

УДК 681.518:681.3.016

О. М. Демиденко, А. В. Воруев, О. В. Быченко, В. А. Никишаев
Гомельский государственный университет им. Ф. Скорины
ул. Советская, 104, 246699 Гомель, Беларусь

Система мониторинга параметров вычислительного процесса и рабочей нагрузки на узел локальной вычислительной сети

Определена актуальность разработки системы мониторинга параметров вычислительного процесса (ВП) и рабочей нагрузки (РН) на узел ЛВС. Изложены принципы формализации ЛВС на четырех уровнях детализации ВП и РН. Обсуждены технология реализации функций диспетчера задач, особенности идентификации процессов, принципы реализации программы сбора статистики, подход к оценке технологических характеристик системы мониторинга (SYSMON). Показано, что SYSMON является ресурсоэкономным измерительным инструментом, слабо искажает динамику реализации ВП в ЛВС и может использоваться для периодического анализа операционной обстановки в ОС узлов ЛВС.

Ключевые слова: система мониторинга, параметры, вычислительный процесс, рабочая нагрузка, ресурсоэкономный инструмент.

Введение

Анализу вычислительного процесса (ВП) и рабочей нагрузки (РН) на вычислительную систему (ВС) посвящено много работ [1, 3]. Большинство из этих работ исследует ВС на основе средств мониторинга в ходе натурных [2] и имитационных экспериментов [4]. Однако, все методики исследований и средства их реализации ориентированы на устаревшую вычислительную технику (ВТ) и соответствующее ей программное обеспечение (ПО) организации обработки информации на ВС. При анализе организации обработки на локальных вычислительных сетях (ЛВС), реализованных на современной ВТ и использующих передовые информационные технологии (ИТ) обслуживания пользователей сети исследователь зачастую испытывает множество трудностей. Поэтому актуально создание средств мониторинга как при решении задач проектного моделирования ВП и ЛВС, так и для постановки натурных экспериментов (НЭ) при различных вариантах организации обработки информации на ЛВС. Исходя из актуальности проблемы

© О. М. Демиденко, А. В. Воруев, О. В. Быченко, В. А. Никишаев

была разработана система мониторинга параметров ВП и РН на узел ЛВС. Ниже излагаются состав и структура системы мониторинга, реализующей эти принципы формализации ВП и РН; технология использования системы мониторинга. Система мониторинга и технология ее использования были апробированы в условиях ЛВС Гомельского госуниверситета им. Ф. Скорины. В ходе апробации определены технологические характеристики, разработанной авторами системы мониторинга параметров ВП и РН.

Формализация вычислительного процесса и рабочей нагрузки на ЛВС

ЛВС представляется состоящей из традиционного состава компонентов, главной из которой является персональный компьютер (ПК), имеющий в качестве ОС Windows XXXX. При этом ВП в ПК представляется с высокой степенью детализации. Декомпозиция ПК на компоненты, соответствует аппаратным частям ПК: процессор (CPU), жесткий диск (HDD), видеоконтроллер (Video), сетевой адаптер (NET), оперативная память (Mem). Все эти компоненты считаются ресурсами ПК. ПО ПК находит свое отражение в алгоритмах функционирования компонент, реализованных в виде отдельных процессов. Объединение компонент ПК осуществляется на основе процессов, выполняющихся в контексте ОС.

Обычно мониторинг ВС проводился над отдельными составляющими аппаратной базы, что не позволяло получить общую картину системы в целом и влекло жесткую зависимость от особенностей внутренней архитектуры оборудования. Для устранения этого недостатка в мониторинг включаются только основные характеристики ЛВС, наиболее сильно влияющие на ход ВП. Поэтому концептуальная модель представляет собой упрощенное алгоритмическое отображение реальной системы. Сложная система представляется в виде многоуровневой конструкции взаимосвязанных элементов, объединенных в подсистемы различных уровней. При этом рассматриваются следующие аспекты отражения реальных процессов: последовательность состояний решаемой задачи, функционирование ОС, разделение ресурсов процессами. ВП в ЛВС рассматривается на четырех уровнях представления по их возможности декомпозиции на части и по связям между этими частями: сетевое взаимодействие, укрупненные программные модули, выполняемые функции и отображение процесса использования ресурсов системы.

