

СБОР И ВИЗУАЛИЗАЦИЯ ЭКСПЛУАТАЦИОННЫХ ДАННЫХ С ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ

Р.Л. Пушков, Е.В. Саламатин, С.В. Евстафиева (ФГБОУ ВО МГТУ "СТАНКИН")

Рассматриваются вопросы сбора эксплуатационных данных с технологического оборудования и машинных параметров системы ЧПУ для контроля правильности использования оборудования, соблюдения регламента его работы. Для хранения и обработки полученных данных используется собственный REST-сервер¹.

Ключевые слова: сбор данных, диагностические данные, архитектура, документоориентированные системы управления базами данных.

Введение

Одной из важнейших задач систем уровня MES является сбор информации, связанной с производством, от систем автоматизации производственного процесса, датчиков, оборудования. Для каждого специалиста, работающего на предприятии, имеют значение различные типы собираемых данных. Так, например, для технолога интерес представляют в реальном времени: параметры режима обработки, отслеживание возможных критических нагрузок на инструмент, время распределения режимов обработки и состояний станка. Для механика необходимы следующие данные: время наработки оборудования, сообщения о критических параметрах обработки, информация с датчиков, анализ температуры технологических узлов оборудования, соблюдение регламента работы с оборудованием до его запуска, в процессе эксплуатации и по окончании работы (например, прогрев шпинделя перед началом работы, контроль предельных параметров работы оборудования). Использование собранных данных позволит: снизить объем брака, переделок, ускорить переналадку оборудования, оптимизировать технологический процесс, повысить точность работы оборудования, продлить срок службы оборудования [1].

Возможны различные способы организации подсистемы сбора данных с технологического оборудования. Одним из вариантов является установка «интеллектуальных» датчиков, которые при изменении значений контролируемых параметров будут отправлять данные в общее облачное хранилище, например, Microsoft Azure, Amazon Cloud. Этот подход имеет несколько недостатков: данные хранятся на удаленном сервере и вопросы их сохранности находятся вне компетенции предприятия. Также отдельно стоит упомянуть о стоимости использования облачного хранилища. В большинстве случаев она зависит от потребленной мощности ресурсов, то есть в случае необходимости решения задач анализа собранных данных стоимость использования ресурса может серьезно возрасти. Второй вариант, который и будет рассматриваться в статье, подразумевает использование собственного сервера с программным обеспечением для сбора, хранения и анализа данных.

Архитектура подсистемы сбора диагностических и информационных данных

Подсистема сбора данных должна взаимодействовать, с одной стороны, с технологическим оборудованием, которое может управляться разными системами ЧПУ, иметь разные протоколы для передачи данных и т.д., а с другой — интегрироваться с MES предприятия для возможности накопления, обработки данных.

Архитектура подсистемы сбора диагностических и информационных данных представлена на рис. 1.

Технологическое оборудование с ЧПУ опрашивается модулями сбора данных с определенным периодом. Для каждого типа станка (токарные, фрезерные, управляемые разными системами ЧПУ) можно определить свой набор данных для сбора с датчиков и/или получения машинных параметров системы [2]. Каждый из модулей сбора данных ориентирован на взаимодействие с определенным оборудованием по определенному протоколу и размещает получен-

