

## АНАЛИЗ ЗАКРИТИЧЕСКОЙ УСТОЙЧИВОСТИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

И.Э. Вишняков

vishnyakov@bmstu.ru

И.П. Иванов

ivanov@bmstu.ru

О.А. Одинцов

odintsov@bmstu.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

---

### Аннотация

Изложен подход к численному анализу устойчивости распределенных информационных систем под нагрузкой, изменяющейся во времени, допускающих кратковременное превышение нагрузкой критических значений. Информационные системы рассматриваются как неидеальные системы массового обслуживания, в которых интенсивность потока сообщений может зависеть от размеров очередей заявок. Исходя из такого подхода, сформулированы критерии устойчивого и неустойчивого поведения системы под нагрузкой, где под устойчивым поведением системы понимается способность системы самостоятельно возвращаться к штатному режиму работы при снижении значения внешней нагрузки менее критического, а под неустойчивым — переход в аварийное состояние. В качестве примеров рассмотрены наиболее часто встречающиеся типы неустойчивых систем, а также пример распределенной системы, реализующей сложную систему очередей, которая в зависимости от характеристик обслуживания может оказаться как устойчивой, так и неустойчивой. Для такой системы рассмотрен пример ее анализа, подтверждаемый результатами численного моделирования. По результатам анализа определены зоны устойчивости и неустойчивости. Рассмотрены особенности реализации распределенных информационных систем реального времени, приведены практические рекомендации по реализации устойчивых информационных систем

### Ключевые слова

*Распределенные информационные системы, системы реального времени, анализ устойчивости, системы массового обслуживания, проектирование информационных систем, системный анализ*

Поступила 27.02.2020

Принята 10.08.2020

© Автор(ы), 2021

---

**Введение.** Распределенная информационная система — это, как правило, информационная система, состоящая из нескольких независимых узлов, взаимодействующих между собой через коммуникационную среду, напри-

мер локальную вычислительную сеть. Примером классической распределенной информационной системы является распределенная система управления базами данных (СУБД). Основным отличием системы реального времени от обычной информационной системы является то, что в них присутствует явное ограничение на возможное время обработки данных, выраженное через параметры, связанные с временными характеристиками каких-либо реальных физических процессов. В процессе своего развития системы реального времени [1] постепенно усложнялись вплоть до появления в них явно выделенного слоя системного программного обеспечения, например операционных систем QNX или VxWorks, совершенствование которых, в частности реализация стандартов POSIX, практически стерло грань между ними и обычными операционными системами с точки зрения системного администрирования и разработки прикладных программ. При этом совершенствование вычислительной техники и методов разработки программного обеспечения, а также усложнение требований к информационным системам привело к тому, что ограничения, характерные для систем реального времени, стали возникать и в обычных распределенных системах, например, использующих web-интерфейс. Это принципиально отличает такие информационные системы от классических, например, от системы электронной почты, построенной на агентах пересылки [2], в которых отсутствует привязка времени обработки данных системой к субъективному восприятию ожидания отклика пользователем.

Само по себе увеличение нагрузки на инфраструктуру информационной системы, связанное с наличием высокой нагрузки на ее входе, не является опасным для типичных информационных систем. Однако в большинстве случаев к информационной системе в явном или неявном виде предъявляются требования по восстановлению штатного режима работы при снижении входной нагрузки. Тем не менее возможна ситуация, когда характер функционирования информационной системы в результате действия интенсивной нагрузки претерпевает качественные изменения, препятствующие ее дальнейшему функционированию. Эта ситуация наиболее часто проявляется в случае с распределенными информационными системами, поскольку для них в связи с наличием коммуникационных каналов характерно использование сложных систем очередей при передаче заявок или обрабатываемых данных между узлами.

Указанное требование восстановления после воздействия интенсивной внешней нагрузки хорошо укладывается в классическое понятие устойчивости как способности системы сохранять свое состояние при внешних воздействиях. Применительно к информационной системе термин «состо-

яние» описывает штатный режим работы. В противовес к штатному режиму в неустойчивых информационных системах существует второй постоянный режим «аварии», в котором система не способна выполнять свои функции и выйти из которого без существенного внешнего воздействия (например, перезапуска серверных блоков) она не может. В терминах теории катастроф [3] переход из режима штатной работоспособности в режим аварии при наличии на входе постоянной (не изменяющейся скачкообразно) внешней нагрузки называется бифуркацией.