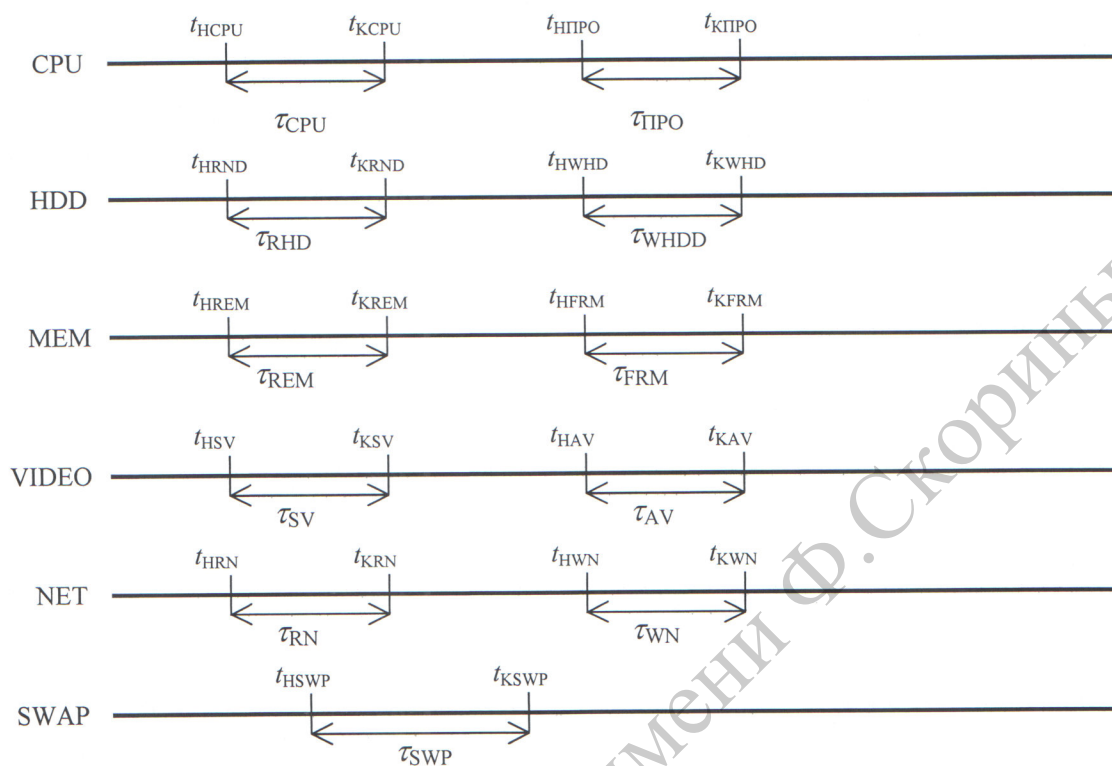
Сетевое взаимодействие представляется в виде отдельных составляющих (узлов ЛВС), которые взаимодействуют через общее устройство обслуживания (среду передачи данных). К функциям узла относятся обслуживание внутренних запросов (диалоговых и фоновых соответственно с интенсивностями λ_D и λ_Φ) и запросов поступающих извне (транзитные с интенсивностью λ_T). Отличие внешних запросов от внутренних заключается только в их маршруте продвижения. Рождаясь в обычном узле, они проходят через сеть, обслуживаются на узле-сервере и вновь возвращаются через сеть на узел-родитель. Все функции сборки-разборки пакетов вынесены за границы узла в системы передачи данных. Поэтому считается, что узел передает в сеть целый запрос, затрачивая на это время τ_{WN} и получает из сети ответ также целым, через интервал времени τ_{RN} . При организации монито-

ринга взаимодействия узлов ЛВС основное внимание необходимо было уделить топологии сети, методу доступа к среде передачи данных; скорости передачи информации по сети. Все эти компоненты ЛВС соответствуют физическому и каналному уровням эталонной модели взаимодействия открытых систем. Выбор был сделан на сети Ethernet, имеющей скорость передачи 10 Мбит/с и шинную топологию (звезда-шина). В рамках этого выбора ЛВС представляется как взаимодействие равноправных узлов сети, объединенных общей средой передачи данных. При этом любой из узлов ЛВС может являться как сервером, так и клиентом в ЛВС. Система мониторинга отслеживает только запросы на сетевое обслуживание дисков, поскольку большинство операций по обслуживанию запросов, пришедших из сети, используют одни и те же функции ОС, что и локальные запросы. Поэтому идентифицировать и отделить их от процессов собственной ОС практически невозможно. С другой стороны все выполняемые процессы используют память на жестком диске и операции обращения к диску можно идентифицировать, и уже по ним определяется является ли запрос сетевым или канальным для данного узла ЛВС.

