

УДК 004.042
DOI: 10.15827/0236-235X.118.227-234

Дата подачи статьи: 19.12.16
2017. Т. 30. № 2. С. 227–234

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБРАБОТКИ ДАННЫХ В UNIX-ПОДОБНЫХ СИСТЕМАХ

*Е.В. Пальчевский, аспирант, teelp@inbox.ru;
А.Р. Халиков, к.ф.-м.н., доцент, khalikov.albert.r@gmail.com
(Уфимский государственный авиационный технический университет,
ул. Карла Маркса, 12, г. Уфа, 450008, Россия)*

Данная статья посвящена распределенно-модульной обработке информации в автоматизированном режиме. Эта разработка позволяет принимать входящие и исходящие данные на физический сервер в объеме до 2,2 GB/s, при этом распределяя потоковую информацию (весь входящий сетевой трафик на сервер) по физическим и логическим ядрам.

Показана нагрузочная зависимость физических ресурсов от входящей информации. Обоснована целесообразность применения разработанного аппаратно-программного комплекса SDP (Speed data processing), а также представлены структура и принципиальная схема работы. На первом этапе создания комплекса осуществляется разработка алгоритма, на втором – его техническая реализация. Приведен фрагмент исходного кода, отвечающего за уведомление на e-mail как о нагрузке на центральный процессор, так и об основных запускаемых процессах. Описан основной функционал со следующими данными: название функции, цель функции, теоретическая нагрузка, лимит передачи данных (в MB/s) и результат выполнения. На третьем этапе проводится тестирование комплекса SDP, представлены среднесуточные результаты за десять дней.

Созданный аппаратно-программный комплекс позволяет эффективно обрабатывать входящую и исходящую информацию в автоматическом режиме для увеличения пропускной способности при приеме и отправке данных в СУБД MySQL, в том числе при DoS- и DDoS-атаках. Одной из частей комплекса является web-модуль для управления как с персонального компьютера, так и с мобильного телефона. В мониторинговой части web-модуля реализована возможность уведомления SMS-сообщениями о состоянии загруженности физического сервера.

Разработанный аппаратно-программный комплекс показал высокую стабильность при обработке больших объемов данных с минимальной нагрузкой на ЭВМ.

Ключевые слова: обработка информации, нагрузка на вычислительные ресурсы, организация данных, снижение нагрузки информационных потоков, потоки информации, SMS-уведомление, распределенная система, центр обработки данных.

Одним из активно развивающихся направлений является использование АСУ при передаче и обработке данных. Подобными вопросами зачастую занимаются дата-центры (*центры обработки данных* (ЦОД)) [1]. Такие центры работают в крупных городах во многих странах, и их число постоянно растет. ЦОД являются местом для хранения и обработки информации. В них располагаются различные ресурсы, такие как web-сайты, почтовые сервисы, облачные технологии и др. В настоящее время ЦОД нередко подвергаются атакам во внешней глобальной сети, которые могут исходить из любой точки планеты. Расположенное в ЦОД оборудование использует внешние сетевые каналы, что создает потенциальную угрозу информационной безопасности. При этом внешние интернет-ресурсы становятся недоступными для клиентов, например, пользователь не может попасть на свой почтовый сервис или не имеет доступа к нужной веб-странице и т.п.

На оборудовании, размещенном в ЦОД, как правило, установлены операционные системы различных семейств: Windows, Linux, UNIX, Amiga, DOS, Solaris, Illumos, Plan, Inferno [2]. Практически каждая система имеет определенные уязвимости, и нельзя быть застрахованным от DoS- и DDoS-атак, приводящих к нарушению доступности информации и потере пакетов. Правильное распределение информационных потоков по ядрам физического

сервера способствует увеличению пропускной способности при обработке данных в автоматизированном режиме и их сохранении. Распределенно-модульная обработка информации в автоматизированном режиме позволяет принимать входящие и исходящие данные на физический сервер до 2,2 GB/s, при этом распределяя потоковую информацию по физическим и логическим ядрам.

Главным требованием к современным технологиям является равномерное распределение выходных данных. Это означает, что упорядоченная входящая информация (например, потоковый сетевой трафик) должна преобразовываться автоматически. Подобная система автоматизации должна исключать энтропию преобразуемых данных [3].

Вопросами изменения автоматизированных информационных систем занимаются уже продолжительное время, и сегодня данным исследованиям посвящено огромное количество публикаций [3]. Одна из первых работ датируется 1980 годом. Ее авторами являются сотрудники корпорации IBM [4]. С тех пор автоматизации обрабатываемой информации придается огромное значение. В таких системах зачастую предлагаются устаревшие подходы к решению поставленной задачи, нагружающие вычислительные ресурсы [5]. Среднегодовая нагрузка на физические данные ЭВМ имеет тенденцию к ежегодному росту [6]. При огромных потоках информации увеличивается риск потери до-

ступности данных, находящихся на вычислительных ресурсах, которые имеют доступ во внешнюю глобальную сеть. Подобный эффект объясняется затруднениями в выявлении и классификации таких угроз, а также сложностью привязки их к конкретным процессам взаимодействия приложений в логируемых файлах [7].