В большинстве известных работ [4, 5] устойчивость информационных систем рассматривается с точки зрения теории надежности аппаратуры, применяемой как к отдельным узлам распределенной информационной системы, так и к коммуникационной среде, обслуживающей взаимодействие между узлами. Применительно к системам массового обслуживания под устойчивостью иногда понимают непрерывность зависимости характеристик анализируемых систем от управляющих параметров [6]. Также широкое распространение имеет подход анализа устойчивости работы информационных систем с точки зрения наличия ошибок в реализации описывающих их поведение алгоритмов [7–9]. Значительный прогресс в повышении надежности аппаратных средств, связанный с внедрением в промышленность нанометровых технологических процессов, а также появление простых и удобных средств разработки и отладки программного обеспечения привели к тому, что острота указанных проблем в последнее время стала снижаться. Поэтому на первый план начали выходить проблемы, связанные с устойчивостью работы информационных систем в условиях неравномерности интенсивности входных данных. Применительно к корпоративным сетям с коммутаторами эта проблема рассмотрена в [10], где показано, что наличие неоднородностей в вероятностных характеристиках входного трафика приводит к переполнению очередей коммутаторов и потере данных.

Значительно более серьезной является проблема построения неустойчивых информационных систем, связанная с неправильным выбором дисциплин взаимодействия очередей на этапе проектирования, причем термин «очередь» в данном случае может описывать объекты достаточно разной природы, например очередь планировщика задач, дисковое хранилище с обрабатываемой информацией, таблицы заданий в базе данных и т. д. В настоящей работе показано, что использующие даже достаточно надежные аппаратные средства и написанные без программных ошибок информационные системы могут иметь второе «аварийное» состояние и соответствующие ему точки бифуркации, если в процессе проектирования не будут

учтены требования к устойчивости их функционирования в закритической области. Такого рода дефекты проектирования информационных систем делают их крайне уязвимыми к атакам типа «отказ в обслуживании» [11], а повышение доступности средств разработки и снижение порога входа в эту область приводит к тому, что с течением времени актуальность данной проблемы только нарастает.

**Простая информационная система с экспоненциальным ростом длины очереди.** Рассмотрим для примера простую информационную систему, состоящую из входного пуассоновского потока заявок, очереди заявок, исполнителя и выходного потока (рис. 1). Введем математические характеристики устойчивости информационной системы.

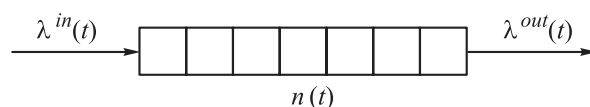


Рис. 1. Простая система с одной очередью

Пусть на входе системы интенсивность потока заявок равна  $\lambda^{in}$  и может изменяться с течением времени. При этом в системе, кроме выходного потока заявок интенсивности  $\lambda^{out}$ , имеется также внутренний поток заявок с интенсивностью, зависящей от текущего размера очереди заявок  $n(t)$ , который учтем в потоке  $\lambda^{out}$ , взяв с противоположным знаком. В дальнейшем будем рассматривать систему в предельном состоянии, в котором случайные величины реализуют какие-то конкретные значения, поэтому система из стохастической становится детерминированной, а возможные стохастические возмущения ее предельного состояния будем моделировать с помощью вектора приращений, принимающего некоторое неизвестное заранее значение.

Условие совместности числа заявок может быть записано в виде дифференциального уравнения

$$\dot{n} = \lambda^{in}(t) - \lambda^{out}(t), \quad (1)$$

где  $\dot{n}$  — скорость изменения числа заявок в очереди по времени.

