

# ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

И.В. Комаров

Санкт-Петербургский институт информатики и автоматизации РАН  
199178, Санкт-Петербург, 14-я линия ВО, д. 39  
<mur@linux-ink.ru>

---

УДК 681.3

*И. В. Комаров. Оценка производительности вычислительных систем // Труды СПИИРАН, Вып. 2, т. 2. — СПб.: Наука, 2005.*

**Аннотация.** В статье дан анализ существующих методов оценки производительности вычислительных систем. Рассмотрены основные компьютерные подсистемы, выбраны характеристики и критерии для оценки производительности. Предложен новый метод оценки, основанный на анализе значений характеристик и свертке этих значений в интегральный показатель. Рассмотрены такие характеристики поведения системы, как эффективность, стабильность и доминирование нагрузки. — Библ. 8 назв.

UDC 681.3

*I. V. Komarov. Performance Evaluation of Computation Systems // SPIIRAS Proceedings. Issue 2, vol. 2. — SPb.: Nauka, 2005.*

**Abstract.** The analysis of existing methods for performance evaluation of computing systems is given. The basic computer subsystems are considered, characteristics and criteria for performance evaluation are chosen. The new method of performance based on the analysis of values of characteristics and convolution of these values in an integrated parameter is offered. Such characteristics of systems behaviour, as efficiency, stability and domination of loading are considered. — Bibl. 8 items.

---

## 1. Введение

Оценка производительности сложная, но необходимая задача как при исследовании работающих, так и при проектировании новых вычислительных систем. Для работающих систем она позволяет выявить и устранить "узкие" места (подсистемы или компоненты системы, на которые приходится максимальная нагрузка), тем самым, повысив надежность работы. Для вновь проектируемых, она помогает выбрать необходимое оборудование, соответствующее планируемым задачам, рассчитать рабочую нагрузку, при которой система будет работать наиболее эффективно.

Оценка производительности особенно актуальна для высокопроизводительных (MPI, GRID) и кластерных систем высокой готовности со сложной архитектурой и топологией, то есть для систем, где приложения могут распределяться между узлами или мигрировать с одного узла на другой.

В 80-ые годы для оценки производительности вычислительных систем использовали метод Гибсона, но он отражает только быстродействие оборудования и неприменим к оценке качества выполнения задач и, кроме того, оставляет открытым выбор коэффициентов для "смесей" [1]. В настоящее время используется несколько методов оценки производительности систем. Наилучшим из них является испытание системы при реальной рабочей нагрузке, однако в случаях, когда этот подход оказывается неприемлемым [2], прибегают к эталонному тестированию — методу определения эффективности системы при определенных (эталонных) нагрузках. Эталонные тесты (benchmarks) можно разделить на две категории: широкого (оценивают эффективность системы в целом по ряду критериев: например SPEC, TPC) и узкого применения (оценивают эффективность

компьютерных сервисов или системных компонентов: например Linpack, I/O zone, Webstone) [3].

Из-за многообразия компьютерных архитектур, операционных систем и различия решаемых задач, на сегодняшний день отсутствуют универсальные методы оценки производительности. В любом случае, она опирается на специфику архитектуры компьютера, где каждый системный компонент (процессор, память, шины и т.д.) имеет свое выражение в переменных и структурах ядра операционной системы. Набор значений этих переменных, каждая из которых отражает состояние одной из подсистем, характеризует состояние системы в определенный момент времени. Исследуя изменение этих характеристик, можно определить ее поведение на любом временном интервале. Ниже предлагается метод оценки производительности вычислительной системы, основанный на выборе этих переменных, их предельных значений и весовых коэффициентов с последующей сверткой этих значений в интегральный показатель.

## 2. Оценка компьютерных подсистем

Выберем необходимые характеристики и критерии для оценки производительности вычислительной системы, работающей под управлением ОС UNIX SysV.