В статье описана разработка *аппаратно-программного комплекса* (АПК) для обработки массивной и многопоточной информации (атака сетевым трафиком, парсинг, различные подключения от пользователей и т.п.) с последующим занесением данных в СУБД MySQL, почтовым и SMS-уведомлением, а также с выводом в web-интерфейс и распределением сетевой нагрузки по физическим и логическим ядрам сервера.

Преимущества и научная новизна

Основные преимущества предлагаемого решения:

- возможность распределения входящих информационных потоков (сетевой нагрузки) по ядрам физического сервера;
- увеличение пропускной способности СУБД MySQL;
- уведомление по SMS на мобильный телефон, а также на электронную почту;
- возможность удаленного управления АПК через web-интерфейс;
- увеличение сетевой пропускной способности (входящих и исходящих сетевых пакетов);
- использование цепей Маркова, впервые доработанных для распределения сетевой нагрузки.

Научная новизна заключается в распределении сетевой нагрузки по физическим ядрам сервера за счет доработки цепей Маркова, что показано на рисунках 1 и 2.

Таким образом, АПК не только позволяет снизить сетевую нагрузку на вычислительные ресурсы, но и способствует своевременному уведомлению (о состоянии ЭВМ) по SMS на мобильный телефон.

Аналогичные решения

Аппаратно-программные (автоматизированные) *системы обработки данных* (СОД) предназначены для создания информационных ресурсов (вычислительные кластеры). Обработка данных происходит на физическом (аппаратном) уровне. Аппаратно-программные СОД могут быть следующими.

1. *Одномашинные СОД*. Данное решение реализовано за счет единственной ЭВМ (физического сервера) с однопроцессорной структуризацией. Существенными недостатками разработанной системы являются ограничение производительности (до 10 млн операций в секунду), а также отсутствие

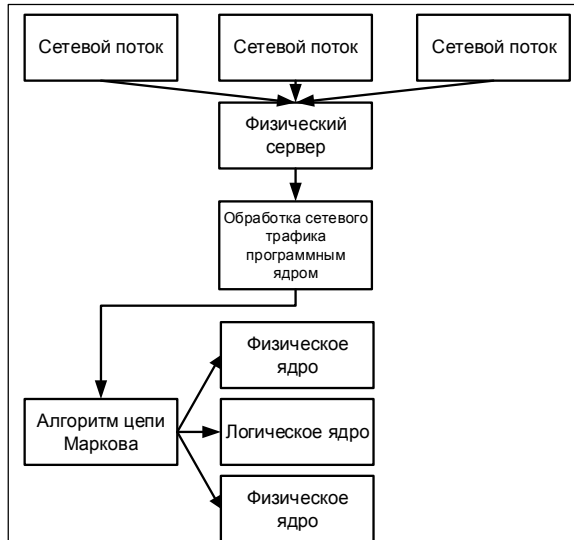


Рис. 1. Взаимодействие алгоритма цепи Маркова с информационными потоками

Fig. 1. Interaction of Markov chain algorithm with information flows

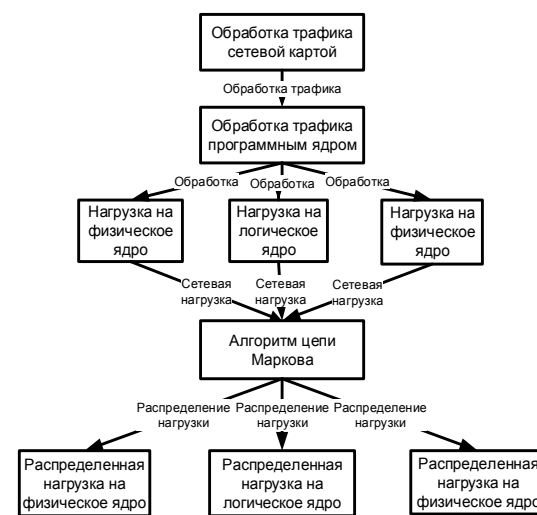


Рис. 2. Распределенная сетевая нагрузка за счет алгоритма цепи Маркова

Fig. 2. Distributed network load due to Markov chain algorithm

многопоточности приема входящего сетевого трафика [8]. В связи с этими недостатками возможен простой системы до 2–3 минут в сутки (по причине отказа в обслуживании оборудования). Структура одномашинной СОД представлена на рисунке 3.

2. *Многомашинные СОД* (вычислительные комплексы). Это решение предполагает связку нескольких физических серверов в единое целое. Данные вычислительные комплексы подразделяются на два вида: с прямой (рис. 4) и с косвенной (рис. 5) связями между ЭВМ. В СОД с косвенной связью взаимодействие происходит через общее запоминающее устройство (доступ с общим набо-