На уровне **укрупненных программных модулей** каждому программному модулю ($ПМ_j$) выделяется совокупность ресурсов для его выполнения, что определяет времена жизни программных модулей $T_{жж}$ от момента зарождения заказа на него ($t_{нж}$) до окончания его обслуживания. Для представления информационной базы данных (ИБД), которая может быть и распределенной по сети, используется граф базы данных (GRB), узлами которого являются модули ИБД, а дуги между ними определяют связи между модулями ИБД. Структура этих связей определяется матрицей связей ($M(Snk)$). Другим примером укрупненного модуля являются части ОС, обслуживающие определенные группы либо запросов пользователей, либо сигналы оборудования ЛВС. Мониторинг взаимодействия программных компонент в ОС Windows осуществляется путем фиксации моментов передач управления и порождения процессов. Процессы обладают правами пользования всеми ресурсами системы и не обладают информацией и параллельно действующих пользовательских процессах. Этим достигается защита от воздействия процессов друг на друга. Вычислительный процесс в многозадачной ВС состоит из периодических передач ОС прав пользования ресурсами системы от процесса к процессу. На рис. 1 представлена структура временных интервалов обслуживания процессов ресурсами ЛВС, фиксируемых системой мониторинга.

На уровне **выполняемых функций** каждый $ПМ_j$ разбивается на последовательность единичных функциональных действий $\tau_{0фj}$, каждое из которых представляет собой группу элементарных операций на уровне запросов к основным компонентам ЛВС, что в сумме определяет единичный интервал времени $\tau_{0фj}$. Каждому классу запросов присваивается соответствующая комбинация элементарных операций.

Уровень **отображения процесса использования ресурсов** системы позволяет получить коэффициенты использования $ПМ_j$ следующих ресурсов системы: процессоры (η_{CPU}), дисковой памяти (η_{HDD}), общей памяти (η_{MEM}), видео-системы (η_{VIDEO}), систем передачи данных (η_{NET}). Сюда также относятся отображение динамики использования ресурсов ЛВС при конкуренции $ПМ_j$. Для всех типов ресурсов ЛВС определяются следующие основные характеристики дина-



τ_{CPU} — интервал обслуживания процесса устройством CPU
 $\tau_{ПРО}$ — простой ВП узла ЛВС
 τ_{RHD} — чтение с устройства HDD
 τ_{WHDD} — запись на устройство HDD
 τ_{REM} — резервирование блока памяти на устройстве MEM
 τ_{FRM} — освобождение блока памяти на устройстве MEM

τ_{SV} — операция синхронного видео на устройстве VIDEO
 τ_{AV} — операция асинхронного видео на устройстве VIDEO
 τ_{RN} — операция сетевого чтения через устройство NET
 τ_{WN} — операция сетевой записи через устройство NET
 τ_{SWP} — свопинговые операции

Рис. 1. Структура временных интервалов обслуживания процессов ресурсами ВС.

мики их перехода из состояния в состояние: вероятность перехода процесса k с ресурса i на ресурс l (P_{kil} , $i \neq j$, $i, l = CPU \div Net$); загруженность ресурса (η_i), среднее время работы процесса k на ресурсе i (t_{ki}). ПО представляется в виде совокупности процессов. Различаем два типа процессов: пользовательские и системные. Первые генерируются по запросам РН на ЛВС. Системные же процессы генерируются ОС для ее нужд. Любая пользовательская программа может отображаться одним или несколькими процессами, которые монопольно захватывают в свое распоряжение ресурс. Поскольку многозадачность и распараллеливание процессов в ЛВС реализуется на основе квантования времени использования процессами ре-

сурсов системы, то при мониторинге важно определить закон распределения кванта времени ($t_{кв}$), являющегося функцией частоты смен состояний ЛВС $n(t)$. ОС представляется совокупностью системных процессов (1-го уровня — System, 2-го уровня — программы системной обработки, например, Explorer). Для системных процессов определяется частота вызовов системных процессов первого уровня (n_{SYS1}), используемая в дальнейшем для анализа текущей операционной обстановки.

