



Math-Net.Ru

Общероссийский математический портал

М. В. Стоцкий, Ю. В. Шевчук, Шаблоны в модуле визуализации web-интерфейса системы мониторинга BotikMon, *Программные системы: теория и приложения*, 2014, том 5, выпуск 2, 63–76

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.9.169

23 марта 2025 г. , 06:53:51



М. В. Стоцкий, Ю. В. Шевчук

Шаблоны в модуле визуализации web-интерфейса системы мониторинга BotikMon

Аннотация. Статья посвящена системе мониторинга BotikMon, разработанной в ИПС им. А. К. Айламазяна РАН. Система предназначена для архивации и визуализации информации, полученной от большого числа датчиков. В данной статье рассматривается вопрос разработки структур данных для страниц «избранное» в графическом интерфейсе пользователя системы BotikMon. Страницы «избранное» дают возможность сводить на одну страницу выбранные оператором данные и настраивать режимы отображения для них. Предлагается подход, основанный на использовании объектов-шаблонов, для которых поддерживается механизм наследования.

Ключевые слова и фразы: система мониторинга, визуализация данных, график, шаблон.

Разработанная в ИПС им. А. К. Айламазяна РАН система мониторинга BotikMon предназначена для мониторинга систем с большим числом источников данных, например суперкомпьютеров или объектов телекоммуникационной инфраструктуры. В настоящее время система эксплуатируется в компьютерной сети «Ботик» г. Переславля-Залесского, работает на рядовом персональном компьютере и контролирует около 200 000 параметров активного сетевого оборудования.

Собранная системой мониторинга информация визуализируется несколькими способами. В настоящей статье рассматривается самый традиционный способ визуализации – представление информации в виде графиков, отображающих изменение выбранного параметра во времени.

Интерфейс системы мониторинга BotikMon предоставляет простую базовую возможность, при использовании которой оператор может найти необходимый параметр и построить график для одного

Результаты, представленные в статье, получены в рамках Программы фундаментальных исследований ОНИТ РАН.

© М. В. Стоцкий, Ю. В. Шевчук, 2014

© ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ ИМЕНИ А. К. АЙЛАМАЗЯНА РАН, 2014

© ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ, 2014

этого параметра. При этом оператору не приходится задавать какие-либо параметры отображения (масштаб, временной интервал, размер графика, сетку, функцию консолидации, цвета), система выбирает их автоматически. Главное достоинство этой базовой возможности – возможность быстро получить график для любого параметра, присутствующего в архиве системы.

Однако на практике этой базовой возможности быстро становится недостаточно. Например, необходимо визуально показать зависимость между двумя или более параметрами. Или требуется следить за какими-то параметрами постоянно. Возникает потребность в изображении нескольких параметров на одном графике, нескольких графиков на одной странице, задания размера графиков, сохранения настроек для повторного отображения выбранных параметров в выбранном виде.

Для решения этих проблем в интерфейсе системы мониторинга VotikMon был предложен механизм «избранное», который позволяет собирать на одной странице выбранные оператором данные, сводить несколько параметров на один график, а также отображать несколько графиков одновременно.

Современные системы мониторинга реализуют различные методы визуализации данных. Во многих из них реализован похожий механизм избранных. Но в изученных самых популярных системах мониторинга возможности механизма избранного недостаточно богаты, чтобы легко подготовить множество однотипных страниц с графиками, отображающими интересующую информацию. Некоторые сетевые системы мониторинга в качестве основной функциональности ограничиваются возможностью просмотра статуса и истории изменения статусов в виде логов [1]. Как правило, в системах мониторинга реализуют простой подход формирования обзорных страниц (избранное) [2–5], который в этой статье описан как подход «Избранное-график-линия».

1. Избранное-график-линия

Механизм «избранное» должен позволять сформировать страницу, на которой может отображаться одновременно несколько графиков, а на каждом графике может быть нарисовано несколько линий. Введем понятия «Избранное», «График» и «Линия» и отобразим их свойства и отношения в виде диаграммы (рис. 1).