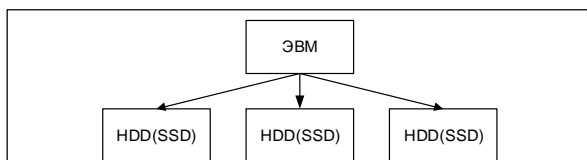


Рис. 3. Одномашинная СОД

Fig. 3. A single-machine data processing system

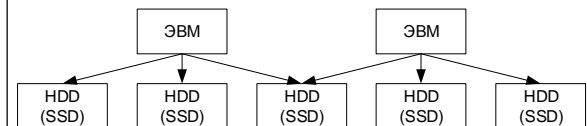


Рис. 4. Многомашинная СОД с косвенной связью

Fig. 4. A multiple-machine data processing system with indirect coupling

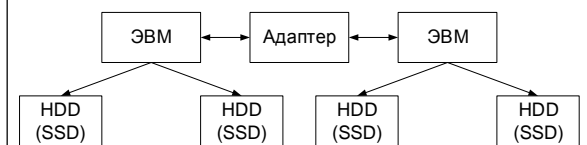


Рис. 5. Многомашинная СОД с прямой связью

Fig. 5. A multiple-machine data processing system with direct coupling

ром данных). В СОД с прямой связью процессы, отвечающие за работу с информацией, имеют доступ ко всему объему данных [9]. Данные СОД являются базовыми средствами обработки информации. В связи с этим в них включено универсальное и стандартизированное (общесистемное) ПО.

Существенным недостатком многомашинной СОД с косвенной связью является низкая пропускная способность принятия и обработки данных [10]. Минус многомашинной СОД с прямой связью: при выходе из строя одной из составляющих (физический сервер) замедляется скорость принятия и обработки данных.

3. СОД, настроенные на заданную область применения (вычислительные системы). Схема работы вычислительных систем показана на рисунке 6.

Все эти СОД основаны на взаимодействии между физическими серверами и вычислительными кластерами. Как правило, такие СОД включают в себя ПО и технические средства, ориентированные на решение определенных поставленных задач [11]. СОД (вычислительные комплексы) подразделяются на два вида:

- вычислительные системы на основе ЭВМ и вычислительных комплексов общего применения;
- вычислительные системы на основе специализированных ЭВМ и вычислительных комплексов.

Недостатком вышеприведенных видов СОД является возможность загрузки ресурсов центрального процессора, что может привести к отказу и задержке обработки данных [12].

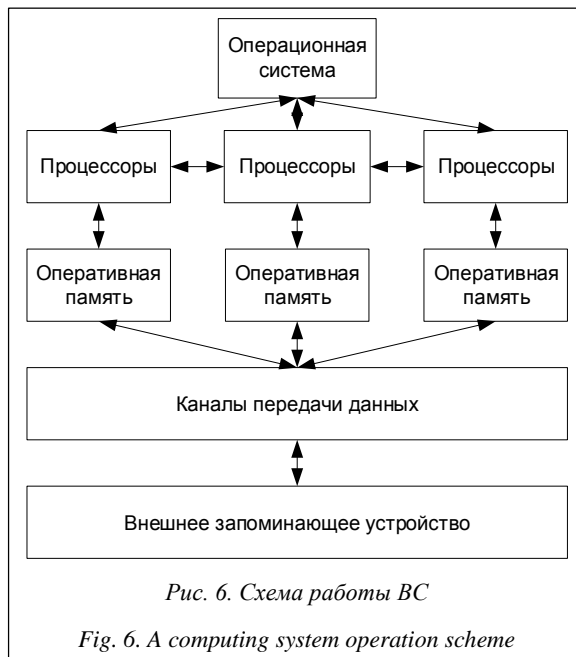


Рис. 6. Схема работы ВС

Fig. 6. A computing system operation scheme

4. Системы телеобработки (ТС). Данное решение предназначено для передачи данных по каналам связи: происходит взаимодействие пользователей с системой за счет терминалов связи (абонентских пунктов). Абонентские пункты подключаются через каналы связи к средствам обработки данных на ЭВМ [13–15]. Схема работы СОД ТС представлена на рисунке 7.

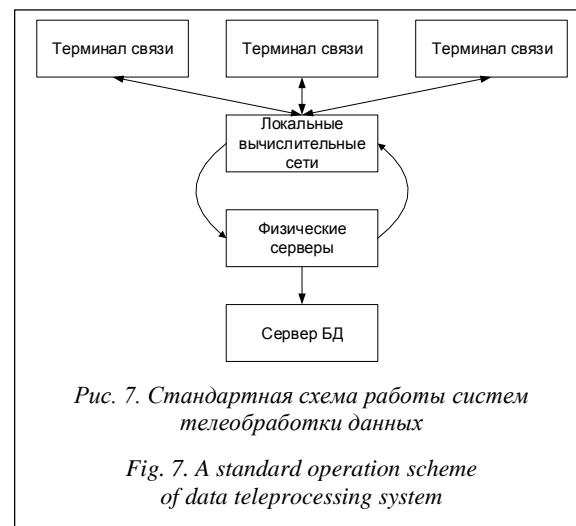


Рис. 7. Стандартная схема работы систем телеобработки данных

Fig. 7. A standard operation scheme of data teleprocessing system

Существенным недостатком подобного рода СОД является невозможность программного расширения пропускной способности канала для увеличения объема принимаемой и обрабатываемой информации.

