

*Ю.Н. КОЛЫБИН*, канд. техн. наук, проф. НТУ "ХПИ",  
*С.В. ПОЛОВИНКО*, магистр НТУ "ХПИ"

## **ИСПОЛЬЗОВАНИЕ ЯЗЫКА XML ДЛЯ ОБМЕНА ДАННЫМИ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ МОНИТОРИНГА СЕТИ**

Приведены наиболее распространенные методы организации обмена данными в распределенных системах мониторинга сети с расширением функций посредством плагинов. Предложен основанный на языке XML протокол обмена данными между компонентами системы мониторинга. Рассмотрены некоторые методы сокращения накладных расходов, связанных с избыточностью XML. Табл.: 3. Библиогр.: 8.

**Ключевые слова:** мониторинг сети, плагины, протокол обмена, распределенные системы, язык XML.

**Постановка проблемы.** Современные средства мониторинга вычислительных сетей предоставляют широкие возможности сбора и анализа диагностической информации. Возможности различных продуктов данной категории могут расширяться за счет использования плагинов, скриптов и других подобных средств [1]. При этом производителям систем мониторинга приходится реализовывать сложные интерфейсы для вызова плагинов и получения результатов их работы. Иногда это отдельный интерфейс для каждого поддерживаемого типа плагина. Чаще всего такие интерфейсы обладают жестко заданной структурой и ограничивают дальнейшее развитие системы, поскольку необходимо обеспечивать обратную совместимость различных версий интерфейсов расширения и плагинов.

Кроме того, зачастую современные системы обслуживания сетей, в том числе и инструменты мониторинга, имеют распределенную архитектуру, то есть состоят из нескольких компонентов, которые обмениваются данными по сети. Наличие жестко определенных протоколов также вносит сложности в контроль версий, усложняет работу разработчиков и администраторов системы, если возникает необходимость адаптировать систему к конкретной ситуации и реализовать сбор каких-либо специфических данных, возможно, очень важных для текущей задачи.

Подобные сложности имеют место при разработке самых различных типов программного обеспечения и в значительной мере решаются за счет гибких самодокументируемых протоколов, иерархических структур данных и средств их обработки, а так же обобщенных языков разметки.

**Анализ литературы.** В ходе анализа существующих решений было выяснено, что часто разработчики систем мониторинга организуют плагины в виде отдельных агентов, опрашиваемых по определенному протоколу (чаще всего SNMP) [1]. Подобное решение позволяет добиться

высокой производительности (в случае использования таких нетребовательных к ресурсам протоколов, например, как SNMP), однако расширяемость такой системы сильно ограничена жесткостью протокола.

Другое распространенное решение – оформление плагинов в виде динамических библиотек, содержащих в себе набор функций, каждая из которых выполняет определенную задачу (инициализация, запрос параметров плагина у пользователя, запуск мониторинга, останов мониторинга). В таком случае каждый плагин является "вещью в себе", и для организации, например, обработки полученных данных другим плагином, если данное действие не предусмотрено интерфейсом, разработчикам приходится внедрять различные нестандартные решения, что усложняет разработку. Кроме того, такой подход является очень платформозависимым.

Еще одним, более универсальным, методом является использование в качестве плагинов обычных консольных приложений, формирующих свой текстовый вывод в соответствии с некоторыми правилами форматирования. Данные о результатах мониторинга система получает из потока текстового вывода плагина и кода его завершения. Такой подход (используется, например, в Nagios [2]) может считаться кроссплатформенным и предоставляет большую свободу действий, хотя и несколько меньшую производительность. Естественным развитием такого форматированного текста можно считать язык XML. Несмотря на большую избыточность, являющуюся его главным недостатком, XML успешно применяется в таких распространенных технологиях как XML-RPC, SOAP [3 – 5], протоколе обмена мгновенными сообщениями jabber (XMPP) и др.