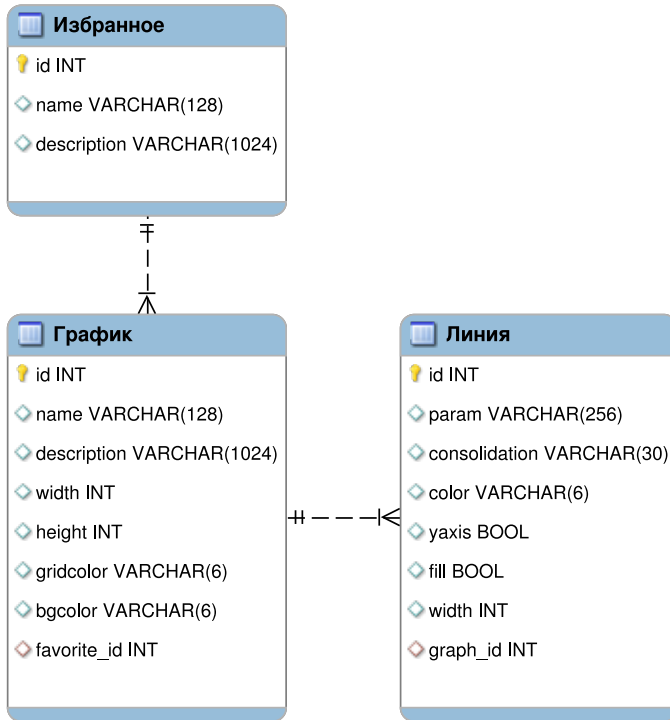


Рис. 1. Первое приближение

Каждый объект среди свойств содержит уникальный идентификатор. Объект «Избранное» состоит из имени и описания. Каждый объект «График» — это имя, описание, ширина и высота графика, цвета фона и сетки (обрамления). Каждый график приписан к некому избранному посредством поля `favorite_id`, таким образом одно избранное может содержать несколько графиков (связь «один ко многим»). Каждый объект «Линия» — это имя параметра в системе мониторинга BotikMon, имя функция консолидации, цвет, флаг привязки к правой оси, флаг заполнения области под линией, толщина линии. Между линией и графиком также определена связь «один ко многим» посредством поля `graph_id`.

Подобный подход к организации визуализации данных позволяет отобразить несколько линий на графике и несколько графиков на одной странице, но оператор часто сталкивается с тем, что ему

приходится наблюдать за группой однотипных объектов. К примеру, оператору требуется следить за набором из N ($N > 0$) однотипных объектов, каждый из которых имеет M одинаковых параметров, которые должны быть отображены на K ($K > 0$) графиках. Причем $M \geq K$, т.к. графики без линий не имеют практического смысла.

Чтобы наглядно сравнить действия оператора при формировании страницы избранного, определим стоимость операций в абстрактных единицах, которые будем обозначать буквой ν :

- Создание объекта требует от оператора одну ν .
- Заполнение или изменение полей объекта — 1ν .
- Копирование объекта с изменением связи — 1ν .

Итак, создание и заполнение полей N объектов «Избранное» требует $2N(\nu)$, создание K объектов «График» для каждого избранного

$$2NK(\nu).$$

Затем создание M объектов «Линия» для графиков

$$2NM(\nu).$$

Итого получаем:

$$2N + 2NK + 2NM(\nu),$$

$$2N(K + M + 1)(\nu).$$

Можно предположить, что создание страниц избранного окажется невероятно утомительным занятием.

2. Шаблоны

Для решения этой проблемы в BotikMon было введено понятие «Шаблон», которое заменило собой «График». Также вводится понятие «Переменная» (рис. 2).

С шаблоном может быть связано несколько переменных, каждая из которых является парой — имя и значение. При определении шаблонов допускается, что различные поля объекта «Шаблон» и объектов «Линия», связанных с шаблоном, могут содержать места для вставки значений переменных. При обработке шаблона места для вставки переменных заменяются на значения связанных с шаблоном переменных.