Взаимодействие компонентов ЛВС представляется выполнением процессов на ресурсах ЛВС в условиях их конкуренции за эти ресурсы. Параметрами этого взаимодействия являются:

- матрица средних времен выполнения процесса k на ресурсе i ($M(t_{ki})$);
- матрица вероятностей переходов процесса по ресурсам ЛВС $M(P_{kij})$.

С помощью этих двух параметров появляется возможность полумарковского представления динамики использования ресурсов ЛВС. Для уточнения особенностей динамики использования ресурсов предполагается, что система мониторинга позволяет определить полное время операции над ресурсом (t_{ki}^{Π}) и точное время использования ресурса (t_{ki}^{τ}), в котором отсутствуют времена работы процессов System.

Различаем следующие классы РН: дистанционная пакетная (RB) и режима разделения времени (TS). При дистанционной пакетной РН фиксируется время ожидания запроса по удаленной линии связи ($\tau_{обд}$). РН характеризуется объемом работ, создаваемых совокупностью элементов работы. Под элементами работы понимается либо выполнение пользовательского процесса на ресурсе i , либо взаимодействие пользовательских и системных процессов с использованием ресурсов ЛВС. В качестве меры объема работ выступает количество ресурса i , необходимое для выполнения элемента работы $Z(Q_{iz}^p)$. Элементами работы Z считаем задачи пользователей в составе запросов пользователя. Поэтому РН на ЛВС представляют собой совокупность задач и данных, которые обрабатываются ВС в течение некоторого промежутка времени. Любая задача во время ее выполнения порождает последовательность запросов на использование различных ресурсов ЛВС. Атрибутами таких запросов являются: тип используемого ресурса i , t_{ki} , момент поступления запроса (t_k^{np}), объем ресурса Q_i . Все запросы реализуются через посредника — ОС (под ОС подразумевается весь спектр ОС Windows).

Состав и структура системы мониторинга

Структура системы мониторинга (SYSMON) определяется необходимостью реализации следующих функций: отслеживание системных событий, сопоставление этих операций процессам, периодическая запись статистики измерений на диск. Системными событиями считаем: переключение потоков, обращение к диску, использование виртуальной памяти, графические операции. SYSMON состоит из трех модулей: перехвата системных событий (MS1), идентификации процессов (MS2), сбора статистики (MS3). MS1 представляет собой драйвер, написанный на языке C++ с использованием библиотеки VToolsD. Он перехватывает управление у

следующих системных операций: переключение потоков, файловые операции, использование виртуальной памяти, графические операции. Для хранения информации драйверу выделяется в невыгружаемой памяти буфер емкостью 2048 событий. Поскольку скорость заполнения этого буфера может изменяться от 5 до 1500 событий в секунду, то модуль MS3 должен своевременно считывать статистику из этого буфера. MS2 представляет собой динамически загружаемую библиотеку, содержащую системную «ловушку». Использование «ловушки» позволяет загрузить библиотеку в адресное пространство каждого процесса. Во время инициализации библиотеки вызывается функция получения системного идентификатора процесса, а затем этот идентификатор отсылается MS3. MS3 представляет собой программу, написанную на языке Borland Delphi 5.0. Он выполняет следующие действия: загружает драйвер перехвата системных событий; регистрирует в системе «ловушку», содержащую модуль идентификации процессов; создает файл журнала событий с уникальным именем, запускает отдельный программный поток, отвечающий за считывание данных из драйвера и записи их в журнал; отображает промежуточные данные на дисплее.