В силу того что процесс не является стационарным, в общем случае правая часть уравнения (1) не будет равной нулю и в системе будет происходить накопление или распределение заявок. Предположим, что поддержание работоспособности очереди заявок требует отправки дополнительных сообщений, интенсивность которых пропорциональна длине очереди  $n$  с некоторым коэффициентом пропорциональности  $\alpha$ . Тогда выражение для интенсивности выходного потока будет иметь вид

$$\lambda^{out}(t) = \lambda^{out}(n(t)) = \lambda^{out0} - \alpha n(t), \quad (2)$$

где  $\lambda^{out0}$  — интенсивность заявок выходного потока при пустой очереди;  $\alpha$  — коэффициент нагрузки, описывающий снижение эффективности обработки при увеличении длины очереди, например, как вероятность отправки заявки при обходе очереди в процессе обслуживания. Подставляя выражение (2) в (1), получаем дифференциальное уравнение

$$\dot{n} - \alpha n = \lambda^{in}(t) - \lambda^{out}(n), \quad (3)$$

в котором для краткости зависимость  $n$  от  $t$  опущена и из которого для случая стационарного входного потока  $\lambda^{in}$  несложно получить решение для функции длины очереди:

$$n(t) = \frac{\psi}{\alpha} + \left( n_0 - \frac{\psi}{\alpha} \right) e^{\alpha t},$$

где  $\psi = \lambda^{out0} - \lambda^{in}$ ;  $n_0 = n|_{t=0}$  — размер очереди в начальный момент времени (момент стабилизации входного потока).

При этом для скорости изменения размера очереди  $\dot{n}$  будет справедливо выражение

$$\dot{n} = (\alpha n_0 - \psi) e^{\alpha t}.$$

Из полученного выражения следует, что для стационарного входного потока при  $n_0 > \psi / \alpha$  скорость изменения числа заявок в очереди будет положительной и увеличивающейся, а при  $n_0 < \psi / \alpha$  — отрицательной и продолжающей уменьшаться.

Более того, подобная тенденция сохранится даже при отсутствии заявок на входе, при  $n_0 > \lambda^{out0} / \alpha$  заявки в очереди будут генерироваться самой системой обслуживания очереди. Итоговый вид зависимости числа заявок в очереди от времени будет носить экспоненциальный характер с положительным показателем степени. Такие зависимости характерны для цепных химических и ядерных реакций в нестационарном режиме [12] и в физических процессах проявляются в виде взрыва. Применительно к информационным системам подобная динамика будет проявляться в резком, не согласованном с входной нагрузкой увеличении потребления системных ресурсов (память, процессорное время) без какой-либо возможности нормализации состояния.

Поэтому для рассматриваемой системы с одной очередью под устойчивостью мы будем понимать выполнение условия  $\dot{n} < 0$ , откуда для рассмотренной системы следует условие  $n < n_{crit}$ , где  $n_{crit} = \lambda^{out0} / \alpha$  — кри-

тическая длина очереди, после превышения которой начинается «самовозбуждение» системы даже при полном снятии внешней нагрузки. То же самое значение  $n_{crit}$  можно получить непосредственно из уравнения (3), приняв условие стационарности  $\dot{n} = 0$ .

При всей искусственности рассмотренного примера такую ситуацию легко получить на практике. Наиболее очевидным образом она проявляется, если в системе имеется протоколирование, реализуемое в виде той же системы очередей, что и обработка входящих заявок, но, возможно, с меньшим объемом данных. В этом случае, при попытке отправки в журнал информации о заявках, находящихся в очереди, будет выполняться увеличение ее длины, а параметр  $\alpha$  будет соответствовать относительному размеру потока сообщений журнала либо вероятности добавления новой заявки при анализе очереди.

**Неустойчивая система с линейным ростом длины очереди.** Более реалистичную ситуацию можно смоделировать, приняв, что каждая заявка, находящаяся в системе, увеличивает время обработки очереди на некоторую небольшую постоянную величину, связанную с необходимостью обслуживания очереди. При этом будем считать, что существует небольшой внутренний поток входящих заявок  $\lambda^{sys}$  фиксированной ненулевой интенсивности, связанный с поддержанием работоспособности системы. В этом случае интенсивность выходного потока будет описываться выражением

$$\lambda^{out}(n) = \frac{\lambda^{out0}}{1 + \beta n} - \lambda^{sys},$$

где параметр  $\beta$  характеризует линейную зависимость времени обработки отдельной заявки от текущей длины очереди.