### 2.1 Процессорная подсистема

Процессорная подсистема, прежде всего, характеризуется показателями загруженности (процентом утилизации) центрального процессора (ЦП): %usr (процент времени на выполнение пользовательских задач), %sys (процент системного времени), %wio (процент времени на ожидание ввода-вывода блочных устройств), %idle (процент времени простоя ЦП). Очевидно, что суммарная утилизация ЦП не превышает 100%:  $\%usr + \%wio + \%sys + \%idle = 100\%$ . Для нормально функционирующей системы, как правило, должны выполняться условия [7]:  $\%wio \leq 30\%$ ,  $\%usr + \%wio + \%sys \leq 70\%$ ,  $\%sys \leq 30\%$ . Показатель %sys может меняться в широких пределах, от типовых 10%-30%, до 50% и более. Это связано с тем, что некоторые сетевые сервисы (например, NFS) являются частью ядра, и любая активность NFS относится не к показателю пользовательской загрузки, а к показателю %sys, следовательно, его нельзя использовать для оценки производительности. Показатель загрузки %wio не должен превышать 30%, в противном случае можно с уверенностью говорить о низкой пропускной способности (а следовательно и производительности) системы ввода-вывода. Суммарный процент утилизации в нормально загруженной системе не должен превышать 70%-80%, так как при увеличении загрузки ЦП увеличивается очередь процессов, ждущих обслуживания, а значит и время их выполнения. Длина очереди процессов определяется характеристикой runq-sz и является более точным показателем загруженности ЦП, чем суммарная утилизация [4]. Большое значение runq-sz указывает на то, что процессор не успевает обслуживать прикладные процессы, то есть для нормальной работы системы необходимо выполнение условия:  $runq - sz \leq 4$ .

Дополнительно можно упомянуть параметры, показывающие частоту системных вызовов, выполняемых ЦП. В UNIX-системах есть возможность разделения общей статистики системных вызовов (scall/s) на вызовы чтения/записи

(sread/s, swrit/s), отражающие активность ввода-вывода, и fork/exec-вызовы (fork/s, exec/s), характерные для командных интерпретаторов. По количеству системных вызовов сложно судить о производительности, но можно получить общее представление об уровне активности и типе исследуемой системы.

## 2.2 Память

При оценке работы памяти, казалось бы, лучше всего ориентироваться на значения свободной и занятой (используемой) памяти. Однако, это невозможно, потому что практически вся память при загрузке операционной системы разбивается на страницы и, далее, попадает в свободный список, отводится под файловый КЭШ или резервируется ядром [2]. Поэтому значения параметров свободной и используемой памяти не отражает реальные процессы и при исследовании производительности они бесполезны.

Для определения характеристик памяти, действительно важных при оценке ее производительности, необходимо рассмотреть механизмы ее работы. В случае, когда все приложения (включая и ядро) одновременно запрашивают объем памяти, который превышает доступный, системе приходится прибегать к механизму подкачки (записи неиспользуемых страниц памяти на диск). В случае, если ситуация с нехваткой памяти усугубляется, на диск записываются все участки памяти, используемые процессом или процессами. Этот механизм называется свопингом. В современных UNIX-системах механизм свопинга начинается только в случае критической нехватки памяти, поэтому все параметры, показывающие его активность важны при оценке производительности системы. Достаточно часто встречаются ситуации, при которых производительность резко падает, но свопинг при этом не начинается. Такое “пограничное” поведение системы возможно при определенной, но некритичной нехватке памяти. В подобных случаях ядро включает механизм страничного сканирования, который пытается освободить страницы памяти с наибольшим временем простоя. Этому механизму в UNIX соответствует параметр частоты сканирования страниц. Активность механизма страничного сканирования может значительно снижать производительность системы.

Загрузку памяти представляют следующие характеристики: rpgin/s, rpgout/s, pgfree/s – соответственно число загруженных, выгруженных и освобожденных страниц в секунду; pgscan/s – частота сканирования страниц для возможного их освобождения; swpot/s, swpin/s – соответственно число процессов в секунду выгруженных и загруженных в своп; swp-sz — длина очереди процессов, выгруженных в своп. Характеристики rpgin/s, rpgout/s, pgfree/s не говорят о качественном состоянии системы, по их изменению можно судить только об уровне активности. Фактически, высокие значения этих характеристик могут говорить об активном вводе-выводе, который реализован через страничную подкачку. В основном, нехватку или достаточность памяти можно определить, основываясь на значениях частоты сканирования (pgscan/s) и активности свопинга (swpot/s, swpin/s, swp-sz). При достаточном количестве памяти в ОС Solaris 8 все эти параметры всегда должны иметь нулевые значения. В ОС Solaris 7 реализован несколько другой механизм сканирования страниц, поэтому значение pgscan/s может быть ненулевым (но не превышать 250) [8]. Кроме того, необходимо добавить ряд важных параметров, показывающих производительность буферного КЭШа, которые тоже должны удовлетворять определенным требованиям:

$\%gcache$  (частота попадания пользовательских запросов на чтение в КЭШ),  $\%wcache$  (частота попадания пользовательских запросов на запись в КЭШ).