Реализация функции отслеживания работы диспетчера задач

ОС Windows для организации параллельного выполнения процессов применяет «вытесняющую многозадачность». Когда ОС Windows планирует использование CPU, она распределяет его между потоками, а не между приложениями. Поэтому потоки являются объектами, получающими в свое распоряжение квант ресурса CPU. Как минимум, каждый процесс имеет хотя бы один поток, но ОС Windows позволяет инициировать в рамках процесса сколько угодно потоков. Интерфейс Win32API позволяет управлять распределением времени CPU между потоками. ОС планирует выделение CPU в соответствии с приоритетами потоков. Когда поток создается, ему назначается приоритет, соответствующий приоритету породившего его процесса. В свою очередь, процессы могут иметь следующие классы приоритетов: реального времени (Realtime), высокий (High), нормальный (Normal), фоновый (Idle). SYSMON отслеживает моменты переключения между потоками, осуществляемые диспетчером задач. Для этого используется VMM-сервис (Call When Thread Switched). Этому сервису передается указатель на функцию обратного вызова (Thread Switch Call Back или Old Thread New Thread). Диспетчер задач вызывает эту функцию при переключении потоков, которая выполняет следующие действия: фиксирует время переключения; заносит в журнал запись о завершении кванта времени работы CPU для старого потока; вычисляет идентификатор текущего процесса; заносит в журнал запись о начале кванта CPU нового потока. Как видим, монитор внедряет свои «перехватчики» на основных информационных магистралях узла ЛВС. Данные, собираемые этими перехватчиками, характеризуют специализированные операции.

Особенности программы идентификации процессов

Поскольку вертикальные уровни ОС изолированы друг от друга, то они не знают ничего о строении внутренних структур друг друга. Взаимодействие между

ними осуществляется через документированные интерфейсы. Для обеспечения совместимости и расширяемости функций нижние уровни не должны знать о существовании верхних уровней. Поэтому при разработке алгоритмов мониторинга возникла проблема, связанная с идентификацией процесса, выполнившего какую-либо из отслеживаемых операций, которую необходимо было решить при написании драйвера SYSMON. Драйвер может определить идентификаторы «нити» и процесса, но не может определить имя процесса. Можно было написать программу, использующую библиотеку ToolHelp, которая перечисляет все «нити» и процессы в системе. При этом получаемые ею идентификаторы не будут совпадать со значениями, полученными драйвером. Это обусловлено тем, что идентификаторы уровня ядра ОС отличаются от идентификаторов пользовательского режима Win32. Поэтому необходимо было найти соответствие между этими идентификаторами. Для решения этой проблемы в SYSMON реализован следующий способ. Драйвер предоставляет пользовательским процессам «сервис», возвращающий текущие системные идентификаторы «нити» и процесса. Но для того, чтобы найти соответствия идентификаторов для всех процессов, надо было выполнить данную процедуру в контексте каждого процесса. Попасть в контекст каждого процесса можно с помощью использования глобальных системных «ловушек». «Ловушки» хранятся в динамически загружаемых библиотеках и после регистрации в системе подключаются к каждому выполняемому процессу. В данном случае функция «ловушки» может не выполнять никаких действий, то есть в ее функции входит просто возвращение управления. Отметим, что эти действия должны выполняться в моменты инициализации и завершения работы библиотеки.

Итак, во время инициализации «ловушки» выполняются следующие действия: загрузка драйвера SYSMON; вызов функции получения системных идентификаторов «нити» и процесса; определение имени процесса; нахождение и получение дескриптора окна программы сбора статистики (MS3); отправка этому окну сообщения о создании процесса, содержащего сведения об искомом процессе. Кроме того, во время завершения работы библиотеки выполняются следующие действия: нахождение и получение дескриптора окна MS3; отправка этому окну сообщения о завершении работы процесса.

Схема внедрения программ идентификации процесса в структуру «нормального» выполнения процесса приведена на рис. 2. Здесь пунктирной линией выделены блоки, внедряемые в «нормальную» работу системы с целью идентификации процесса. Время жизни процесса состоит из четырех больших составляющих: создание, запуск, работа, завершение работы процесса. Как видно из рис. 2, система идентификации внедряется только в процедуры запуска процесса и его завершения, а на саму работу процесса MS2 влияния не оказывает.

Обработка событий реализуется следующей последовательностью действий: работа пользовательского процесса, перехват запроса, его идентификация, фиксация события в буфер, проверка заполнения локального буфера. При заполнении буфера осуществляется сжатие информации и запись ее на диск, а затем возврат управления обработчику процессов.