**Целью статьи** является разработка собственного протокола на основе XML для связи компонентов распределенной системы мониторинга вычислительной сети между собой и с плагинами, для повышения расширяемости и упрощения систем мониторинга. При этом описания в разрабатываемом формате должны быть достаточно просты для того, чтобы человек мог их читать и редактировать.

**Датчики.** Для мониторинга узлов сети используются, в основном, числовые и строковые значения различных параметров работы этих узлов. Для описания таких значений примем абстракцию "датчик" ("probe"). В синтаксисе XML простейший датчик можно описать так:

Таблица 1

Описание простейшего датчика с использованием языка XML

Описание датчика	Значение датчика
"Число отправленных пакетов"	
<P n="PacketsSent" />	<P n="PacketsSent" /> 100</P>

Значения некоторых датчиков могут быть получены непосредственно в результате опроса состояния системы, другие вычисляются на основе этих

первичных значений. В дальнейшем эти значения используются для визуализации, как входные данные для систем поддержки принятия решений и т.п.

При использовании XML-описаний датчиков в протоколе обмена данными, основная программа опрашивает плагины для получения описаний, а для получения значений конкретного датчика (или нескольких датчиков), отсылает плагину запрос, совпадающий по содержанию с описанием.

Датчик может быть простым, представляя собой одно значение, а может описывать структуру или массив:

Таблица 2

Примеры сложных датчиков

Описание датчика	Значение датчика
<b>"Статистика протокола TCP"</b>	
<pre>&lt;P n="TCP"&gt;   &lt;P n="SegmentsSent" /&gt;   &lt;P n="SegmentsRecieved" /&gt;   &lt;P n="SegmentsResent" /&gt;   ... &lt;/P&gt;</pre>	<pre>&lt;P n="TCP"&gt;   &lt;P n="SegmentsSent"&gt;37&lt;/P&gt;   &lt;P n="SegmentsRecieved"&gt;81&lt;/P&gt;   &lt;P n="SegmentsResent"&gt;4&lt;/P&gt;   ... &lt;/P&gt;</pre>
<b>"Открытые TCP соединения"</b>	
<pre>&lt;P n="Connections" index=""&gt;   &lt;P n="RemotePort" /&gt;   &lt;P n="LocalPort" /&gt;   &lt;P n="RemoteAddr" /&gt;   ... &lt;/P&gt;</pre>	<pre>&lt;P n="Connections" index="0"&gt;   &lt;P n="RemotePort"&gt;80&lt;/P&gt;   &lt;P n="LocalPort"&gt;8890&lt;/P&gt;   &lt;P n="RemoteAddr"&gt; "10.1.3.4"&lt;/P&gt;   ... &lt;/P&gt; &lt;P n="Connections" index="1"&gt;   &lt;P n="RemotePort"&gt;80&lt;/P&gt;   &lt;P n="LocalPort"&gt;8810&lt;/P&gt;   &lt;P n="RemoteAddr"&gt; "10.3.2.5"&lt;/P&gt;   ... &lt;/P&gt;</pre>

Для вычисления датчика на основе значений других датчиков можно использовать подобные описания: "

```
<P n = "ResentProcent" />
100*SegmentsResent / SegmentsSent </P>
```

". Значение датчика ResentProcent будет равняться проценту сегментов, которые узел был вынужден отправить повторно из-за каких-либо неполадок в работе сети. Для простоты все численные значения датчиков в предлагаемом протоколе будут считаться вещественными. В связи с такой организацией работы с числами предлагается ввести оператор округления "@". Пример его использования – "(100\*SegmentsResent/SegmentsSent)@2" – округление значения выражения в скобках до двух знаков после запятой.

При обращении к элементу массива используется оператор "[ ]". Использование его идентично использованию в языке Си.

**Функции.** Кроме датчиков в вычислении выражений могут использоваться функции. Функция представляет из себя датчик, значение которого может меняться в зависимости от переданного ему аргумента.