Предположим, что определен шаблон T , с которым связана линия со значением поля `param`

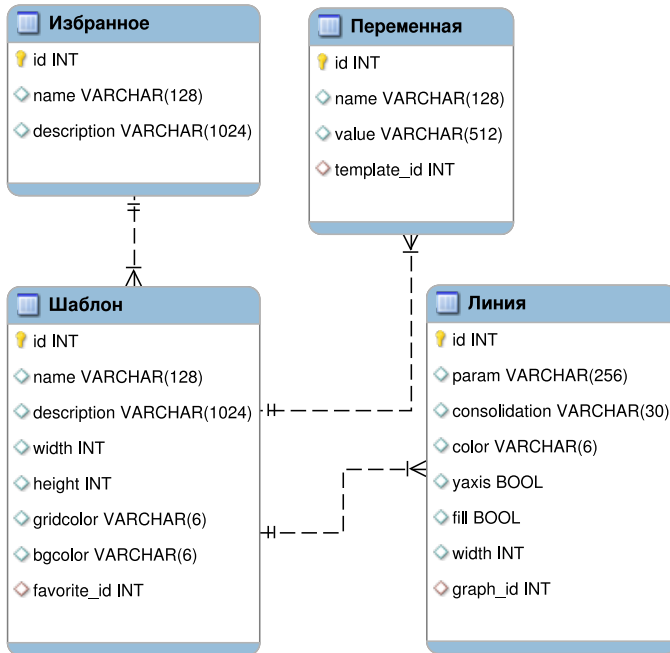


Рис. 2. Шаблоны

`[% server %]/cpu_temp.`

Если с шаблоном T связана переменная с именем `server` и значением, например, `example.com`, то при отображение шаблона в виде графика отобразится история изменения параметра

`example.com/cpu_temp.`

Вернемся к описанной выше задаче о N объектах с M параметрами на K графиках.

Снова необходимо создать N объектов «Избранное». На это потребуется $2N(\nu)$. Затем для одного из избранного создать K шаблонов, на которые потребуется $2K(\nu)$. Далее для шаблонов необходимо определить линии, которые в поле `param` будут содержать место для подстановки значения переменной `server`

$2M(\nu)$.

Не забудем создать объект «Переменная» с именем `server` для каждого шаблона, на которые потребуется еще $2K(\nu)$. Затем, так как

шаблоны будут идентичны для каждого из N объектов, просто $N - 1$ раз скопируем созданные объекты (K шаблонов, M линий и K переменных)

$$(K + M + K)(N - 1)(\nu).$$

Изменение значения переменных с именем `server` в шаблонах

$$K(N - 1)(\nu).$$

Итого:

$$2N + 2K + 2M + 2K + (K + M + K)(N - 1) + K(N - 1)(\nu),$$

$$N(2 + 3K + M) + K + M(\nu).$$

С использованием шаблонов теперь можно единожды создать несколько шаблонов для одной страницы избранного, а затем просто их скопировать столько раз сколько имеется страниц избранного, определяя лишь объекты «Переменная» для каждого шаблона. Но меньше ли операций займет подготовка страниц избранного с использованием нового подхода с использованием шаблонов? Рассмотрим неравенство

$$N(2 + 3K + M) + K + M < 2N(1 + K + M).$$

Левая часть неравенства представляет формула для расчета числа операций для подхода с шаблонами, а правая часть — «Избранное-график-линия».

$$2N + 3NK + MN + K + M < 2N + 2NK + 2MN,$$

$$NK + K + M < MN,$$

$$K < \frac{M(N - 1)}{N + 1}.$$

Если N , M и K удовлетворяют этому неравенству, то подход с шаблонами оправдан для подготовки страниц избранного, иначе лучше использовать первый простой подход. Неравенство показывает, что при большем количестве объектов «Избранное» имеет смысл пользоваться шаблонами. Это интуитивно понятно, если рассмотреть создание только одной страницы избранного. При создании одной страницы избранного помимо объектов «Линия» и «Шаблон» добавляются объекты «Переменная», которых нет в подходе «Избранное-график-линия», а создание дополнительных объектов добавляет лишних операций.

Подход с использованием шаблонов дает некоторые преимущества, но их явно недостаточно. При попытке изменить что-нибудь в графиках, например, цвет линии, отображающей историю изменения утилизации процессора, придется редактировать все ранее созданные шаблоны.