Разработка и реализация автоматизированной обработки информации

Система SDP (Speed data processing) – разрабатываемый АПК, предназначенный для массивной и

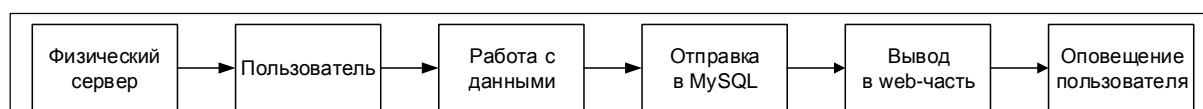


Рис. 8. Структура АПК SDP

Fig. 8. The structure of SDP complex

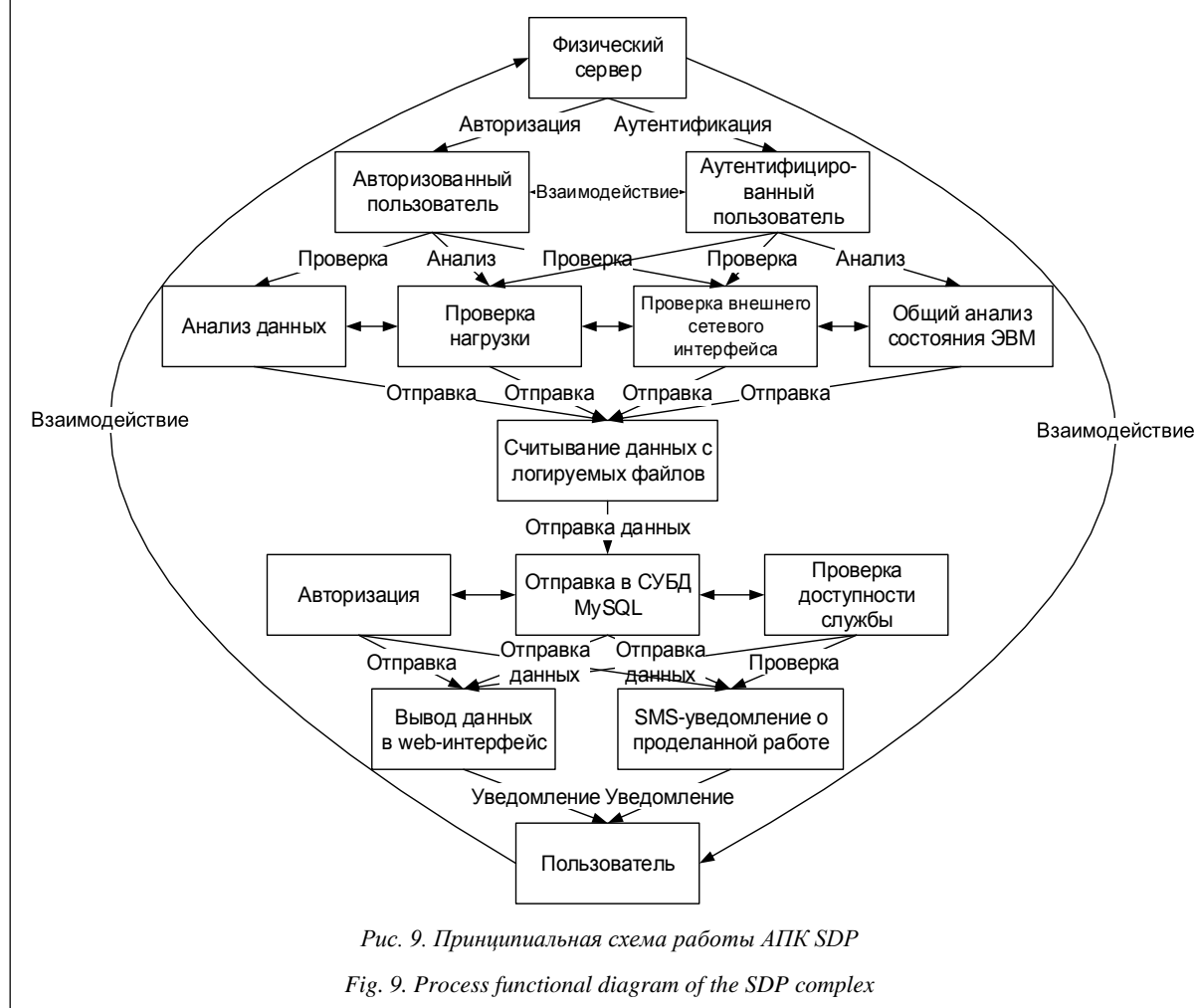


Рис. 9. Принципиальная схема работы АПК SDP

Fig. 9. Process functional diagram of the SDP complex

многопоточной обработки информации в автоматизированном режиме. Отличительной особенностью реализуемого АПК является наличие web-интерфейса и отправки полученных данных в СУБД MySQL. Структура и принципиальная схема работы SDP представлены на рисунках 8 и 9.

Все объекты, приведенные на рисунке 9, взаимодействуют друг с другом.

Канал передачи данных в СУБД MySQL расширен до 2,2 GB/s. Подобный эффект объясняется необходимостью логирования всех действий, в том числе и при сетевых атаках, в ходе которых происходит запись всех входящих и исходящих сетевых пакетов. В качестве передаваемых данных используются получаемые сведения о состоянии ЭВМ, входящие и исходящие сетевые пакеты (их количество, IP-адреса и т.п.), а также все происходящие на физическом сервере действия.