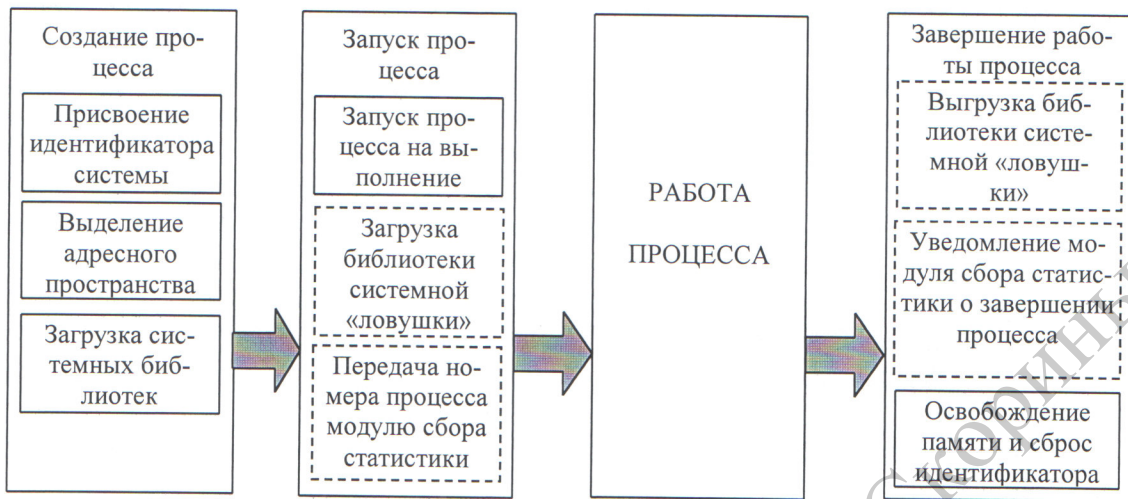


Рис. 2. Схема внедрения программы идентификации процессов в структуру «нормального» выполнения процесса.

Принципы реализации программы сбора статистики

Драйвер SYSMON записывает информацию обо всех отслеживаемых событиях в свой локальный буфер, размер которого ограничен 4 тысячами записей. При этом MS3 выполняет следующие действия: запуск драйвера SYSMON; периодическое считывание данных из буфера; идентификация процессов в системе, запись последовательности событий в журнал. MS3 взаимодействует с MS1 и MS2. Основной модуль MS3 работает постоянно в фоновом режиме. При инициализации MS3 осуществляется две операции: загрузка драйвера SYSMON; загрузка модуля идентификации процессов. Сам процесс сбора статистики включает в себя реализацию трех основных функций: обработка событий запуска и завершения процессов, поступающих от модуля идентификации процессов; считывание данных о системных событиях из буфера драйвера SYSMON; запись журнала событий на диск. Завершение работы MS3 реализуется двумя операциями: выгрузка драйвера SYSMON; выгрузка модуля идентификации процессов.

MS3 является основным «запускаемым» модулем SYSMON и устанавливается в меню автозагрузки ОС Windows. По умолчанию она не выводит на экран никаких дополнительных сведений, кроме списка включенных процессов и количества событий в секунду. Можно установить режим вывода списка событий на экран. По завершении работы MS3 происходит выгрузка драйвера из памяти, закрытие файла журнала и прекращение отслеживания системных событий. Периодическое считывание данных из локального буфера реализовано в виде цикла, функционирующего в отдельном потоке с максимальным приоритетом. Обновление данных происходит каждую секунду. Изменение интервала включения этой процедуры реализуется с помощью функции ожидания Sleep, а точность его представления обеспечивается приоритетом реализации этого потока (в реальном времени). Поскольку процедура считывания выполняется быстро, а поток в течение только

одной секунды до следующего события простаивает, то эта операция не оказывает существенного влияния на диаграмму работы системы в целом.

MS3 при запуске регистрирует появление в системе «ловушки», отвечающей за идентификацию процессов в системе. Затем при запуске нового процесса MS3 получает все сведения о нем, включая его системные идентификаторы. По этим сведениям строится таблица соответствия идентификаторов. После каждого считывания данных из драйвера происходит преобразование идентификаторов с помощью таблицы соответствия. Так как записывать данные в журнал на диске после каждого считывания ресурсоемко, то эти данные вначале накапливаются в буфере MS3, который в десять раз больше буфера драйвера. В дальнейшем запись данных в журнал происходит только после заполнения этого буфера.