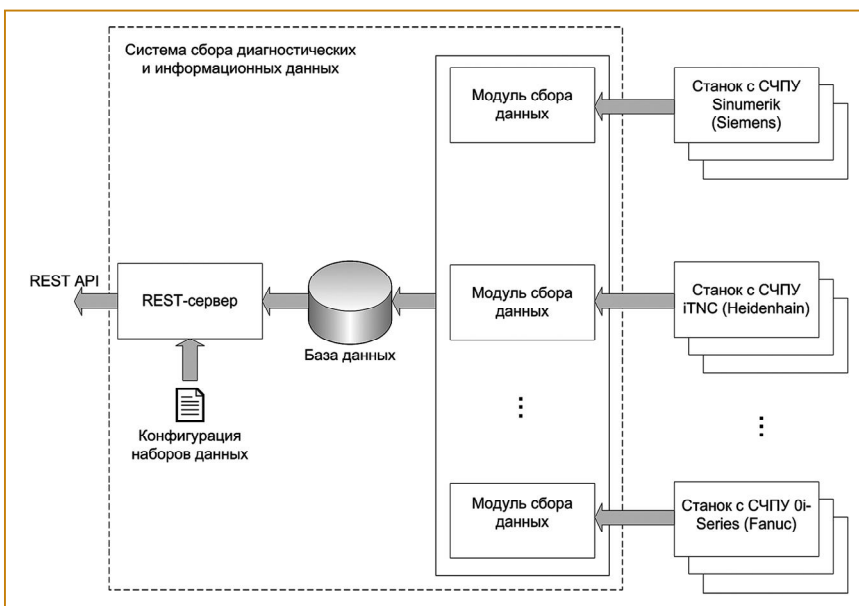


Рис. 1. Архитектура подсистемы сбора диагностических и информационных данных

¹ Работа выполнена при поддержке Министерства образования и науки РФ в рамках выполнения Госзадания (№ 2.1237.2017/4.6) и проводилось с использованием оборудования, предоставленного центром коллективного пользования ФГБОУ ВО МГТУ "СТАНКИН".

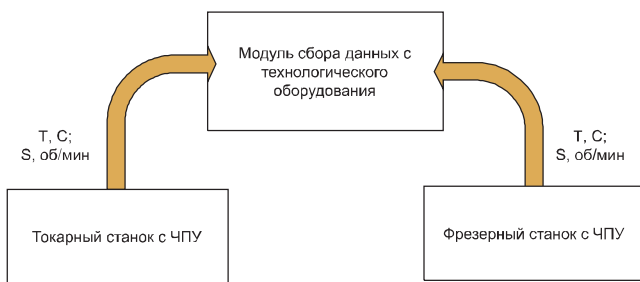


Рис. 2 Сбор информации о температуре и частоте вращения шпинделя с двух станков

ные данные в документоориентированной базе данных. REST-сервер (Representational State Transfer — «передача состояния представления») принимает запросы от клиента. Это удаленный терминал либо модуль, использующий данные для последующих расчетов. Он выдает информацию об оборудовании, наборах данных или сами данные в соответствии с заданной конфигурацией, описанной для сервера [3, 4, 5].

```

{
  "machines": [
    {
      "id": 1,
      "name": "Токарный станок",
      "description": "Модель ТС-200"
    },
    {
      "id": 2,
      "name": "Фрезерный обрабатывающий центр",
      "description": "Модель МП-200"
    }
  ],
  "datasets": [
    {
      "id": 1,
      "machine_id": 1,
      "unit": "об/мин",
      "description": "Обороты шпинделя"
    },
    {
      "id": 2,
      "machine_id": 1,
      "unit": "С",
      "description": "Температура шпинделя"
    }
  ]
}
    
```

Листинг. Фрагмент файла конфигурации

Конфигурация данных

В рассматриваемой архитектуре предусматривается возможность конфигурации данных. Файл конфигурации данных содержит метаданные для всех наборов, собираемых сервером. Файл в формате JSON (JavaScript Object Notation — текстовый формат обмена данными, основанный на JavaScript.) содержит массив описаний элементов оборудования (например, станков) и массив описаний наборов данных.

Рассмотрим систему, в которой эксплуатационные данные поступают в модуль сбора данных с двух станков — токарного и фрезерного (рис. 2). На каждом станке установлен датчик температуры шпинделя, также получается информация о частоте вращения шпинделя [6, 7, 8].

Соответствующий собираемым данным фрагмент файла конфигурации приведен в листинге. В качестве конфигурационных данных указываются типы и названия станков, данные с которых будут собираться, указываются необходимые для получения данные, ассоциированные со станком, также дополнительно можно указать единицы измерения и описание данных. Подобным образом можно конфигурировать все получаемые данные.

Данные из файла конфигурации запрашиваются человеко-машинным интерфейсом для отображения наборов данных, их описаний и принадлежности определенному оборудованию.