Сказанное позволяет сделать предварительные выводы о том, что для нормальной работы подсистемы памяти необходимо выполнение следующих условий:  $pgscan/s \sim 0$ ,  $swp - sz \sim 0$ ,  $\%rcache > 90\%$ ,  $\%wcache > 60\%$

## 2.3 Система ввода-вывода

Система ввода-вывода, прежде всего, отождествляется с активностью блочных устройств: дисками, массивами, лентами и т.д. Частично к ней можно отнести параметры, характеризующие состояние файловой системы и КЭШа. Каждому блочному устройству соответствует ряд параметров, отражающих его работу:  $r+w/s$  — число операций чтения или записи, выполняемых в секунду;  $\%busy$  — доля времени в процентах, когда устройство было чем-то занято;  $await$ ,  $avserv$  — соответственно среднее время ожидания и выполнения запроса, затраченное устройством, в миллисекундах;  $\%ufsipf$  — процент времени, на которое из списка свободной памяти брался индексный дескриптор с ассоциированными повторно используемыми страницами.

Отдельно необходимо описать параметр  $\%ufsipf$ , который можно отнести также и к подсистеме памяти. Он отображает весьма важную информацию, которая показывает, насколько часто ядру приходилось повторно обращаться к неактивному индексному дескриптору, у которого были ассоциированные с ним страницу в буферном КЭШе. Ненулевое значение параметра, как правило, является признаком того, что таблица  $inode$  слишком мала, и поэтому полезные данные из буферного КЭШа приходится выбрасывать [4].

Из перечисленных характеристик только число операций чтения/записи ( $r+w/s$ ) сложно интерпретируемо и отражает лишь активность устройства подсистемы, остальные параметры должны удовлетворять следующим требованиям:  $\%busy \leq 3\%$ ,  $await + avserv \leq 50ms$ ,  $\%ufsipf \sim 0$

## 3. Интегральная оценка состояния системы

Для оценки производительности вычислительной системы при ее эксплуатации целесообразно пользоваться методом сводных показателей, суть которого состоит в "свертке" многих оценок сложного объекта в единую, интегрированную оценку. При построении сводного показателя учитываются как оценки отдельных характеристик объектов, так и информация об относительной весомости отдельных характеристик, то есть степени влияния оценок по скалярным критериям на сводную оценку в целом [5].

Сформируем вектор исходных характеристик, достаточных для оценивания производительности:

$X = (runq - sz, \%wio, pgscan/s, swp - sz, \%ufsipf, \%busy, await + avserv, \%rcache, \%wcache)$

Сгруппировав все требования, предъявляемые к характеристикам, получим следующую систему оценочных критериев, разделенных на классы соответствия подсистемам:

$$\begin{cases} runq - sz \leq 4 \\ \%wio < 30\% \end{cases} \begin{cases} pgscan/s \sim 0 \\ swp - sz \sim 0 \\ \%rcache > 90\% \\ \%wcache > 60\% \end{cases} \begin{cases} \%busy \leq 3 \\ await + avserv \leq 50ms \\ \%ufsipf \sim 0 \end{cases}$$