Таблица 3

Примеры описания функций

Описание датчика	Значение датчика
<b>"Информация о диске"</b>	
<pre>&lt;P n="DriveInfo" arg=""&gt;   &lt;P n="Capacity" /&gt;   &lt;P n="FreeSpace" /&gt;   &lt;P n="FileSystem" /&gt; &lt;/P&gt;</pre>	<pre>&lt;P n="DriveInfo" arg="C"&gt;   &lt;P n="Capacity"&gt;4194304&lt;/P&gt;   &lt;P n="FreeSpace"&gt;20971&lt;/P&gt;   &lt;P n="FileSystem"&gt;FAT32&lt;/P&gt; &lt;/P&gt;</pre>
<b>"Входящие Http-соединения"</b>	
<pre>&lt;P n="IncomeConns" arg="" index=""&gt;   &lt;P n="RemotePort" /&gt;   &lt;P n="RemoteAddr" /&gt; &lt;/P&gt;</pre>	<pre>&lt;P n="IncomeConns" arg="80" index="0"&gt;   &lt;P n="RemotePort"&gt;4023&lt;/P&gt;   &lt;P n="RemoteAddr"&gt;"10.4.5.2"&lt;/P&gt; &lt;/P&gt; &lt;P n="IncomeConns" arg="80" index="1"&gt;   &lt;P n="RemotePort"&gt;8032&lt;/P&gt;   &lt;P n="RemoteAddr"&gt;"10.4.1.8"&lt;/P&gt; &lt;/P&gt;</pre>

Как видно из примера "Входящие Http-соединения", значение функции, как и любого датчика, может быть массивом, что значительно расширяет их возможности. В описанном синтаксисе, однако, имеется тот недостаток, что формат аргумента не описывается явно, а это потребует от разработчиков функции (если она реализована в плагине) написания документации с описанием, иначе использование функции будет затруднено. На данный момент мы видим два решения данной проблемы:

- Использование поясняющих комментариев в XML-описании.
- Задание формата аргумента в виде самостоятельного датчика. Данные,

подающиеся в качестве аргумента функции, должны полностью копировать структуру этого датчика.

**Невычисляемые выражения.** Если хотя бы одно из используемых в выражении значений не может быть вычислено, датчик принимает значение – "недоступен" (#NA). Если в дальнейшем недоступный датчик будет использован для вычисления значения другого датчика, этот датчик тоже станет недоступным. То же самое происходит, если выражение использует значение неопisanного датчика (например, если не подключен нужный плагин). Если в выражении по ошибке будет использован путь к пространству имен, также будет получено значение #NA.

С использованием выражений, предложенных выше, пользователь может расширять возможности системы без внесения изменений в код плагинов.

Разумеется, такой подход не может обеспечить все возможные потребности конечного пользователя, но во многих случаях поможет произвести нужные вычисления без разработки специального плагина.

#### **Достоинства предложенного протокола:**

1. Гибкость протокола обмена. Фактически, он может формироваться "на лету", динамически. 2. XML очень распространен и большинство современных языков программирования имеют средства для работы с ним [6]. 3. Некоторые простые вычисления можно производить при помощи выражений в XML-описании, без использования специальных плагинов. 4. Использование XML позволит значительно упростить систему. Например, при использовании XSLT-преобразований, в качестве консоли мониторинга, в некоторых случаях можно использовать обычный браузер, что вообще позволяет отказаться от разработки консоли мониторинга как отдельного приложения [7]. 5. Удобный для чтения и редактирования человеком формат.

Предложенный протокол был реализован для обмена данными между компонентами демонстрационной программы, служащей для мониторинга работы малых и средних по размеру сетей. Так как в процессе работы над проектом программа многократно изменялась, внесение изменений в протокол с заранее определенными полями и константами было бы чрезвычайно трудоемко, использование же предложенного протокола обеспечило простоту модификации системы мониторинга, что, конечно, показывает насколько просто такие изменения возможно производить при реальном использовании. Подобная гибкость была бы невозможна при использовании, например, SNMP.