Представление данных на сервере

Для хранения значений параметров на сервере можно воспользоваться одним из видов баз данных. Так как

значения различных параметров могут иметь различные типы и составить однозначную схему базы данных сложно, применение реляционных систем управления базами данных весьма проблематично. Для хранения подобных наборов данных лучше подходят документоориентированные системы управления базами данных, например MongoDB. Схемы данных и непосредственно данные хранятся в бинарном формате BSON.

Каждой точке данных соответствует документ, структура которого содержит:

- `_id` — параметр типа ObjectId, который является уникальным идентификатором (первичным ключом) и генерируется СУБД автоматически при добавлении точки в базу данных;
- `dataset` — идентификатор набора данных (int, 32 разряда), определяет принадлежность набора данных определенному источнику. Каждый источник данных имеет свой уникальный в пределах системы идентификатор;
- `time` — параметр типа Date, который содержит временную

метку данных в формате Unix;

- `value` — непосредственно значение любого допустимого в BSON типа.

Запросы к серверу

Запросы к REST-серверу осуществляются по REST API и разделены на три группы.

1. `Machines` — получение сведений об оборудовании, с которого принимаются данные.
 - Вызов `GET/machines` возвращает список оборудования, для которого осуществляется сбор данных.
2. `Datasets` — получение сведений о наборах данных с оборудования.
 - Вызов `GET/datasets` возвращает список наборов данных, которые собираются сервером.
 - Вызов `GET/datasets/{machine_id}` возвращает список наборов данных, которые собираются для оборудования с идентификатором `machine_id`.
3. `Dataset` — получение данных из определенного набора данных.
 - Вызов `GET/dataset/{id}` возвращает все данные набора с идентификатором ID.
 - Вызов `POST/dataset/{id}` возвращает данные набора с идентификатором ID за определенное время. Диапазон времени передается в виде дополнительных параметров вызова.

Практическая реализация

Многие инструкции по эксплуатации станков, особенно с высокоскоростными шпинделями, требуют проведение прогрева шпиндельного узла после длительного простоя. Связано это с двумя факторами.

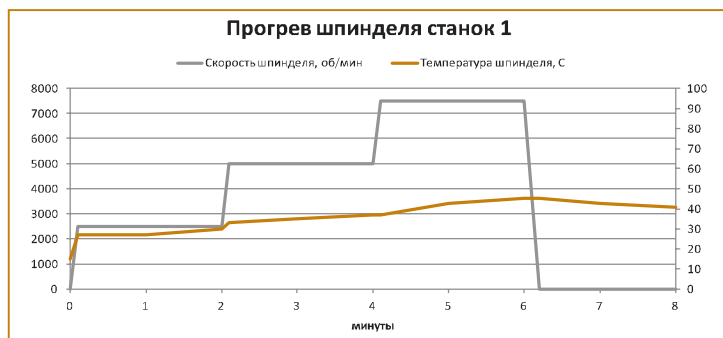


Рис. 3. Изменение температуры и скорости шпинделя после включения станка

1. Необходимо медленно прогреть подшипники и разогнать смазку по механизмам шпиндельного узла до того, как шпиндельный узел станет работать под нагрузкой.

2. Если шпиндельный узел не достиг номинальной температуры, то возможны изменения его геометрических параметров, которые приведут к снижению точности изготавливаемых деталей.

Например, в документации на шпиндельный узел указано, что при простое станка свыше 4 ч, но менее 8 дней необходимо запустить цикл прогрева по следующей схеме.

1. Запустить шпиндель с числом оборотов 25% от максимального значения, проработать 2 мин.

2. Установить число оборотов 50% от максимального, проработать 2 мин.

3. Установить число оборотов 75% от максимального, проработать 2 мин.

4. Шпиндель готов к эксплуатации.

Информация с датчиков температуры шпиндельного узла и скорости шпинделя поступает на сервер системы сбора. В дальнейшем она может быть получена, проанализирована и использована в любых других целях. Например, можно проконтролировать, выполняются ли оператором предписанные инструкции по прогреву шпиндельного узла, или при обосновании отказа гарантийного ремонта за несоблюдение регламента эксплуатации.