В качестве примера, рассмотрим применение критериев для четырехпроцессорной системы под управлением ОС SOLARIS 8, предназначенной для обработки запросов к базе данных (активный узел кластера высокой готовности), работающей в крупном банке и обслуживающей сотни клиентов. Сбор статистических данных о заданных характеристиках проводился в течение одного рабочего дня с периодичностью 20 секунд средствами анализатора системной активности SAR. Далее, исходя из предварительной оценки измерений, были выбраны два временных интервала: интервал, в течение которого система работала устойчиво, и интервал пиковой активности с нестабильными характеристиками рабочей нагрузки. Для каждого были рассчитаны текущие значения интегрального показателя производительности  $I(t)$ , представленные в виде графиков на рис 1. Для расчета  $I(t)$  использовалась Оболочка Системы Поддержки Принятия Решений (ОСППР) АСПИД-3W, предназначенная для всестороннего оценивания сложных объектов в условиях неопределенности [6]. На первом этапе весовые коэффициенты характеристик считались равными. Из графиков рис. 1 на временном интервале с пиковой нагрузкой (15:05-15:25) хорошо виден провал, соответствующий снижению производительности (пик нагрузки). В реальности, этот временной промежуток соответствует увеличению и изменению типа нагрузки (послеобеденная активность пользователей системы). На временном интервале с равномерной нагрузкой (14:15-14:35) во время обеденного перерыва, напротив, наблюдается достаточно стабильное поведение кривой с некоторыми пиками в сторону повышения интегрального показателя производительности.

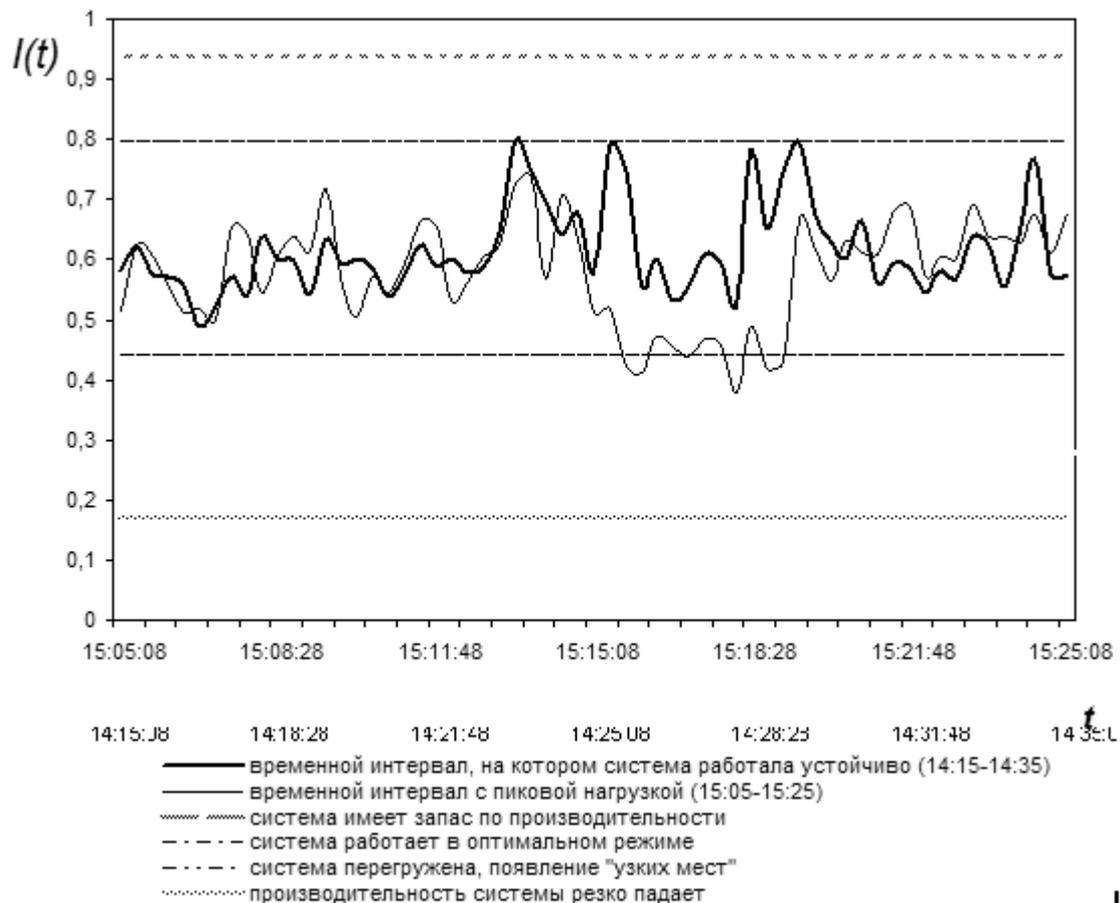


Рис. 1 Интегральная оценка производительности работы системы для различных временных интервалов.

