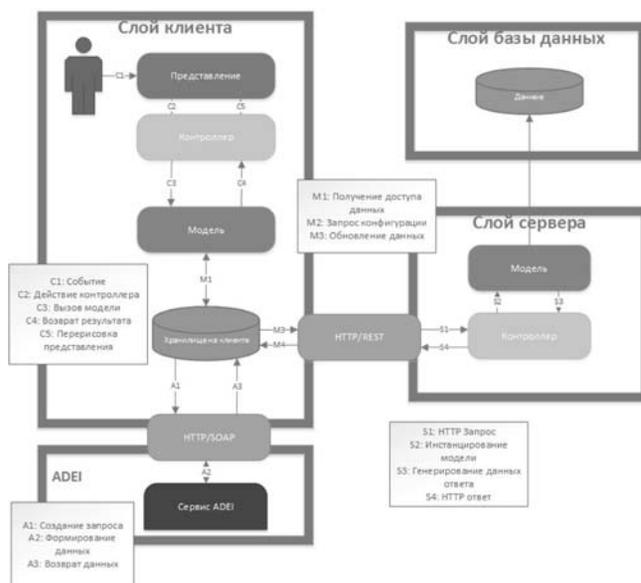


Рис. 4. Архитектура приложения

создания, удаления и управления всеми визуальными элементами использован объект `WidgetManager`.

Контроллер управляет запросами пользователей. В зависимости от типа запроса и информации, содержащейся в нем, контроллер выполняет необходимые действия, управляя ресурсами и объектами. Представление состоит из шаблона, данных сенсоров и кода, который генерирует из шаблона код HTML с помощью элемента конфигурации. Модель — хранилище данных и правила для работы с ними. Модель реагирует на определенные запросы и изменяет свое состояние в зависимости от типа запроса и может содержать такие данные, как размер инструмента визуализации или координаты на экране.

Для хранения и получения данных используется система ADEI. Она предоставляет сервисы для получения многомерных временных рядов с раз-

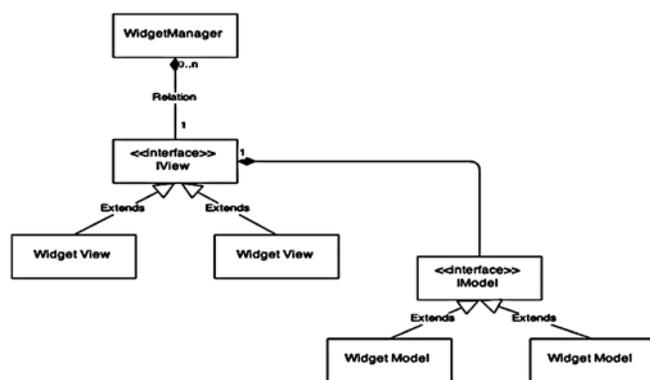


Рис. 5. Схема работы с элементом визуализации

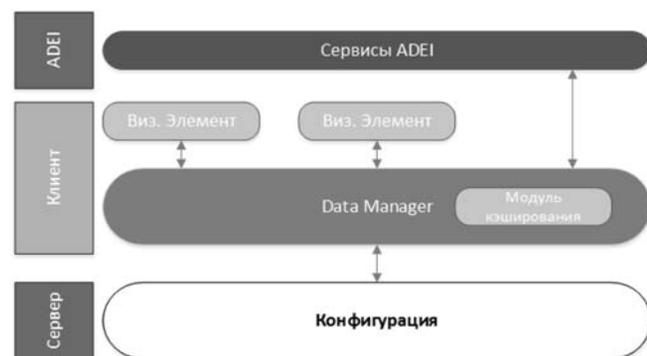


Рис. 6. Схема элементов взаимодействия с данными

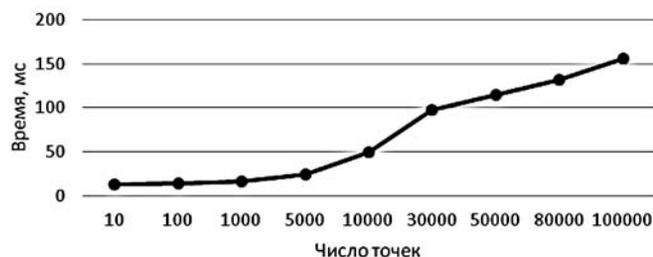


Рис. 7. Время вставки значений в график

личных сенсоров, присутствующих в физических экспериментах. Для получения данных необходимо сформировать запрос для системы ADEI, которая передаст нужные данные веб-приложению [10]. Далее данные кэшируются на клиентской части, чтобы уменьшить нагрузку на сервер. Схема работы с данными представлена на рис. 6, где объект `DataManager` используется для хранения информации об источниках данных и получения данных от сервисов ADEI. При получении данных происходит обновление элементов визуализации.