Система выйдет на стационарный режим при  $\lambda^{out} = \lambda^{in}$ , откуда при  $\lambda^{in} = 0$

$$n_{crit} = \frac{\lambda^{out0} - \lambda^{sys}}{\beta \lambda^{sys}}.$$

Решение трансцендентного дифференциального уравнения (1) для этого случая может быть выражено через  $W$ -функцию Ламберта, но для практического использования значение имеет лишь динамика увеличения длины очереди, т. е. вид функции  $\dot{n}$  при больших значениях  $n$ . В этом случае первое слагаемое в выражении для  $\lambda^{out}$  будет несоизмеримо мало по сравнению со вторым и, следовательно,  $\dot{n} = \lambda^{in} + \lambda^{sys}$ , т. е. система будет вести себя так, как будто обработка заявок в ней полностью остановле-



на. Таким образом, в отличие от предыдущего случая численное решение  $n(t)$  для  $n_0 > n_{crit}$  при больших  $n$  имеет вид, близкий к линейному, что оставляет некоторое время персоналу, обслуживающему информационную систему, на принятие решений по стабилизации ее работы. Однако, как и в предыдущем случае, при выходе из зоны устойчивости без внешнего вмешательства в работу системы она не сможет продолжить свое функционирование.

Подобное поведение характерно для систем, в которых с заявками, находящимися в очереди, должны регулярно выполняться некоторые действия, например проверяться тайм-аут их ожидания или готовность к обработке, причем сложность этих действий является линейной относительно длины очереди, что является характерной особенностью простых очередей. В этом случае для системы при отсутствии внешней нагрузки остается возможность вернуться в рабочее состояние лишь путем снижения интенсивности служебных заявок  $\lambda^{sys}$  ниже уровня  $\lambda^{out0} / (1 + \beta n)$ . Проблема заключается в том, что при больших текущих значениях  $n$  возврат системы в устойчивое состояние  $n < n_{crit}$  может занять продолжительное время, поскольку даже при нулевом потоке входящих заявок скорость обработки очереди для больших  $n$  будет близка к нулю. Такое поведение может оказаться недопустимым для систем реального времени и также привести к переходу системы в состояние аварии.

**Система с экспоненциальным замедлением обработки.** Рассмотрим пример информационной системы, достаточно часто встречающийся на практике. Пусть имеется система, реализующая прием запросов по схеме клиент–сервер. При поступлении запроса сервер создает новый поток, принимает запрос, разбирает его параметры и запускает в этом потоке выполнение программного кода, реализующего запрос. Результаты выполнения программного кода отправляются клиенту. Будем считать, что время выполнения одного запроса при отсутствии остальных равно  $T$ , а многозадачность в системе реализована по вытесняющей схеме с достаточно малым квантом времени.

Для анализа этой системы перейдем от реального времени  $t$  к процессорному времени  $\tau$ , тогда  $d\tau = dt / n$ , где  $n$  — текущее число активных потоков в системе (число элементов системной очереди планировщика задач).

Запишем условие равновесия потока как

$$\frac{dn}{dt} = \lambda^{in} - \lambda^{out},$$

где интенсивность входящего потока в реальном времени будем считать постоянной. Поскольку

$$\frac{dn}{dt} = \frac{dn}{d\tau} \frac{d\tau}{dt} = \frac{1}{n} \frac{dn}{d\tau},$$