Разрабатываемый функционал и теоретическая нагрузка отражены в таблице 1.

Представим фрагмент исходного кода, реализованный на языке программирования BASH и предназначенный для прямого уведомления (на почтовый ящик) о нагрузке на центральный процессор (CPU) и основных запускаемых процессах:

```
#!/bin/bash
LOAD=`uptime | grep -o 'load average.*' | cut -c 15-18`
MAX_LOAD=$CPU
if [ $LOAD > $MAX_LOAD ]; then
ApacheThread=`ps -A|grep apache2|wc -l`
NginxThread=`ps -A|grep nginx|wc -l`
MysqldThread=`ps -A|grep mysql|wc -l`
SshThread=`ps -A|grep ssh|wc -l`
echo $LOAD | mail -s 'High load [SERVERNAME]teelxp@inbox.ru;`
fi
```

Фрагмент исходного кода, разработанный на языке программирования BASH, представляет со-

Таблица 1

Функционал SDP и его нагрузка на вычислительные ресурсы ЭВМ

Table 1

SDP functionality and its load on computing resources

Название функции	Цель функции	Теоретическая нагрузка, %	Лимит передачи данных, МВ/с	Результат функций
Анализ данных	Анализ поступающей с различных датчиков информации	0,03	450	Прием обрабатываемых данных в режиме автоматизации
Проверка нагрузки	Сверка нагрузки с заданной нормой и оптимизация вычислительных ресурсов	0,10	690	Сверка нагрузки с последующей оптимизацией вычислительных ресурсов ЭВМ
Проверка внешнего сетевого интерфейса	Проверка сетевой активности на аномалии	0,05	1110	Проверка на аномалии сетевого трафика и их устранение
Общий анализ состояния ЭВМ	Проверка работоспособности физического сервера	0,02	1650	Проверка работоспособности и уведомление пользователя
Считывание логируемых файлов	Считывание информации	0,01	2200	Обработка информации в автоматизированном режиме и отправка в MySQL

бой очистку логируемых файлов после какой-либо DoS- и DDoS-атаки:

```
#!/bin/bash
filename=ddos.log
cat /dev/null > $filename; echo "Очистка созданного файла"
tail /var/log/messages > $filename
echo "В $filename была произведена запись об отчетности прошедшей атаки"
```

Далее приведена часть исходного кода, созданная на языке программирования PHP и отвечающая за вызов функций в web-интерфейсе:

```
<?php
function microTimer_start()
{
    global $starttime;
    $mtime = microtime();
    $mtime = explode(' ', $mtime);
    $mtime = $mtime[1] + $mtime[0];
    $starttime = $mtime;
}
function microTimer_stop()
{
    global $starttime;
    $mtime = microtime();
    $mtime = explode(' ', $mtime);
    $mtime = $mtime[1] + $mtime[0];
    $endtime = $mtime;
    $totaltime = round( ($endtime - $starttime), 5 );
    return $totaltime;
}
function filters_input($check = 'all')
{
    $safehtml = new safehtml();
    $safehtml->protocolFiltering = "black";
    $safeinput = new safeinput();
    $safeinput->safeinput_check($check);
    unset($safehtml);
    unset($safeinput);
}
```

```
function filters_input_one($data, $type = "black", $nl2br =
= false, $LB_specchar = false)
{
    $safehtml = new safehtml();
    $safehtml->protocolFiltering = $type; // white или black
    $data = $safehtml->parse($data, $nl2br, $LB_specchar);
    unset($safehtml);
    return $data;
}
```

Фрагмент исходного кода, написанный на языке программирования высокого уровня C и отвечающий за распределение сетевой нагрузки по потокам:

```
bool execute_strict_cpu_affinity = false;
bool use_multiple_fanout_processes = true;
int main() {
    int fanout_group_id = getpid() & 0xffff;
    boost::thread speed_printer_thread( speed_printer );
    if(use_multiple_fanout_processes) {
        boost::thread_group packet_receiver_thread_group;
        unsigned int num_cpus = $CPU;
        for (int cpu = 0; cpu < num_cpus; cpu++) {
            boost::thread::attributes thread_attrs;
            if(execute_strict_cpu_affinity) {
                cpu_set_t current_cpu_set;
                int cpu_to_bind = cpu % num_cpus;
                CPU_ZERO(&current_cpu_set);
                CPU_SET(cpu_to_bind, &current_cpu_set);
                Int set_affinity_result = pthread_attr_setaffinity_np(thread_attrs.native_handle(), sizeof(cpu_set_t), &current_cpu_set);
                if (set_affinity_result != 0) {
                    printf("It is impossible to parallelize loading on physical kernels\n");
                }
            }
            packet_receiver_thread_group.add_thread(
                new boost::thread(thread_attrs, boost::bind(start_af_packet_capture, "eth6", fanout_group_id)
                );
        }
    }
}
```

Примеры SMS-уведомлений.