3. Наследование шаблонов

Чтобы решить проблему однотипных объектов, разработчики системы мониторинга BotikMon пришли к идее наследования шаблонов. Помимо того, что объект «Шаблон» связан с объектом «Избранное», он может быть также связан с другим шаблоном. При наследовании шаблон-потомок наследует все линии и переменные, которые связаны с шаблоном-родителем. Кроме этого шаблон-потомок может переопределить переменные шаблона-родителя. Изображение объектов и связей между сущностями очень похоже на предыдущий рисунок, добавляется лишь связь объекта «Шаблон» с самим собой (рис. 3). Эта связь означает, что один объект «Шаблон» может ссылаться на другой объект «Шаблон», при этом первый шаблон является потомком, а второй родителем в отношении наследования.

Теперь шаблоны могут образовывать целые родословные древа. Если оператор запросил страницу избранного, то шаги формирования страницы будут следующими:

- (1) поиск избранного F ;
- (2) поиск всех шаблонов T_s , связанных с избранным F ;
- (3) для каждого шаблона T из $T_s \dots$
 - (a) инициализировать набор переменных TV_s переменными, связанными с шаблоном T ;
 - (b) инициализировать набор линий L_s линиями, связанными с шаблоном T ;
 - (c) T_c присвоить T ;
 - (d) пока определен родитель T_p шаблона $T_c \dots$
 - (i) добавить переменные в набор TV_s , которые связаны с шаблоном T_p , и которых еще нет в наборе TV_s ;
 - (ii) добавить все линии в набор L_s , которые связаны с шаблоном T_p ;
 - (iii) T_c присвоить T_p .
 - (e) применить вставку переменных из набора TV_s к шаблону T и ко всем линиям из набора L_s ;
 - (f) шаблон T с набором линий L_s отобразить в виде графика.

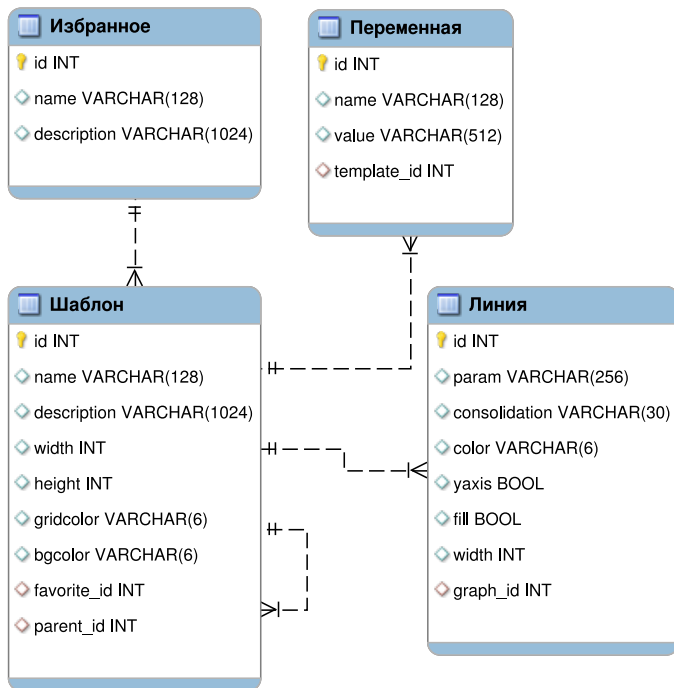


Рис. 3. Наследование шаблонов

Итак, механизм наследования помогает решить проблему наблюдения за однотипными объектами. Теперь чтобы создать набор избранных страниц для N одинаковых объектов, надо создать избранные, затратив $2N(\nu)$. Для одного из избранных создать шаблоны — $2K(\nu)$, M линий для шаблонов — $2M(\nu)$, одна переменная для каждого шаблона — $2K(\nu)$. Для еще одного избранного создать K шаблонов, которые будут потомками от первых K шаблонов — $2K(\nu)$, для каждого потомка определяем переменную **server** — еще $2K(\nu)$. И наконец копирование шаблонов потомков и их переменных для оставшихся избранных потребует $2K(N - 2)(\nu)$, а также изменение