Одним из преимуществ использования подобного подхода и технологии является возможность представления данных о состоянии технологического оборудования в Web-интерфейсе [9,10].

Выводы

Предложенный подход по организации программного обеспечения сбора данных позволяет осуществлять сбор данных с разного технологического оборудования. Для работы с различными типами оборудования, в том числе по разным протоколам передачи данных используются разные модули сбора данных. Информация со всех модулей поступает в единую базу

данных, где накапливается и может быть в дальнейшем обработана и использована. Пользователю в большинстве случаев нужна не вся информация, хранящаяся в базе данных о состоянии всех элементов технологического оборудования предприятия, а только какая-то ее часть, например, информация о температуре шпиндельного узла на нескольких станках, или о загруженности оборудования конкретного участка, или информация с датчиков одного конкретного станка. Выбрать данные и представить их в удобной для пользователя форме позволяет REST-сервер с функцией конфигурации данных.

Список литературы

1. Третьяков И. Мониторинг оборудования с ЧПУ: сбор и обработка машинных данных // Промышленные страницы Сибири. 2017. №12. с.28-31.
2. Мартинов Г.М., Пушков Р.Л., Евстафиева С.В. Основы построения однокомпьютерной системы ЧПУ с программно реализованным ядром и открытой модульной архитектурой // Вестник МГТУ "Станкин". 2008. №4. С. 82-93.
3. Никисечкин П.А., Ковалев И.А., Григорьев А.С., Никич А.Н. Построение кроссплатформенной системы для сбора и обработки диагностической информации о работе технологического оборудования на промышленных предприятиях // Автоматизированные технологии и производства. 2016. №4(14). с.51-56.
4. Никисечкин П.А., Ковалев И.А., Григорьев А.С., Никич А.Н. Кроссплатформенная система сбора и обработки диагностической информации о работе технологического оборудования // Вестник МГТУ Станкин. 2017. № 1 (40). С. 94-98.
5. Nikishechkin P. A., Kovalev I.A., Nikich A.N. An approach to building a cross-platform system for the collection and processing of diagnostic information about working technological equipment for industrial enterprises. // MATEC Web Conf. Vol.129, 2017 (International Conference on Modern Trends in Manufacturing Technologies and Equipment (ICMTMTE 2017)).
6. Martinov G.M., Sokolov S.V., Martinova L.L., Grigoryev A.S., Nikishechkin P.A. Approach to the Diagnosis and Configuration of Servo Drives in Heterogeneous Machine Control Systems // 8th International Conference, ICSI 2017 Fukuoka, Japan. 2017. Part II. pp.586-594.
7. Martinova L.L., Grigoryev A.S., Nikishechkin P.A. Tools for Monitoring and Parameter Visualization in Computer Control Systems of Industrial Robots // Advances in Swarm and Computational Intelligence. 6th International Conference, ICSI 2015. Beijing. 2015. Part II. p.200-207.
8. Григорьев С.Н., Мартинов Г.М., Соколов С.В., Козак Н.В. Разработка и применение специализированного инструментария диагностики и настройки следящих приводов в гетерогенных системах управления промышленным оборудованием // Автоматизация в промышленности. 2017. №5. с.29-33.
9. Пушков Р.Л., Евстафиева С.В., Соколов С.В. и др. Практические аспекты построения многотерминального человеко-машинного интерфейса на примере системы ЧПУ "АксиО-МА Контроль" // Автоматизация в промышленности. 2013. №5. С.37-41.
10. Nikishechkin P., Chervonnova N., Nikich A. Approach to the construction of specialized portable terminals for monitoring and controlling technological equipment. In: MATEC Web Conf. Vol. 224, 2018. International Conference on Modern Trends in Manufacturing Technologies and Equipment (ICMTMTE 2018). Sevastopol, Russia. 2018. pp.1-9.

Пушков Роман Львович – ст. преподаватель, Саламатин Евгений Валерьевич – инженер, Евстафиева Светлана Владимировна – ст. преподаватель кафедры компьютерных систем управления ФГБОУ ВО МГТУ "СТАНКИН".

Контактный телефон +7 (499) 972-94-40.
E-mail: pushkov@ncsystems.ru