Оценка технологических характеристик системы мониторинга

Апробация SYSMON и оценка ее технологических характеристик проводилась в Гомельском государственном университете им. Ф. Скорины. Для проведения НЭ использовалось 13 вычислительных комплексов различных конфигураций. Все компьютеры были объединены в общую ЛВС в рамках одного домена сети Ethernet. Суммарное время сбора статистики при исследовании ВП в узле ЛВС в ходе НЭ составило 300 часов машинного времени. В результате серии НЭ были вычислены накладные расходы на мониторинг ЛВС, которые выражаются в потреблении вычислительных ресурсов системы. Исследования проводились сеансами трех типов:

— короткие (до 100 с), когда SYSMON «разгоняется» и «сворачивается» в течении одного сеанса работы пользователей ЛВС, поэтому процент потребления ресурсов в этом случае существенно выше среднего;

— средние (от 100 с до 10000 с), обеспечивающие основной объем статистики ВП и РН на ЛВС;

— длинные (свыше 10000 с).

Величина **накладных расходов**, вычисленная по средним значениям для серий НЭ составила $\Delta_{\text{SYSMON}} = 0,0186$ при диапазоне ее изменения от 0,0022 до 0,3639. **Информативность** мониторинга изменялась в диапазоне от 0,4859 до 0,9989 при среднем значении $I_{\text{SYSMON}} = 0,9019$. Анализ статистики НЭ показал, что SYSMON обладает следующими технологическими характеристиками:

— **средний процент потребления** всех ресурсов ЛВС SYSMON равен $\Delta_{\text{SYSMON}} = 1,9\%$;

— **средний коэффициент загрузки CPU** на нужды мониторинга равен $\eta_{\text{CPU M}} = 1,8\%$;

— **средний коэффициент загрузки HDD SYSMON** равен $\eta_{\text{HDD M}} = 7\%$;

— **размер потребляемой SYSMON оперативной памяти** оказался равным $V_{\text{MEM SYSMON}} = 256 \text{ Кб}$;

— **максимальная скорость потребления дисковой памяти** для хранения журнала SYSMON составляет 10 Мб/час;

— **информативность SYSMON** оказалась равной $I_{\text{SYSMON}} = 90\%$.

Заключение

SYSMON является открытой для пополнения ее функциональных возможностей. Увеличение уровня детализации ВП и РН на узел ЛВС приведет к появлению новых подсистем. Такая работа ведется коллективом разработчиков SYSMON. Поэтому проведенное описание состава и возможностей следует рассматривать как первую версию реализации мониторинга на ЛВС. Неполно отражены возможности регистрации SYSMON сетевых аспектов организации обработки в ЛВС. Но даже первая версия позволяет регистрировать многие аспекты взаимодействия узлов ЛВС. Поэтому, с нашей точки зрения, SYSMON обладает хорошими технологическими характеристиками, является ресурсоэкономной и слабо искажает динамику ВП в ЛВС. Это позволяет использовать SYSMON в условиях реальной работы ЛВС, периодически наблюдая за характеристиками ВП и РН на ЛВС и, в случае их ухудшения, указывать на необходимость адаптации ВП под изменившуюся РН на ЛВС.

1. Додонов А.Г. Анализ отраслевых информационно-вычислительных сетей. — Ленинград: Судостроение, 1990. — 360 с.
2. Демуськов А.Б. Организация мониторинга параметров вычислительного процесса в комплексах и сетях ЭВМ при натуральных и имитационных экспериментах: Дисс... канд. техн. наук. — Гомель, 1999. — 207 с.
3. Зайченко Е.Ю. Анализ и синтез структуры глобальных вычислительных сетей. — К.: ЗАО «Укрспецмонтажпроект». — 1998. — 150 с.
4. Филлипс Д., Гарсиа-Диаз А. Методы анализа сетей. — М.: Радио и связь, 1984. — 496 с.

Поступила в редакцию 29.05.2001