значение переменных с именем `server` — $K(N - 2)(\nu)$. Итого:

$$\begin{aligned} 2N + 2K + 2M + 2K + 2K + 2K + 2K(N - 2) + K(N - 2)(\nu), \\ 2N + 8K + 2M + 2KN - 4K + KN - 2K(\nu), \\ 2N + 2K + 2M + 3KN(\nu), \\ 2(N + K + M) + 3KN(\nu). \end{aligned}$$

Снова встает вопрос об эффективности данного подхода. И снова сравним подход с использованием наследования с предыдущим подходом, при котором используются только шаблоны:

$$\begin{aligned} 2(N + K + M) + 3KN < N(2 + 3K + M) + K + M, \\ 2N + 2K + 2M + 3KN < 2N + 3KN + MN + K + M, \\ K + M < MN, \\ K < M(N - 1). \end{aligned}$$

Так как $K > 0$, $M \geq K$ и $N > 0$, то неравенство верно когда

$$N > 2.$$

Конечно подход с использование шаблонов и наследования дает преимущества при создании нескольких однотипных страниц избранного. При этом получается, что при изменении одной линии в родительском шаблоне изменения «автоматически» произойдут и во всех остальных шаблонах, которые являются его производными.

4. Наследование избранных

Встает резонный вопрос о том, почему наследовать можно только шаблоны? Что если можно было бы наследовать не только шаблоны, но и сами объекты избранного?

У избранных появляется связь с другими объектами избранного, подобно связи между объектами-шаблонами из предыдущего раздела статьи. А также объекты-переменные могут теперь быть связаны не только с объектами-шаблонами, но и с избранными (рис. 4).

Для простоты на схеме связь переменных с избранными и шаблонами осуществляется при помощи двух отдельных полей, что на

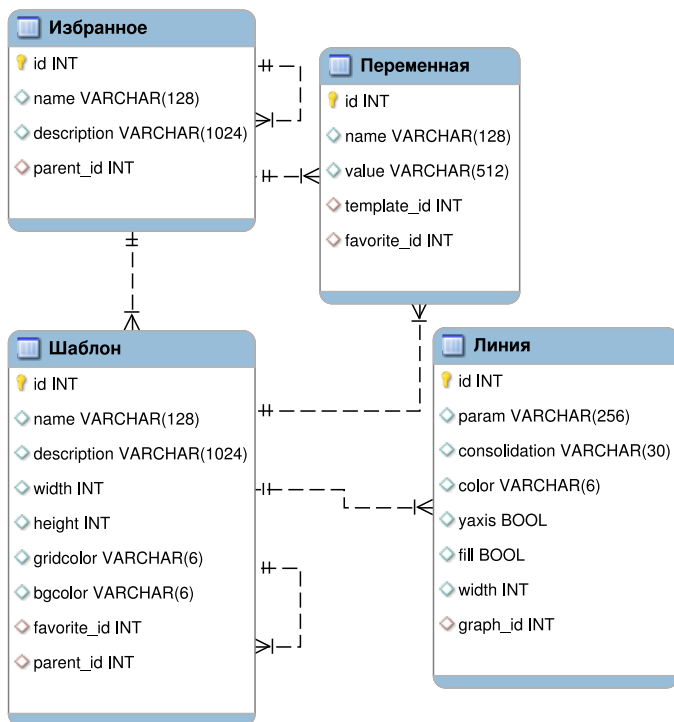


Рис. 4. Наследование избранных

практике неприемлемо. При реализации данного подхода следует использовать так называемую «общую связь», когда связь между объектами описывается парой полей: идентификатор связанного объекта и идентификатор типа объекта. В этом случае переменная может быть связана только с одним объектом (шаблон или избранное) в один момент времени. Но эта тема выходит за рамки данной статьи.

