

А. А. Ардентов, А. А. Московский, А. Ю. Первин,  
М. В. Стоцкий

## Алгоритмы прогнозирования аппаратных сбоев в системе мониторинга кластерных установок

Научный руководитель: к.х.н. А. А. Московский

Аннотация. В работе описан прототип системы прогнозирования аппаратных сбоев. Предполагается, что об отказе узла кластерной установки можно узнать заранее на основе показаний сенсоров: температура процессора, скорость вращения вентилятора, напряжение. Описывается подход к построению прогнозов и анализу показаний сенсоров. Описывается алгоритм определения опасного состояния кластерных систем.

### 1. Введение

Основная тенденция в развитии высокопроизводительных параллельных компьютеров состоит в том, что среднее число вычислительных узлов, образующих кластер, растет очень быстро [1]. Однако с ростом размера кластера возрастает и вероятность выхода из строя компонентов кластера. Для высокопроизводительного компьютера, включающего в себя тысячи процессоров, жестких дисков, модулей памяти и вентиляторов, вероятность отказа возрастает во много раз по сравнению с однопроцессорной установкой. В связи с этим особое значение приобретает задача обеспечения надежности вычислительных узлов кластера.

Существует несколько подходов, с помощью которых можно достигнуть необходимого уровня отказоустойчивости. Один из способов состоит в создании избыточных аппаратных средств с поддержкой «горячего» подключения. Другой способ увеличить уровень отказоустойчивости — прогнозирование аппаратных сбоев. Это перспективная концепция, благодаря которой можно снизить вероятность возникновения аппаратных сбоев, одновременно снизив эксплуатационные затраты.

---

Представлено по тематике: *Программное обеспечение для суперЭВМ*.

Наше исследование направлено на разработку интеллектуальной Системы Прогнозирования Отказов Кластерных Установок (сокращенно СПОКУ) [2], взаимодействующей с различными компонентами вычислительных кластеров [3] (датчики напряжения на материнской плате, сенсоры температуры процессора, статистика использования сетевой карты и другие). Мы считаем, что анализируя динамику развития «здоровья» кластера, наше программное обеспечение будет способно прогнозировать аппаратные сбои, предоставлять информацию о состоянии кластера. Также такая система должна предупреждать о возможных неполадках в текущей работе кластерной установки, таких как перегревы, скачки напряжения и др.

## 2. Архитектура СПОКУ

Разрабатываемая система прогнозирования сбоев построена на принципе веб-сервисов. Для того чтобы подписаться на услугу прогнозирования сбоев, необходимо установить пакет программ «Агент» на целевой кластер и сообщить организаторам СПОКУ учетную информацию о кластере. Работу СПОКУ можно разбить на следующие пункты:

- Первый шаг — извлечение первичной информации. На кластере, подписанном на услугу прогнозирования сбоев, запускается специальная программа «Сборщик», входящая в состав «Агента», которая отвечает за получение информации от сенсоров. Сбор данных производится периодически через равные интервалы времени, которые при необходимости можно изменять.
- На следующем шаге накопленная статистическая информация от сенсоров пересылается во временное хранилище на головном вычислительном узле кластера. Далее данные в сжатом виде передаются в центральное хранилище для последующей обработки.
- Третий шаг — хранение и анализ. На этом шаге накопленная информация аккумулируется в общее хранилище. Данные о подписанных кластерах хранятся в различных документах единой базы данных. Во избежание нежелательных потерь данных регулярно производится резервное копирование базы данных с последующим архивированием. Таким образом в центральном хранилище доступна информация о состоянии кластера за весь период подписки.