- В случае стабильной работы физического сервера (приходит ежечасное оповещение на мобильное устройство):

Дата: 10.08.2016;
 Время: 12:00;
 Размер потребляемого трафика на данный момент: 9 ТВ;
 Текущая нагрузка: 5 %, в пределах нормы;
 Состояние кластера стабильное.

- При каком-либо сбое ЭВМ, а также при наличии сетевой атаки:

Дата: 19.08.2016;
 Время: 05:35;
 Размер потребляемого трафика на данный момент: 34 ТВ;
 Текущая нагрузка: 75 %, выше нормы!
 Необходимо срочно принять меры!

Тестирование АПК

После подготовки алгоритма и написания кода АПК SDP необходимо произвести его тестирование для получения значений нагрузки на вычислительные ресурсы физического сервера в двух режимах – обычном и боевом. Обычный режим работы проявляется при упорядоченной (нормализованной) работе ЭВМ без воздействия внешних факторов. Под боевой работой подразумевается работоспособность ЭВМ с применением вредоносного сетевого трафика, направленного на нарушение удаленной работы серверного пространства.

Нагрузка на физические ресурсы ЭВМ при отсутствии каких-либо внешних факторов (обычный режим) представлена в таблицах 2 и 3.

В данных таблицах приняты следующие сокращения: А – количество обрабатываемой информации, МВ/с; В – количество потоков, шт./с; С – нагрузка при старте, %; D – нагрузка при рестарте, %; E – нагрузка при обработке информации, %; F – входящие сетевые пакеты, млн шт./с.

Средняя нагрузка из данных таблиц 2 и 3 показывает достаточно низкие результаты: центральный процессор – 0,77 %, SSD-накопитель – 0,53 %, оперативная память – 0,15 %.

Таблица 2

Потребление вычислительной мощности центрального процессора в течение десяти дней

Consumption of processing capacity by CPU during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	50	100	150	200	250	350	450	580	760	900
B	5	10	15	20	25	30	35	40	45	50
C	0,10	0,19	0,24	0,35	0,46	0,59	0,64	0,78	0,99	1,15
D	0,11	0,20	0,25	0,36	0,47	0,60	0,65	0,79	1,00	1,16
E	0,14	0,28	0,42	0,56	0,70	0,84	0,98	1,12	1,26	1,40
F	0,22	0,44	0,66	0,88	1,10	1,32	1,54	1,76	1,98	2,20

Таблица 3

Нагрузка на твердотельный накопитель (SSD) в течение десяти дней

Table 3

Load on a solid-state drive (SSD) during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	50	100	150	200	250	350	450	580	760	900
B	5	10	15	20	25	30	35	40	45	50
C	0,08	0,13	0,18	0,24	0,35	0,46	0,52	0,69	0,78	0,85
D	0,09	0,14	0,19	0,25	0,36	0,47	0,53	0,70	0,79	0,86
E	0,11	0,22	0,33	0,44	0,55	0,66	0,77	0,88	0,99	1,10
F	0,12	0,24	0,36	0,48	0,60	0,72	0,84	0,96	1,08	1,20

Полученные данные (табл. 2, 3) позволяют констатировать, что нагрузка на вычислительные ресурсы ЭВМ достаточно низкая и предоставляет возможность широкомасштабной обработки данных с последующей стабилизацией серверной рабочей среды.

Нагрузка на ресурсы физического сервера при интенсивной атаке вредоносным сетевым трафиком (DoS и DDoS) показана в таблицах 4 и 5.

Таблица 4

Нагрузка на центральный процессор в течение десяти дней

Table 4

CPU usage during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	0,1	0,3	0,5	0,8	1,0	1,2	1,4	1,6	1,9	2,2
B	20	40	60	80	100	120	140	160	180	200
C	2,0	4,0	6,0	8,0	10,0	12,0	14,0	16,0	18,0	19,0
D	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
E	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
F	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0
G	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0
H	4,0	5,0	6,0	7,0	8,0	9,0	10,0	11,0	12,0	13,0

Таблица 5

Нагрузка на SSD-накопитель в течение десяти дней

Table 5

Load on SSD-accumulator during 10 days

Показатель	День									
	1	2	3	4	5	6	7	8	9	10
A	0,1	0,3	0,5	0,8	1,0	1,2	1,4	1,6	1,9	2,2
B	20	40	60	80	100	120	140	160	180	200
C	2,0	4,0	6,0	8,0	10,0	12,0	14,0	16,0	18,0	19,0
D	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
E	0,55	0,6	0,7	0,9	1,18	1,45	1,67	1,88	2,03	2,06
F	0,56	0,7	0,8	1,0	1,19	1,46	1,68	1,89	2,04	2,07
G	0,8	1,6	2,4	3,2	4,0	4,8	5,6	6,4	7,2	8,0
H	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0

В данных таблицах приняты следующие обозначения: А – количество обрабатываемой информации, GB/s; В – количество потоков, шт./с; С – скорость атаки, GB/s; D – входящие сетевые пакеты, млн шт./с; Е – нагрузка при старте, %; F – нагрузка при рестарте, %; G – нагрузка при обработке информации, %; H – нагрузка при отправке данных в СУБД MySQL, %.