При использовании наследования избранных алгоритм отображения страницы избранного немного усложнится:

- (1) поиск избранного F ;
- (2) инициализировать набор переменных FV_s переменными, связанными с избранным F ;
- (3) инициализировать набор шаблонов T_s шаблонами, связанными с избранным F ;
- (4) F_c присвоить F ;

- (5) пока определен родитель F_p шаблона $F_c \dots$
 - (a) добавить переменные в набор FV_s , которые связаны с избранным F_p , и которых еще нет в наборе FV_s ;
 - (b) добавить все шаблоны в набор T_s , которые связаны с избранным F_p ;
 - (c) F_c присвоить F_p .

Результатом первого этапа подготовки избранного будет набор шаблонов T_s и набор переменных FV_s . Процесс построения графиков очень похож на шаги из прошлого раздела, но немного изменится и будет следующим:

- (1) для каждого шаблона T из $T_s \dots$
 - (a) инициализировать набор переменных TV_s переменными связанными с шаблоном T ;
 - (b) добавить переменные в набор TV_s из набора FV_s , которых еще нет в наборе TV_s ;
 - (c) инициализировать набор линий L_s линиями, связанными с шаблоном T ;
 - (d) T_c присвоить T ;
 - (e) пока определен родитель T_p шаблона $T_c \dots$
 - (i) добавить переменные в набор TV_s , которые связаны с шаблоном T_p , и которых еще нет в наборе TV_s ;
 - (ii) добавить все линии в набор L_s , которые связаны с шаблоном T_p ;
 - (iii) T_c присвоить T_p .
 - (f) применить вставку переменных из набора TV_s к шаблону T и ко всем линиям из набора L_s ;
 - (g) шаблон T с набором линий L_s отобразить в виде графика.

Таким образом наследование можно использовать и на более высоком уровне — на уровне избранных. Становится возможным наследовать и шаблоны между избранными, и линии между шаблонами. Это также позволяет не только определять переменные на уровне шаблонов, но и для группы шаблонов в пределах избранного.

И снова рассмотрим задачу подготовки N избранных с K графиками и M линиями. Теперь для начала необходимо создать один объект «Избранное» (2ν) и одну переменную, привязанную к этому избранному (2ν). Затем потратить $2K\nu$ на K шаблонов. Заметим, что переменные для шаблонов теперь создавать нет необходимости, так как создана одна переменная на уровне избранного. Далее определе-

ние объектов линий с привязкой к шаблонам потребует $2M\nu$ операций. Остальные $N - 1$ избранные создаются и наследуются от первого. Для этого потребуется $2(N - 1)(\nu)$. Плюс создание для каждого избранного переменной, значение которой будет переопределять значение переменной избранного родителя, а это еще $2(N - 1)(\nu)$. Итого:

$$2 + 2 + 2K + 2M + 2(N - 1) + 2(N - 1)(\nu),$$

$$4N + 2M + 2K(\nu).$$

На первый взгляд кажется, что для описанной задачи подход с наследованием избранных самый эффективный с точки зрения количества действий на подготовку страниц избранных. Но так ли это?

$$4N + 2M + 2K < 2(N + K + M) + 3KN,$$

$$2N < 3KN,$$

т.к. $N > 0$, то

$$2 < 3K.$$

То есть наследование избранных всегда будет эффективнее для рассмотренной задачи, чем подход с использованием наследования шаблонов, так как $K > 0$. И почти всегда эффективнее других рассмотренных подходов.

Стоит заметить, что наследование шаблонов вовсе не использовалось при разборе последнего подхода. Но наследование шаблонов — хороший инструмент и может пригодиться в других задачах, когда может возникнуть необходимость отобразить похожие графики в различных избранных без наследования самих избранных.

5. Заключение

Начиная с простого и очевидного подхода «Избранное-график-линия», мы достигли более сложной структуры представления страниц «Избранное» в системе. При этом получилось так, что каждый новый подход обратно совместим с предыдущим. Это означает, что если оператор освоил простой подход, когда линия связана с графиком, а график с избранным, он может использовать его при работе с избранными, не задумываясь о шаблонах и наследовании, пока не ощутит потребность в них. Другими словами, от оператора не требуется сразу понимание всех тонкостей структур данных интерфейса, а обучение может проходить поэтапно от простого к сложному.