На следующем этапе для каждого из временных интервалов были проведены расчеты  $I(t)$  с различными порядковыми отношениями между весовыми коэффициентами характеристик. При этом учитывались классы характеристик, то есть их принадлежность одной из трех базовых подсистем. При оценке системы ввода-вывода "веса" характеристик этого класса считались более значимыми, чем "веса" характеристик других подсистем. Такой же подход применялся и в отношении процессорной подсистемы и подсистемы памяти. Векторы весовых коэффициентов, соответствующие подсистемам ввода-вывода, процессорной и памяти соответственно, имеют вид:

$$W_{I/O} = (W_{\%busy}, W_{avserv+await}, W_{\%ufsipf});$$

$$W_{CPU} = (W_{runq-sz}, W_{\%wio});$$

$$W_{MEMORY} = (W_{pgscan/s}, W_{swp-sz}, W_{\%rcache}, W_{\%wcache});$$

Результаты расчетов  $I(t)$  производительности подсистем ввода-вывода ( $W_{I/O} > W_{CPU}; W_{I/O} > W_{MEMORY}$ ), процессора ( $W_{CPU} > W_{I/O}; W_{CPU} > W_{MEMORY}$ ) и памяти ( $W_{MEMORY} > W_{CPU}; W_{MEMORY} > W_{I/O}$ ) для каждого из временных интервалов представлены в виде графиков на рис. 2 и 3.

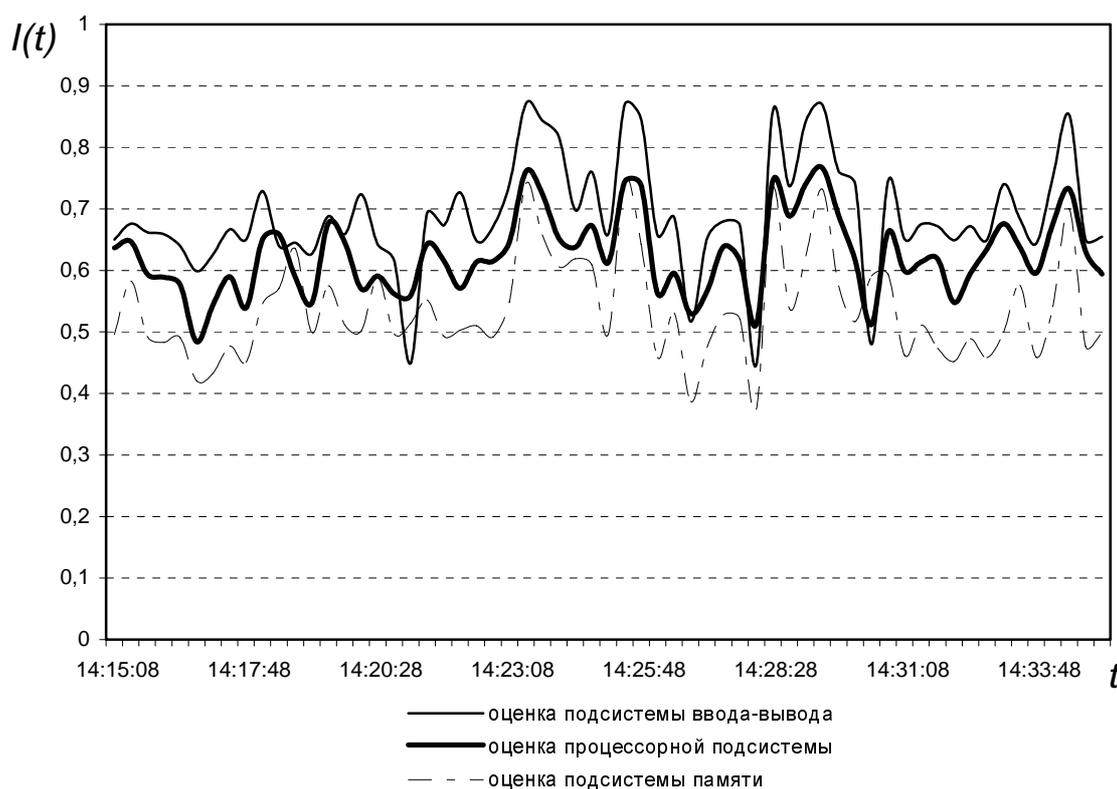


Рис. 2. Интегральная оценка производительности подсистем на временном интервале, в течение которого нагрузка была равномерной.