Одной из главных метрик эффективности приложения является время вставки данных в график, поскольку оно напрямую влияет на время загрузки приложения. На рис. 7 представлены результаты теста времени вставки разного числа точек.

Конфигурация широко используется для описания настроек экранов визуализации. С ее помощью можно систематически идентифицировать, устанавливать связи, сопровождать и управлять различными компонентами системы. На рис. 8 представлен пример структуры конфигурации, содержащей список элементов, описание источников данных и глобальные настройки приложения.

Для передачи данных на сторону клиента используется формат CSV (*comma-separated values*), где значения разделены запятыми. Данный формат выбран ввиду компактности и легкости обработки, поскольку имеет простую структуру. На рис. 9 (см.

```

1 {
2   "screen": {
3     "fluid": false,
4     "glb_sensor_size": 3,
5     "blocksize": 50,
6     "boardsize": 50,
7     "boardsizey": 21
8   },
9   "datasource": {
10    "host": "http://katrin.kit.edu/adei/",
11    "dbgroup": "0",
12    "dbname": "katrin_rep",
13    "server": "fpd"
14  },
15  "elements": {
16    "sensorgroup1": {
17      "type": "sensorgroup",
18      "name": "Magnet Pinch",
19      "size": [6, 5],
20      "coords": [17, 13],
21      "diffsensors": false,
22      "diffgroups": false,
23      "sensors": [{
24        "id": "sensid1",
25        "mask": "102",
26        "unit": "A",
27        "name": "Current",
28        "min": 1.0,
29        "max": 2.0,
30        "precision": 0
31      }, {
32        "id": "sensid2",
33        "mask": "106",
34        "unit": "K",
35        "name": "T1",
36        "precision": 3
37      }
38    ]
39  }
40 }
41

```

Рис. 8. Пример конфигурации экрана визуализации

четвертую сторону обложки) представлена диаграмма сравнения скорости обработки данных объемом в 50 тыс. записей на клиентской части с использованием форматов JSON (*Javascript Object Notation*), XML (*Extensible Markup Language*) и CSV. Результаты показали, что формат CSV обрабатывается быстрее, чем все другие представленные форматы.

4. Графический интерфейс системы

Реализован интерфейс главного экрана эксперимента KATRIN, представленный на рис. 10 (см. четвертую сторону обложки), где отображены результаты визуализации таких данных, как температура главного спектрометра, сила тока, вольтаж детекторов и другие характеристики в реальном времени. Для реализации этого интерфейса была использована идея дизайнерского языка компании Microsoft — Metro, основанная на принципах дизайна швейцарского стиля в типографике.

Разработан интерфейс графического редактора, где пользователь может создать или отредактировать уже созданный экран визуализации эксперимента (рис. 11). Это возможно сделать с помощью как встроенного текстового редактора, так и графического интерфейса, поддерживающего технологию Drag and Drop для удобного изменения координат визуального элемента.

Заключение

Спроектированная система позволяет визуализировать разные виды данных и является важным шагом в процессе разработки прикладных информационно-телекоммуникационных систем, предоставляющим специалистам из разных областей науки возможности надежного и гибкого анализа разнородных данных физических экспериментов. Доступность созданной системы позволяет ученым сконцентрироваться на исследованиях, обеспечивая последних актуальной информацией в удобном формате.

Предложенные архитектура и модель взаимодействия данных могут помочь разработчикам подобных программ реализовывать собственные системы визуализации. В новой версии разрабатываемой системы получения данных ADEI также планируется использовать подобную технологию построения интерфейса и принцип модульности в архитектуре. Разработанное решение будет внедрено в виде основного инструмента визуализации данных таких физических экспериментов, как KATRIN и KITCUBE.