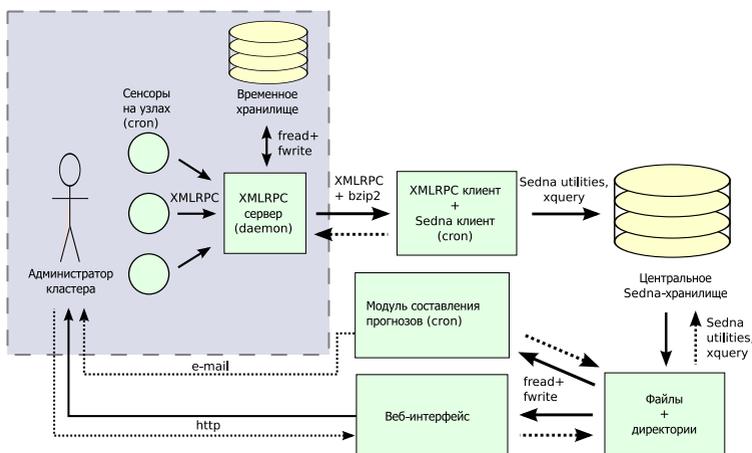


Рис. 1. Схема взаимодействия компонент и потоков данных в системе

В нашей реализации используется системный планировщик задач для того, чтобы периодически запускать «Сборщик» на вычислительных узлах кластера и передавать информацию от сенсоров в центральное хранилище. Текущий прототип СПОКУ способен получать от сенсоров информацию о температуре процессора, напряжении, скорости вентиляторов и количестве ошибок на сетевых интерфейсах.

Выше перечисленная информация о «здоровье» кластера передается и хранится в формате XML [4]. Несмотря на очевидную избыточность XML для хранения сенсорных данных, мы думаем, что XML СУБД существенно упростит систему в целом. Мы надеемся, что, в конечном счете, накладные расходы окажутся невелики.

### 3. Прогноз

#### 3.1. Алгоритмы прогнозирования для СПОКУ

Основное назначение системы прогнозирования состоит в том, чтобы строить точный прогноз состояния узла кластера и вовремя сообщать о вероятной поломке оборудования.

Для того чтобы построить прогноз, по некоторому промежутку времени (период основания прогноза) строится аппроксимирующая

функция, которая затем вычисляется в другом промежутке времени (период упреждения прогноза). В ходе исследования было рассмотрено несколько алгоритмов поиска приближающих функций. Среди них был выбран алгоритм, основанный на методе наименьших квадратов, в котором аппроксимирующая функция ищется в виде линейной комбинации некоторых заранее выбранных функций, называемых базисными [5]. Так как значения оцениваемых параметров ограничены и зачастую периодичны, то в качестве базисных функций были выбраны  $\cos(\frac{2\pi}{a_i}x)$ ,  $\sin(\frac{2\pi}{a_i}x)$  и 1, где  $a_i$  задают периоды колебаний. Коэффициенты линейной комбинации вычисляются автоматически для заданного набора параметров  $a_i$ . Поэтому необходимо заранее выбрать такие  $a_i$ , которые соответствуют сезонным периодам колебаний наблюдаемых характеристик. Изначально за основу были взяты периоды длиной в год, месяц, неделю и сутки. Но эти периоды хоть и описывают общую тенденцию поведения прогнозируемых величин, но зачастую не дают достаточно точного приближения. Поэтому предпринимаются всяческие попытки дополнить текущий набор периодов  $a_i$ , например четверть года, половина суток и т.п. Но иногда система сталкивается с такими ситуациями, при которых наличие добавочных функций лишь ухудшает качество прогноза. Поэтому в системе был реализован алгоритм автоматического подбора оптимального для текущей ситуации набор периодов. Для этого были определены методы оценки качества прогноза.

### 3.2. Методы оценки качества прогноза

Качество прогноза определяется парой — точностью прогноза и стандартным отклонением. Где под точностью понимается процент точек, попавших в доверительный интервал.

Для того чтобы вычислить точность прогноза, необходимо построить доверительный интервал с некоторой фиксированной достоверностью прогноза  $\mu$ . Длина доверительного интервала вычисляется исходя из периода основания прогноза таким образом, чтобы действительное значение попадало в доверительный интервал с вероятностью  $\mu$ . Иначе говоря, выбираем такую ширину интервала, чтобы отношение количества попавших в полосу к общему количеству значений равнялось  $\mu$ .

Стандартное (среднеквадратичное) отклонение вычисляется по известной формуле

$$\sqrt{\frac{1}{n-1} \sum_{i=t_0}^{t_0+n} (x_i - y_i)^2},$$

где  $t_0$  — начальный момент времени,  $n$  — количество точек,  $x_i$  — значение сенсоров в  $i$ -ый момент времени, а  $y_i$  — соответствующее значение аппроксимирующей функции. Значения стандартного отклонения всегда больше или равно нулю. Тем самым вычисляется отклонение действительных значений от значений аппроксимирующей функции.