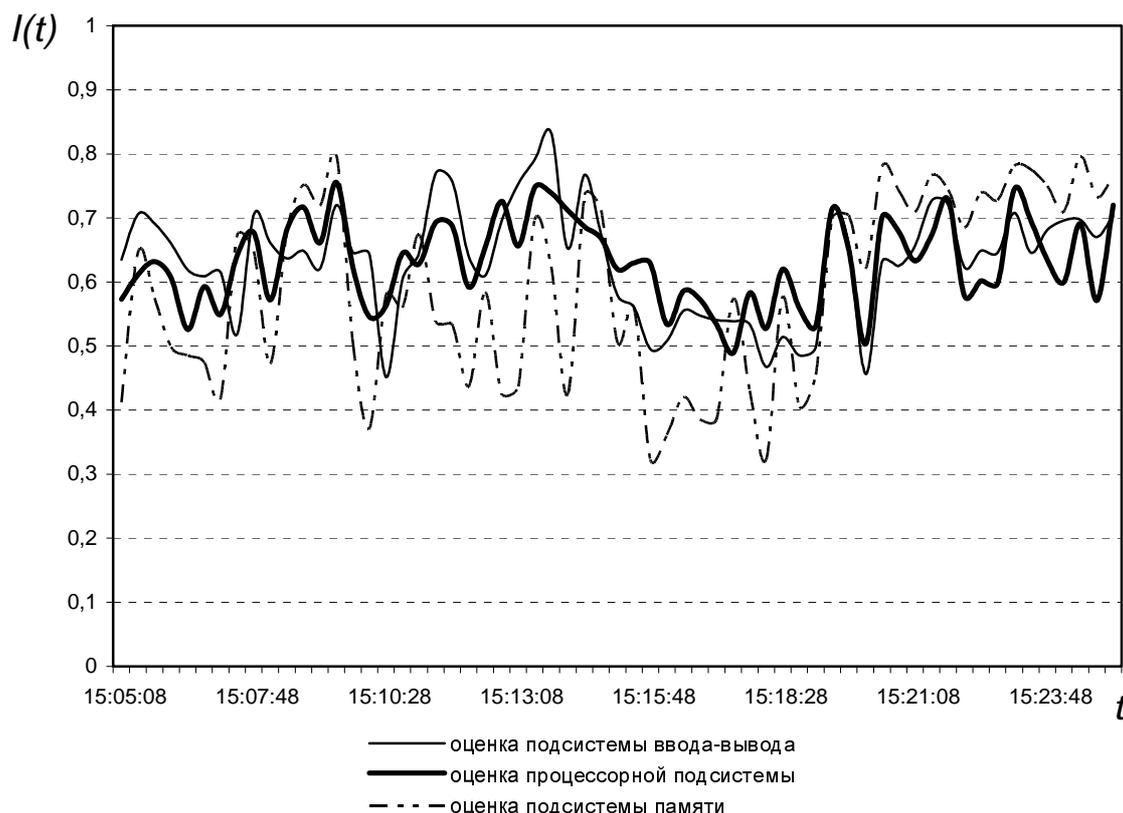


Рис. 3. Интегральная оценка производительности подсистем на временном интервале с пиковой и неравномерной нагрузкой.