Средняя нагрузка (табл. 4, 5) остается достаточно низкой при многопоточной обработке информации большого объема: центральный процессор – 7,00 %, SSD-накопитель – 3,14 %, оперативная память – 0,75 %.

Результаты, приведенные в таблицах 4 и 5, свидетельствуют о высокой производительности АПК SDP. Таким образом, разработанный комплекс позволяет обрабатывать значительный объем информации и в режиме реального времени отправлять его в СУБД MySQL. Также появляется возможность запуска сложных вычислительных процессов без нарушения рабочей среды ЭВМ и при достаточно высокой производительности разработанного АПК.

Тестирование АПК SDP проводилось на следующем оборудовании: процессор 2xIntel Xeon 5670 (24 потока); оперативная память – 32 GB; винчестеры – RAID 10 (Intel S3710 SSDSC2BA012T401 800 GB каждый); суммарный сетевой канал – 5 GB/s; операционная система UBUNTU 14.04.

После апробации в течение одного месяца АПК SDP был окончательно инсталлирован на физический сервер в ЦОД (в г. Москва) на оборудование со следующими характеристиками: количество физических серверов – 30; процессор Intel Xeon 5660 (CPU – 60, физических ядер – 360, количество потоков – 720); оперативная память – 960 GB; винчестеры – RAID 10 (Intel S3710 SSDSC2BA012T401 800 GB каждый); внешний сетевой канал – 20 GB/s.

Заключение

В ходе проведенных исследований и разработок получены следующие результаты.

Разработан АПК по обработке потоковой информации в больших объемах. Это способствует совершенствованию пропускной способности между физическим сервером и реляционной БД MySQL.

Обоснована целесообразность применения АПК SDP на вычислительных ресурсах различной мощности. Предложена методика для упорядоченной и последовательной обработки информации на серверных вычислительных ресурсах.

Проведено множество тестирований, позволяющих определить эффективность работы АПК SDP. Средняя нагрузка на ресурсы центрального процессора варьируется от 0,77 % до 7,00 %. Данная тен-

денция предоставляет возможность запуска вычислительных процессов различного уровня, а также создает производительную рабочую среду, несмотря на большой поток обрабатываемой информации (до 2,2 GB/s). Загруженность твердотельного накопителя (SSD) колеблется в диапазоне от 0,53 до 3,14 %. Небольшая нагрузка на ячейки Solid-State Drive предоставляет возможность увеличения пропускной способности информационного канала передачи данных. Среднее потребление оперативной памяти составляет от 0,15 % до 0,75 %. Низкая нагрузка на вычислительные ресурсы позволяет выполнять скоростную обработку информации в автоматизированном режиме с последующим уведомлением (на выбор) системного администратора.

Таким образом, выявлена, проанализирована и исследована рабочая среда физического сервера с последующим переходом в кластер. Удобство в управлении и обработка информации на серверных вычислительных ресурсах позволили организовать стабильную работу ЭВМ.

Литература

1. Palchevsky E.V., Khalikov A.R. TCP/IP network STACK optimization under high load on UNIX-like systems. Proc. DSPTEch'2015. USATU Publ., 2015, vol. 1, pp. 130–135.
2. Пальчевский Е.В., Халиков А.Р. Техника инструментирования кода и оптимизация кодовых строк при моделировании фазовых переходов на языке C++ // Труды Ин-та системного программирования РАН. 2015. Т. 27. № 6. С. 87–96.
3. Пальчевский Е.В., Халиков А.Р. Равномерное распределение нагрузки аппаратно-программного ядра в UNIX-системах // Труды Ин-та системного программирования РАН. 2016. Т. 28. № 1. С. 93–102.
4. IBM a management system for the information business. White Plains, NY, IBM Publ., 1980.
5. Olifer V.G., Olifer N.A. Computer networks. Principles, technologies, protocols. St. Petersburg, 2007, 960 p.
6. Колисниченко Д. Linux. От новичка к профессионалу. 2-е изд. СПб: БХВ-Петербург, 2010. 764 с.
7. Crist E.F., Keijser J.J. Mastering OpenVPN. Birmingham – Mumbai, Puckt Publ., 2015, 364 p.
8. Ватаманюк А.К. Создание, обслуживание и администрирование сетей на 100 %. СПб: Питер, 2010. 350 с.
9. Dubrova E. Fault-Tolerant Design. Springer, 2013, 185 p.
10. Pargo C., Стивенс У. UNIX. Профессиональное программирование, 3-е изд. СПб, 2013. 1104 с.
11. Krylov V., Kravtsov K. DDoS attack and interception resistance IP fast hopping based protocol: 23rd Intern. Conf. on Software Engineering and Data Engineering, 2014, pp. 43–48.
12. Borodin A., Chernyshov A. The study of the network stack under the influence of static load. In the world of scientific discovery. 2014, vol. 60, no. 12.2, pp. 601–612.
13. Блум Р., Бреснахэн К. Командная строка Linux и сценарии оболочки. М.: Вильямс, 2013. 784 с.
14. Сокольников А.М. Сравнительный анализ подходов к разработке архитектуры и систем управления базами данных для высоконагруженных web-сервисов // Кибернетика и программирование. 2014. № 4. С. 1–13.
15. Leu F.-Y., Lin I-L. A DoS/DDoS attack detection system using chi-square statistic approach. Jour. of Systemics, Cybernetics and Informatics. 2010, vol. 8, no. 2, pp. 41–51.