В будущем планируется адаптировать систему к ряду крупных экспериментов, в числе которых

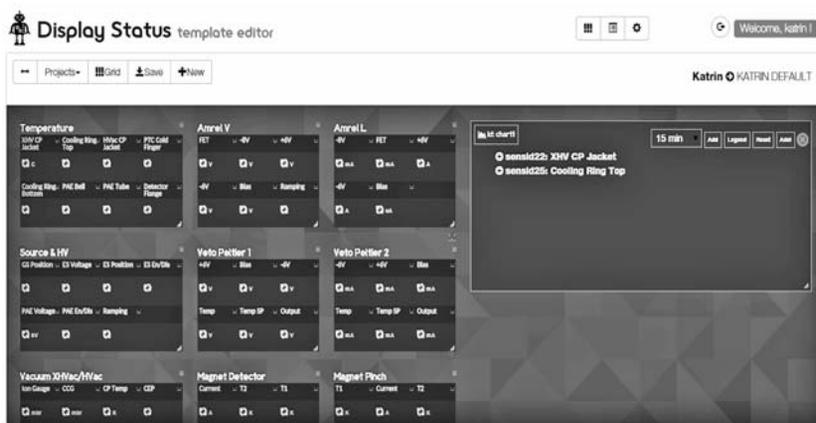


Рис. 11. Интерфейс графического редактора

TOSKA (Toroidal Solenoid Test Facility Karlsruhe) [12], тестирующий сверхпроводимые соленоиды, используемые для исследования реакций термоядерного синтеза, и Edelweiss [13], направленный на изучение частиц темной материи. Кроме того, планируется расширение набора визуальных элементов для обеспечения трехмерной визуализации данных.

Список литературы

1. Riviera J., Van der Meulen R. Gartner Survey Reveals That 73 Percent of Organizations Have Invested or Plan to Invest in Big Data in the Next Two Years [Electronic resource]. URL: <http://www.gartner.com/newsroom/id/2848718> (accessed: 14.03.2015).
2. Plaisant C. The challenge of information visualization evaluation // OSDI Oper. Syst. Des. Implement. 2004. P. 109–116.
3. Wolf J. The KATRIN neutrino mass experiment, Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers // Detect. Assoc. Equip. 2010. Vol. 623. N 1. P. 442–444.
4. Kalthoff N., Bianca A., Andreas W., Martin K. et al. KITcube, A mobile observation platform for convection studies deployed during HyMeX // Meteorol. Zeitschrift. 2013. Vol. 22. N 6. P. 633–647.
5. Andersen D., Bansal D., Curtis D., Seshan S., Balakrishnan H. System Support for Bandwidth Management and Content Adaptation in Internet Applications // Proceeding of the 4th conference on Symposium on Operating System Design & Implementation. 2001. Vol. 4. P. 14.
6. Why Should You Have a Centralized System? | Effective Database Management [Electronic resource]. URL: <http://effectivedatabase.com/resources/why-should-you-have-a-centralized-system/> (accessed: 14.03.2015).
7. Jaki M. Representational State Transfer (REST) // Pro PHP XML and Web Services SE — 17. 2006. P. 633–672.
8. Sghneider F., Agarwal S., Alpan T., Feldmann A. The New Web: Characterizing AJAX Traffic // Lect. Notes Comput. Sci. 2008. Vol. 4979. P. 31–40.
9. Doyle M., Cunningham P. On balancing client-server load in intelligent web-based applications involving dialog. 1999. N 25. P. 1–28.
10. Chilingaryan S., Beglarian A., Kopmann A., Vöcking S. Advanced data extraction infrastructure: Web based system for management of time series data // J. Phys. Conf Ser 2010. Vol. 219, N 4. P. 1–6.
11. Zyp K. AMD: The Definitive Source [Electronic resource]. URL: <http://www.sitepen.com/blog/2012/06/25/amd-the-definitive-source/> (accessed: 14.03.2015).
12. ICPS 2014: Experiments at Campus North [Electronic resource]. URL: http://www.icps2014.com/kit_trip/experiments/ (accessed: 14.03.2015).
13. Edelweiss-III [Electronic resource]. URL: <http://edelweiss.in2p3.fr/> (accessed: 14.03.2015).