Степень загрузки системы в каждый момент времени может быть определена путем соотношения между текущим значением показателя  $I(t)$  и значениями  $I_i$  типовых режимов работы системы. Расчет значений  $I_i$  для четырех режимов работы (слабая, нормальная, сильная загрузка и перегрузка системы) был проведен методом свертки с использованием экспертных оценок значений характеристик и дал следующие результаты:  $I_1 = 0,937$ ,  $I_2 = 0,799$ ,  $I_3 = 0,442$ ,  $I_4 = 0,171$ . Таким образом, можно утверждать, что система слабо загружена и имеет запас по производительности при  $I(t) \in [I_2; I_1]$ , имеет оптимальную загрузку при  $I(t) \in [I_3; I_2]$  и перегружена при  $I(t) \in [I_4; I_3]$ . При  $I(t) > I_1$  и  $I(t) < I_4$  можно говорить о неоправданной недогрузке (или отсутствии прикладных задач) и критической перегруженности. Так как  $I(t)$  определяется рядом не зависящих друг от друга и часто неучтенных факторов, изменяющихся во времени, то можно говорить, что  $I(t)$  носит псевдослучайный характер. Это позволяет оперировать такими характеристиками, как математическое ожидание  $I_0(T)$  и дисперсия  $d(T)$  на конечном временном интервале  $T$ . Для заданного интервала значение  $I_0(T)$  характеризует **эффективность**, а  $d(T)$  — **стабильность** (устойчивость) работы системы. Отсюда можно сделать важный вывод: для  $I(t)$  увеличение  $I_0(T)$  в интервале  $[I_4; I_2]$  и уменьшение  $d(T)$  свидетельствует об увеличении эффективности и стабильности.

Дополнительную информацию о поведении системы дают графики оценки производительности подсистем (см. рис. 2, 3). На рис 2. кривые  $I(t)$  всех подсистем подобны (изменяются одинаково), что говорит о равномерной их загрузке. На рис 3, напротив, кривые расходятся, в некоторых точках наблюдается увеличение производительности одной подсистемы в ущерб другой, что указывает на появление "узких мест, характерное при  $I(t) \in [I_4; I_3]$  или  $I(t) < I_4$ . Таким образом графики  $I(t)$ , полученные для подсистем, позволяют сделать выводы о **доминировании загрузки** той или иной подсистемы, то есть выявить причину наблюдаемой нестабильности режима работы системы в целом.

#### 4. Заключение

Анализ экспериментальных данных подтверждает эффективность предлагаемого метода оценки производительности системы по интегральному показателю  $I(t)$ . Его применение позволяет не только оценить, но и моделировать процессы, происходящие в сложных системах. Такие характеристики поведения системы, как эффективность, стабильность и доминирование нагрузки дают возможность объективно оценивать процессы, происходящие в системе, планировать нагрузку и принимать обоснованные решения, направленные на повышение производительности. Применение метода возможно и при рассмотрении более сложных систем, таких как кластеры высокой готовности и распределенные сети GRID. В этом случае к трем рассматриваемым подсистемам добавляется еще одна – сетевая, описываемая параметрами загруженности каналов и процентом ошибок при передаче данных, и.т.д.

Использование аппарата математической статистики позволяет уточнить значения  $I_0$  и  $I_i$  путем нахождения среднеквадратических отклонений  $\sigma_0$  и  $\sigma_i$ , определить достоверность ранжирования и тем самым получить более точные данные о поведении системы. Однако, для получения этих оценок и применимости метода для более сложных систем требуется существенно больший объем экспериментальных данных и детальная спецификация оцениваемой системы, что является предметом дальнейших исследований.

#### Литература

- [1] Л.Н. Королев. Структуры ЭВМ и их математическое обеспечение. М., Наука 1978. — 35-37 с.
- [2] Ж-П. Д. Мусумеси, М. Лукидес. Настройка производительности UNIX-систем. СПб: Символ-плюс, 2003. — 14 с., 218-228 с.
- [3] Д.А. Менаске, В.А.Ф. Алмейда. Производительность Web-служб: анализ, оценка и планирование. СПб: ДиаСофтЮП, 2003. — 224 с., 227 с.
- [4] Амир Маджидимер. Оптимизация производительности UNIX. М: Альфа-букс, 2002. — 178-198 с.
- [5] В.И. Воробьев. Инструментальные и измерительные средства переноса программ в сети ЭВМ. Диссертация на соискание ученой степени доктора технических наук. СПб. 1994 — 105 с.
- [6] Хованов К.Н. Комплекс алгоритмов генерации композиций для построения систем поддержки принятия решений. СПб. 2004 — 8 с.
- [7] Siemens Nixdorf Informationsysteme AG. Reliant UNIX 5.44: Tuning Guide. SNI AG, 1998. — 33-82 с.
- [8] A. Cockroft. Sun Performance and Tuning, Java and the Internet, 2<sup>nd</sup> ed. SUN Microsystems, 1998. — 277-282 с.