#### **Недостатки предложенного протокола:**

1. Большая избыточность кода: теги XML-разметки и длинные имена датчиков составляют большие текстовые объемы, что при частом обмене данными отрицательно сказывается на производительности системы.

2. Для реализации возможности использования датчиков непосредственно в XML-описании необходимо создание собственного языка выражений. Однако синтаксис такого языка будет чрезвычайно простым, поэтому он, скорее всего, не будет вызывать сложностей в освоении.

В некоторой мере устранить первый недостаток позволяют следующие средства:

– При организации обмена системы с плагинами XML-описание используется только на первом этапе. После опроса всех плагинов с целью выяснения структуры дерева датчиков составляется связный граф – интерпретация XML с нумерованными узлами – и в дальнейшем весь обмен осуществляется с использованием этих номеров как идентификаторов датчиков.

– Выражения, используемые при вычислении значений датчиков, можно скомпилировать, переведя в машинные команды. Это позволит избавиться от фазы интерпретации при вычислении каждого выражения. Платформа .NET позволяет производить такую компиляцию относительно просто [8].

– При обмене по сети тексты, размеченные XML, предлагается сжимать одним из распространенных быстрых алгоритмов, что многократно уменьшит объем передаваемых данных.

**Выводы.** Предложен основанный на языке XML протокол для обмена данными между компонентами системы мониторинга сети. Приведены основные особенности протокола. Проведена апробация протокола в реальных условиях. Выявлены некоторые недостатки, однако наличие существенных достоинств показывает, что использование такого протокола в сочетании с некоторыми дополнительными решениями оправдано.

**Список литературы:** 1. Уилсон Э. Мониторинг и анализ сетей. Методы выявления неисправностей / Э. Уилсон. – М.: Лори, 2002. – 350 с. 2. Barth W. Nagios System and network monitoring / W. Barth – San Francisco: No starch press, 2006. – 462 p. 3. Хантер Д. Введение в XML / Д. Хантер. – М.: Лори, 2001. – 638 с. 4. Мак-Лахлин Б. Java и XML / Б. Мак-Лахлин – СПб.: Символ\_Плюс, 2002. – 544 с. 5. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стен. – СПб.: Питер, 2003. – 877 с. 6. Спенсер П. XML Проектирование и реализация / П. Спенсер. – М.: Лори, 2001. – 526 с. 7. Холзнер С. XML. Энциклопедия / С. Холзнер – СПб.: Питер, 2004. – 1104 с. 8. Рухтер Д. Clr via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Д. Рухтер. – СПб.: Питер, 2008. – 656 с.

УДК 004.652

**Використання XML для обміну даними в розподілених системах моніторингу мережі / Колибін Ю.М., Половинко С.В. // Вісник НТУ "ХПІ". Тематичний випуск: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2010. – № 21. – С. 96 – 101.**

Наведено найбільш поширені методи організації обміну даними в розподілених системах моніторингу мережі з розширенням функціоналу за допомогою плагінів. Запропоновано заснований на мові XML протокол обміну даними між компонентами системи моніторингу. Розглянуто деякі методи скорочення накладних витрат, пов'язаних з надмірністю XML. Табл.: 3. Бібліогр.: 8 назв.

**Ключові слова:** моніторинг мережі, плагіни, протокол обміну, розподілені системи, мова XML.

UDC 004.652

**Use of XML for data exchange in a distributed network monitoring system / Kolibin Y.N., Polovinko S.V. // Herald of the National Technical University "KhPI". Subject issue: Information Science and Modelling. – Kharkov: NTU "KhPI". – 2010. – №. 21. – P. 96 – 101.**

The most common methods of data exchange in distributed systems for network monitoring with the extension via plugins are presented. An XML-based protocol for exchanging data between components of a distributed system for monitoring network is suggested. Some methods of reducing computational burden associated with redundancy of XML are reviewed. Tabl.: 3. Refs: 8 titles.

**Keywords:** network monitoring, plugins, protocol for exchanging, distributed systems, XML language.

*Поступила в редакцію 15.04.2010.*