Чем ближе к нулю отклонение, тем ближе показания сенсоров к аппроксимирующей функции. Чем ближе к 100% точность, тем больше точек попало в доверительный интервал. Но т. к. доверительный интервал можно брать сколь угодно большим, стопроцентная точность всегда может быть достижима. Поэтому наряду с точностью следует учитывать длину доверительного интервала.

### 3.3. Система предупреждения выхода за доверительные интервалы

На основе описанных выше методов оценки качества прогноза разработана система предупреждения выхода показателей датчиков за доверительные интервалы.

Периодически вычисляется аппроксимирующая функция вместе с доверительными интервалами. Затем каждая новая поступающая статистика проверяется на нормальность, т. е. вычисляется в поступившей точке значение аппроксимирующей функции и оценивается разность этого значения с действительным (попало ли в доверительный интервал). В случае аномальности система высылает администратору кластера электронное письмо с оповещением.

Изначально система разрабатывалась для анализа сенсорной статистики, но такой подход может быть применен при анализе временных рядов другого типа. Для этого необходимо подобрать нужные функции. Чтобы добиться более точного приближения можно

использовать понятие корреляции. Если некоторая последовательность  $\{x_i\}$  сильно коррелирует с последовательностью, которая задана функцией  $F$ , то  $F$  можно использовать в построении аппроксимирующей функции. В качестве  $F$  также допустимо брать табличные функции, некоторым образом интерполированные.

Поиск коэффициента корреляции реализован в системе. Например, можно в качестве табличной функции рассматривать показания температуры воздуха в городе, где расположен обслуживаемый кластер. Были посчитаны коэффициенты между последовательностью температур воздуха и показателями температур процессора для кластеров в городах Минск и Переславль-Залесский. После чего для кластера «Первенец-М» в Институте Программных Систем РАН была выявлена зависимость между температурой воздуха за окном и температурой процессора. Но для кластера СКИФ-К1000 в Объединенном Институте Проблем Информатики НАН Беларуси подобной тенденции не наблюдается.

Следствием этих наблюдений было включение функции погоды в аппроксимирующую функцию. Теперь на основе прогноза погоды на неделю (который легко можно узнать из какого-либо источника) можно строить прогноз температуры процессора.

### 3.4. Алгоритм подбора наилучшей комбинации базисных функций

Разработан алгоритм автоматического подбора оптимального набора базисных функций согласно методам оценки качества прогноза. Для сравнения прогнозов между собой была введена штрафная характеристика.

$$\begin{cases} \Omega_{t_i} = a\Omega_{t_{i-1}} + (1-a)Q_{t_i}, \\ \Omega_{t_0} = Q_{t_0}, \\ Q_{t_i} = dS(1.1 - p) \end{cases}$$

где  $t_i$  — это текущий момент времени;  $t_0$  — начальный момент времени;  $a \in [0, 1]$  параметр, который определяет зависимость текущего  $\Omega$  от более старых данных;  $d$  — ширина доверительного интервала;  $S$  — среднеквадратичное отклонение;  $p$  — точность прогноза. Оптимальным является такой набор базисных функций, для которого величина  $\Omega$  имеет наименьшее значение. Следует отметить, что с ростом  $i$  растет точность оценки качества прогноза, что влечет за собой более адекватную аппроксимирующую функцию.

Количество комбинаций базисных функций зависит экспоненциально от их числа, поэтому алгоритм подбора наилучшей комбинации базисных функций требует много вычислительных ресурсов. Чтобы уменьшить время работы алгоритма была создана параллельная версия алгоритма. В качестве средства распараллеливания была выбрана библиотека Tsim шаблонных классов C++, реализующая автоматическое динамическое распараллеливание. Таким образом, система способна определить оптимальный набор функций, наилучшим образом аппроксимирующий недавно полученные данные от сенсоров и экстраполировать их на ближайшее будущее.

Алгоритм подбора оптимальной комбинации базисных функций был встроен в систему построения прогнозов. Роль алгоритма заключается в том, чтобы «отсеивать» базисные функции, которые имеют малую корреляцию с показаниями сенсоров на кластере. Так например, если мы следим за кластером, который находится в специальном помещении, температура в котором не меняется в течении суток и в течении года, то можно с уверенностью сказать, что температура воздуха на улице никак не повлияет на температуру процессора. И поэтому при прогнозировании можно не учитывать функцию погоды. Также если для какого-то кластера статистики накоплено в базе данных всего лишь за месяц, то, при составлении прогноза, не имеет смысла учитывать годовые или полугодовые колебания.