AUTOMATED DATA PROCESSING SYSTEM IN UNIX-LIKE SYSTEMS

E.V. Palchevsky¹, *Postgraduate Student, teelxp@inbox.ru*

A.R. Khalikov¹, *Ph.D. (Physics and Mathematics), Associate Professor, khalikov.albert.r@gmail.com*

¹ *Ufa State Aviation Technical University, K. Marks St. 12, Ufa, 450000, Russian Federation*

Abstract. The article considers development of distributed and modular information processing in the automated mode. Implementation of this project allows accepting input and output data on a physical server up to 2,2 GB/s in volume while distributing stream information (all incoming network traffic on the server) on physical and logical kernels.

The paper shows a load dependence of physical resources on the input information. It also justifies feasibility of the developed hardware and software SDP (Speed data processing) use. There is also a structure and basic operation diagrams. The first stage of creating a complex is the development of an algorithm. The second stage includes its technical implementation. The article gives a fragment of a source code, which is responsible for notification messages on an email address as CPU load, and the main launched processes. There is a description of the main functionality with the following data: a function name, a function purpose, theoretical loading, data transfer limit (in MB/s) and a result of execution. The third stage includes testing SDP complex. Its average daily results are provided within ten days.

The created hardware and software complex allows processing input and output information effectively in an automatic mode in order to increase capacity of reception and sending data in MySQL DBMS, including cases with DoS- and the DDoS-attacks. One of the parts of the complex is WEB-module, which is responsible for control, both as from a personal computer, as well as from a mobile phone. The possibility of notifying about a physical server usage by sms is realized in a monitoring part of the WEB module.

The developed hardware and software complex showed high stability when handling data in large volumes with a minimum load on a computer.

Keywords: information processing, load on computing resources, data structure, lowering of information flow loading, information flows, SMS notification, distributed systems, DPC, data processing center.

References

1. Palchevsky E.V., Khalikov A.R. TCP/IP network STACK optimization under high load on UNIX-like systems. *Proc. DSPTech '2015*. UFA, USATU Publ., 2015, vol. 1, pp. 130–135.
2. Palchevsky E.V., Khalikov A.R. Technique the Instrumentation A Code And Optimization of Code Lines in Modeling Phase Transitions on the Programming Language C++. *Proc. Institute for system Programming of the RAS*. 2015, vol. 27, iss. 6, pp. 87–96 (in Russ.).
3. Palchevsky E.V., Khalikov A.R. Uniformly distributed load of hardware and software core in the UNIX-based systems. *Proc. Institute for system Programming of the RAS*. 2016, vol. 28, iss. 1, pp. 93–102 (in Russ.).
4. *IBM A Management System for the Information Business*. White Plains, NY, IBM Publ., 1980.
5. Olifer V.G., Olifer N.A. *Computer networks. Principles, technologies, protocols*. St. Petersburg, 2007, 960 p.
6. Kolisnichenko D. *Linux. Ot novichka k professionalu* [Linux. From a Beginner to a Professional]. 2nd ed., St. Petersburg, BHV-Peterburg Publ., 2010, 764 p.
7. Crist E.F., Keijser J.J. *Mastering OpenVPN*. Birmingham – Mumbai, Puckt Publ., 2015, 364 p.
8. Vatamanyuk A.K. *Sozдание, obsluzhivanie i administrirovanie setey na 100 %* [100 % Creation, Service and Administration of Networks]. St. Petersburg, Piter Publ., 2010, 350 p.
9. Dubrova E. *Fault-Tolerant Design*. Springer Publ., 2013, 185 p.
10. Rago S.A., Stevens W.R. *Advanced Programming in the UNIX Environment*. 3rd ed., Addison-Wesley Professional Publ., 2013, 1024 p. (Russ. ed.: St. Petersburg, 2013, 1104 p.).
11. Krylov V., Kravtsov K. DDoS attack and interception resistance IP fast hopping based protocol. *Proc. 23rd Int. Conf. on Software Engineering and Data Engineering*. 2014, pp. 43–48.
12. Borodin A., Chernyshov A. The study of the network stack under the influence of static load. *V mire nauchnykh otkryty* [In the World of Scientific Discoveries]. 2014, vol. 60, no. 12.2, pp. 601–612 (in Russ.).
13. Blum R., Bresnakhen K. *Linux Command Line and Shell Scripting Bible*. Wiley Publ., 816 p. (Russ.ed.: Villiams Publ., 2013, 784 p.).
14. Sokolnikov A.M. Comparative analysis of the approaches in development of the database management systems and its' architecture for highly loaded web-services. *Kibernetika i programmirovaniye* [Cybernetics and programming]. 2014, no. 4, pp. 1–13 (in Russ.).
15. Leu F.-Y., Lin I.-L. A DoS/DDoS Attack Detection System Using Chi-Square Statistic Approach. *Jour. of Systemics, Cybernetics and Informatics*. 2010, vol. 8, no. 2, pp. 41–51.