Для наглядности рассмотрим пример из жизни. Допустим, что оператору системы мониторинга необходимо наблюдать за 1000 коммутаторов Ethernet ($N = 1000$). У каждого коммутатора имеется 19 портов, про каждый из которых известно о 30 параметрах. Всего 570 линий ($M = 570$) на 30 графиках ($K = 30$).

Подход «Избранное-график-линия»:

$$2 * 1000 * (30 + 570 + 1) = 1202000v.$$

Использование шаблонов:

$$1000 * (2 + 3 * 30 + 570) + 30 + 570 = 662600v.$$

Использование наследования шаблонов:

$$2 * (1000 + 30 + 570) + 3 * 30 * 570 = 54500v.$$

Использование наследования избранных:

$$4 * 1000 + 2 * 570 + 2 * 30 = 5200v.$$

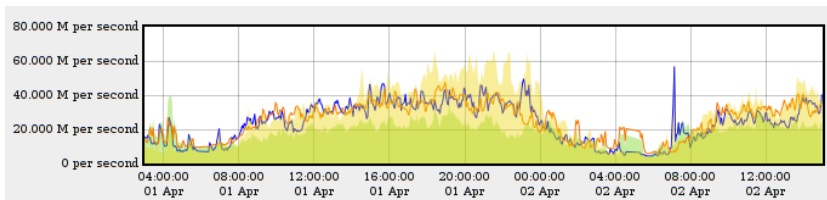


Рис. 5. Пример графика

Рассмотренный подход в настоящее время реализован в системе мониторинга BotikMon (рис. 5). В Web-интерфейсе системы BotikMon реализованы механизмы просмотра и редактирования страниц «избранного», шаблонов и линий. В дальнейшем планируется сделать интерфейс более удобным и привлекательным, чтобы еще больше упростить работу оператора системы.

Список литературы

- [1] *Icinga: Open Source Monitoring*, 2014, URL <http://www.icinga.org/>. ↑64
- [2] M.L. Massie, B.N. Chun, D.E. Culler. *The ganglia distributed monitoring system: design, implementation, and experience* // *Parallel Computing*, 2004. Vol. **30**, no. 7, p. 817–840. ↑64
- [3] Nagios - The Industry Standard in IT Infrastructure Monitoring, 2014. ↑64

- [4] *Opsview / Enterprise IT Monitoring for Networks, Applications, Virtual Servers and the Cloud*, 2014, URL <http://www.opsview.com/>. ↑64
- [5] *Zabbix — An Enterprise-Class Open Source Distributed Monitoring Solution*, 2014, URL <http://www.zabbix.com/>. ↑64

Рекомендовал к публикации

д.ф.-м.н. С. В. Знаменский

Об авторах:



Максим Викторович Стоцкий

Инженер ИПС им. А. К. Айламазяна РАН. Мониторинг и управление, web-программирование интерфейса пользователя

e-mail:

ms@pereslavl.ru



Юрий Владимирович Шевчук

Зав. лабораторией телекоммуникаций ИПС им. А. К. Айламазяна РАН, к.т.н. Сенсорные сети, мониторинг и управление территориально распределенными объектами

e-mail:

shevchuk@botik.ru

Образец ссылки на эту публикацию:

М. В. Стоцкий, Ю. В. Шевчук. *Шаблоны в модуле визуализации веб-интерфейса системы мониторинга BotikMon // Программные системы: теория и приложения: электрон. научн. журн.* 2014. Т. 5, № 2(20), с. 63–76.

URL: http://psta.psiras.ru/read/psta2014_2_63-76.pdf

Max Stotsky, Yury Shevchuk. *Visualization templates in the BotikMon monitoring system.*

ABSTRACT. The article is about the BotikMon monitoring system developed by A.K. Ailamazyan Program Systems Institute of RAS. The system is intended for archival and visualization of data from large number of sources. The visualization subsystem of BotikMon offers so called “selected pages” that show combined data graphs highly configurable by the user. The article focuses on the means developed to facilitate this configuration based on template hierarchy related by inheritance. (*in Russian*).

Key Words and Phrases: monitoring system, data vizualization, graph, template.