### 3.5. Web-интерфейс

В любой момент пользователи кластера могут проанализировать статистику с помощью описанной ранее системы, используя в качестве инструмента web-интерфейс. В web-интерфейсе необходимо указать имя узла в кластере, процессор, какую характеристику изобразить (температура на процессоре, скорость вращения вентилятора, напряжение и т. д.), выбрать периоды упреждения и основания прогноза. В Web-интерфейсе есть возможность выбора набора базисных функций, которые будут участвовать в составлении прогноза.

Имеется опция построения линий, соответствующих доверительным интервалам, с помощью них можно увидеть, на сколько точно строится прогноз. Также вычисляются точность и стандартное отклонение прогноза в случае, если имеется статистика вне периода основания прогноза, попавшая в период упреждения прогноза. Таким образом, пользователь может сам оценивать качество прогноза.

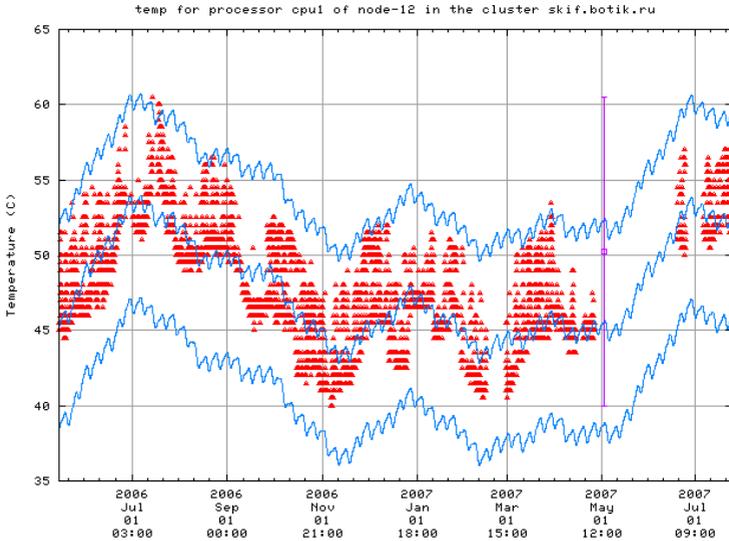


Рис. 2. график с температурами на процессоре

На рис. 2 и рис. 3 представлены примеры графиков, построенных с помощью web-интерфейса. Обозначенные красным цветом точки — фактические значения от сенсоров. Средняя синяя линия соответствует графику аппроксимирующей функции, а верхняя и нижняя — графикам аппроксимирующей функции, сдвинутой на величину доверительного интервала. Вертикальная фиолетовая линия отображает правую границу периода основания прогноза. Проще говоря, слева от фиолетовой линии отображены показания сенсоров, на которых основывается прогноз, и их аппроксимация, а справа — показания, которые прогнозируем, и сам прогноз.

Web-интерфейс — это CGI (Common Gateway Interface) приложение, написанное на языке программирования Perl. Пользователь заполняет форму интерфейса и получает график, нарисованный с помощью программы gnuplot. Для рисования графиков используется модуль для языка программирования Perl Chart::Graph::Gnuplot. Основываясь на информации, введенной пользователем, Perl-скрипт составляет аргументы для функции gnuplot(), которая определена в