A. S. Korovin, Student, e-mail: akorovin00@gmail.com,
I. P. Skirnevskij, Assistant, e-mail: igorskir@yandex.ru,
National Research Tomsk Polytechnic University, Tomsk

Web-Application for Real-Time Big Data Visualization of Complex Physical Experiments

This article examines a tool that provides the flexible user interface for representing status of various physical experiments using different visualization approaches. These experiments produce large volumes of heterogeneous data in real-time, therefore the data processing procedure is complicated. We implemented the tool as a web-based application due to capabilities of modern browsers for interactive visualization, regardless of the operating system and device type (mobile, desktop). Using the architectural style REST simplified component implementation, improved the effectiveness of performance tuning. Frequent changes in experiment conditions require the method for adapting the visualization screens, so we created the graphical user interface builder to arrange graphical control elements using a drag-and-drop technology and WYSIWYG editor. The designed system is a step in the development of applied information technology systems and allows specialists from different fields of science to analyze heterogeneous data of physical experiments effectively.

Keywords: visualization system, big data, data monitoring, client-server architecture, flexible interface

References

1. **Riviera J., Van der Meulen R.** *Gartner Survey Reveals That 73 Percent of Organizations Have Invested or Plan to Invest in Big Data in the Next Two Years* [Electronic resource]. URL: <http://www.gartner.com/newsroom/id/2848718> (accessed: 14.03.2015).
2. **Plaisant C.** The challenge of information visualization evaluation, *OSDI Oper. Syst. Des. Implement*, 2004, pp. 109–116.
3. **Wolf J.** The KATRIN neutrino mass experiment, *Nucl. Instalments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip*, 2010, vol. 623, no. 1, pp. 442–444.
4. **Kalthoff N., Bianca A., Andreas W., Martin K.** et al. KITcube, A mobile observation platform for convection studies deployed during HyMeX, *Meteorol. Zeitschrift*, 2013, vol. 22, no 6, pp. 633–647.
5. **Andersen D., Bansal D., Curtis D., Seshan S. Balakrishnan H.** System Support for Bandwidth Management and Content Adaptation in Internet Applications, *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, 2001, vol. 4, pp. 14.
6. **Why Should You Have a Centralized System?** [Effective Database Management [Electronic resource]. URL: <http://effectivedata-base.com/resources/why-should-you-have-a-centralized-system/> (accessed: 14.03.2015).
7. **Jakl M.** Representational State Transfer (REST), *Pro PHP XML and Web Services SE — 17*, 2006, pp. 633–672.
8. **Schneider F., Agarwal S., Alpan T., Feldmann A.** The New Web: Characterizing AJAX Traffic, *Lect. Notes Comput. Sci.*, 2008, vol. 4979, pp. 31–40.
9. **Doyle M., Cunningham P.** On balancing client-server load in intelligent web-based applications involving dialog. 1999, no. 25, pp. 1–28.
10. **Chilingaryan S., Beglarian A., Kopmann A., Vöcking S.** Advanced data extraction infrastructure: Web based system for management of time series data, *J. Phys. Conf. Ser.* 2010, vol. 219, no. 4, pp. 1–6.
11. **Zyp K.** AMD: The Definitive Source [Electronic resource]. URL: <http://www.sitepen.com/blog/2012/06/25/amd-the-definitive-source/> (accessed: 14.03.2015).
12. **ICPS 2014: Experiments at Campus North** [Electronic resource]. URL: http://www.icps2014.com/kit_trip/experiments/ (accessed: 14.03.2015).
13. **Edelweiss-III** [Electronic resource]. URL: <http://edelweiss.in2p3.fr/> (accessed: 14.03.2015).

ИНФОРМАЦИЯ

Двенадцатая Конференция CEE-SECR / Разработка ПО Digital October, Москва, 27–28 (29) октября 2016 г.



Разработка ПО/CEE-SECR (www.secr.ru) — ежегодная независимая конференция, собирающая около 1000 участников, среди которых разработчики ПО, исследователи, инженеры-практики, лидеры общественного мнения, предприниматели и инвесторы. За годы работы конференция CEE-SECR завоевала заслуженный авторитет главного профессионального события в индустрии разработки программного обеспечения в России и странах Восточной Европы.

В состав программного комитета конференции CEE-SECR входят эксперты мирового уровня, представляющие как ИТ-индустрию, так и научно-исследовательские организации.

Подробности: <http://2016.secr.ru/>