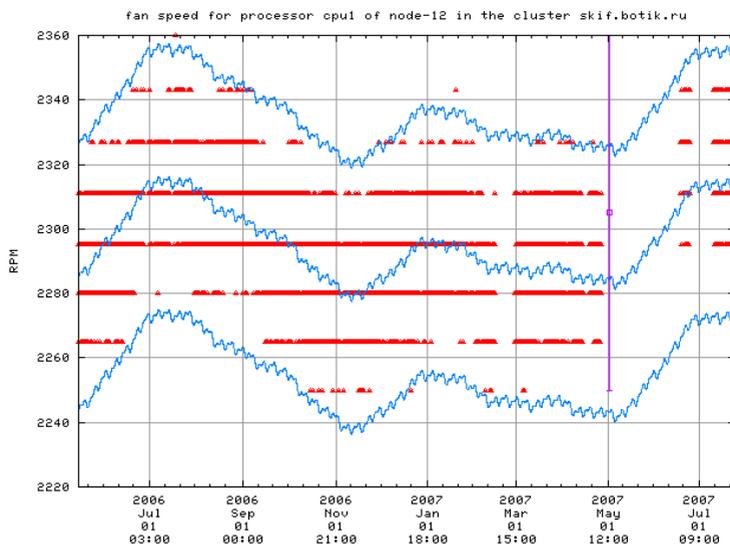


Рис. 3. график со скоростями вентилятора на процессоре

модуле `Chart::Graph::Gnuplot`. Далее библиотечные функции генерируют скрипт, в котором указаны все необходимые опции и параметры, после чего происходит вызов программы `gnuplot` с этим скриптом в качестве аргумента.

#### 4. Выводы

В ходе исследовательской работы был разработан прототип системы прогнозирования аппаратных сбоев. Мы полагаем, что наша разработка может повысить надежность кластерной установки для пользователей и в то же самое время уменьшить общую стоимость владения кластером. Прототип установлен и тестируется на кластерах «Первенец-М» в Институте программных систем РАН, СКИФ-К1000 в Объединенном институте проблем информатики НАН Беларуси, на кластере вычислительного комплекса НИВЦ МГУ, на кластере Infinity Южно-Уральского государственного университета. В центральном хранилище накоплена статистическая информация об этих кластерах за тестовый период.

Несмотря на то, что наша разработка ориентирована прежде всего на работу в высокопроизводительных кластерах, однако, мы полагаем, что сервис прогнозирования отказов может быть востребован для других систем, таких как банки данных, фермы серверов и даже персональные компьютеры.

## Благодарности

Данная работа была частично поддержана и выполнялась в рамках проекта «Сравнительное исследование параметров и характеристик перспективных аппаратно-программных архитектур кластерных мультипроцессорных систем».

## Список литературы

- [1] Jack Dongarra Supercomputers and Clusters and Grids, Oh My! — <http://www.hpc2n.tamu.se/para06/index.php?content=ip1>: University of Tennessee, Oak Ridge National Laboratory. ↑1
- [2] Московский А.А., Первин А.Ю., Стоцкий М.В. *Прогнозирование аппаратных сбоев вычислительных кластеров с помощью систем мониторинга* // «Труды Всероссийской научной конференции Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ», 2007. ↑1
- [3] Lm\_sensors — Linux hardware monitoring. — <http://secure.netroedge.com/~lm78/>. ↑1
- [4] O'Reilly Media I. XML.com: XML From the Inside Out — XML development, XML resources, XML specifications. — <http://www.xml.com>. ↑2
- [5] Суслов В.И., Ибрагимов Н.М., Талышева Л.П., Цыплаков А.А. Эконометрия. Книга третья. Анализ временных рядов. — Новосибирск: Изд-во СО РАН, 2005. ↑3.1
- [6] RAID — Wikipedia, the free encyclopedia. — [http://en.wikipedia.org/wiki/Redundant\\_array\\_of\\_independent\\_disks](http://en.wikipedia.org/wiki/Redundant_array_of_independent_disks). ↑
- [7] IBM Self-Monitoring, Analysis and Reporting Technology (SMART). — <http://smartlinux.sourceforge.net/smart>. ↑
- [8] ISPRAS MODIS MODIS group. — <http://modis.ispras.ru/sedna/index.htm>. ↑
- [9] Лаборатория Параллельных информационных технологий НИВЦ МГУ. PARALLEL.RU — Информационно-аналитический центр по параллельным вычислениям. — <http://parallel.ru>. ↑
- [10] Воеводин В. В., Воеводин В. В. Параллельные вычисления.: БХВ-Петербург, 2002, 600 с. ↑
- [11] Жуматий С. А. Программная среда поддержки эффективного выполнения задач на параллельных вычислительных системах. — Москва, 2005. ↑

ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

УГП, 4М41

A. A. Ardentov, A. A. Moskovsky, A. Y. Pervin, M. V. Stotsky. *Cluster Hardware Monitoring for Failure Predictive Analysis*. (in Russian.)

ABSTRACT. This paper is devoted to research of the failure predictive analysis systems. We consider several ways to achieve necessary level of reliability of hardware components of compute clusters. We introduce the prototype of the cluster hardware monitoring system for failure predictive analysis and its applications in real conditions.

*Перевод проверен:* к.х.н. А. А. Московский