получаем аналогичное уравнение, выраженное в процессорном времени,

$$\frac{dn}{d\tau} = n(\lambda^{in} - \lambda^{out}).$$

Следовательно, относительно процессорного времени интенсивность входного потока увеличивается пропорционально числу выполняющихся запросов. Все запросы должны быть выполнены через интервал процессорного времени  $T$ , поэтому интенсивность выполненных запросов на выходе, отнесенных к процессорному времени данного узла в момент времени  $\tau$ , будет равна интенсивности поступивших запросов в момент времени  $\tau - T$ , т. е.  $n(\tau)\lambda^{out} = n(\tau - T)\lambda^{in}$ , следовательно,

$$\frac{dn}{d\tau} = \lambda^{in} (n(\tau) - n(\tau - T)). \quad (4)$$

Решение этого уравнения будем искать в экспоненциальном виде как  $n = n_0 e^{\alpha\tau}$ , где  $n_0$  — размер очереди планировщика задач в начальный момент времени;  $\alpha$  — неизвестный параметр решения. Подставив пробное решение в уравнение (4), получаем трансцендентное характеристическое уравнение вида  $(\lambda^{in} - \alpha)e^{\alpha T} = \lambda^{in}$ , решение которого можно выразить через  $W$ -функцию Ламберта. Очевидно, что при увеличении параметра  $T$  решение для  $\alpha$  достаточно быстро должно стремиться к  $\lambda^{in}$ , следовательно,  $n(\tau) \approx n_0 e^{\lambda^{in}\tau}$  и относительно процессорного времени очередь растет экспоненциально.

При этом интенсивность выходного потока  $\lambda^{out}$ , отнесенная к реальному времени, будет равна

$$\lambda^{out} = \lambda^{in} \frac{n(\tau - T)}{n(\tau)} \approx \lambda^{in} e^{-\lambda^{in}T} = \text{const},$$

т. е. либо при увеличении интенсивности входного потока  $\lambda^{in}$ , либо при увеличении времени обработки  $T$  она будет экспоненциально стремиться к нулю, оставаясь при этом постоянной во времени при постоянной  $\lambda^{in}$ . Из этого можно сделать вывод, что в реальном времени при постоянной интенсивности на входе длина очереди планировщика задач будет расти



с постоянной скоростью, практически равной интенсивности входящих заявок, а скорость выдачи результатов обработки стабилизируется на крайне незначительном уровне.

Для определения реального времени выполнения запроса  $T_r$  учтем, что

$$T_r = \int_{t-T}^t dt(\tau) = \int_{t-T}^t n(\tau)d\tau.$$

Следовательно,

$$T_r = \frac{n(t)}{\alpha} (1 - e^{-\alpha T}) = \frac{n(t)}{\lambda^{in}},$$

т. е. для такой  $\lambda^{in}$  очереди выполняется закон Литтла, при этом полученное выражение свидетельствует об экспоненциальном росте реального времени обработки при увеличении интенсивности входного потока или процессорного времени обработки  $T$ .

При наличии в системе нескольких процессорных ядер этот эффект оказывается не таким очевидным. Действительно, пока в системе имеются свободные ядра, время обработки заявок не будет увеличиваться. Однако сразу же по исчерпанию этого ресурса ситуация станет аналогичной описанной ранее с параметром времени обработки  $T^* = T/n_p$ , где  $n_p$  — число процессорных ядер.

Результаты имитационного моделирования размера очереди планировщика задач  $n$  от времени при изменяющейся интенсивности входного потока запросов  $\lambda^{in}$  для  $T = 2$  приведены на рис. 2. Из графика следует, что даже при снижении интенсивности входного потока менее критической величины  $1/T$  разгрузка системы начинает выполняться крайне медленно. В корпоративных сетях такая ситуация является достаточно частой, что связано с регулярным возникновением неоднородностей в вероятностных характеристиках входного трафика [13]. Для сравнения на том же графике приведена зависимость длины очереди от времени для случая последовательного выполнения задач ( $n_q$ ). Видно, что хотя общее время разгрузки очереди остается таким же (поскольку общее число запросов и средняя производительность их обработки совпадают), разгрузка очереди при последовательном выполнении происходит с постоянной интенсивностью и начинается непосредственно после снятия нагрузки, что важно для систем реального времени. Также из графика следует, что при этом используется значительно меньше ресурсов. На практике этот

эффект усиливается ввиду наличия дополнительных расходов на обслуживание очереди задач планировщика. При этом подобная система является устойчивой, т. е. способной разгружаться самостоятельно при снижении интенсивности входного потока менее фиксированной критической величины вне зависимости от текущего состояния системы.

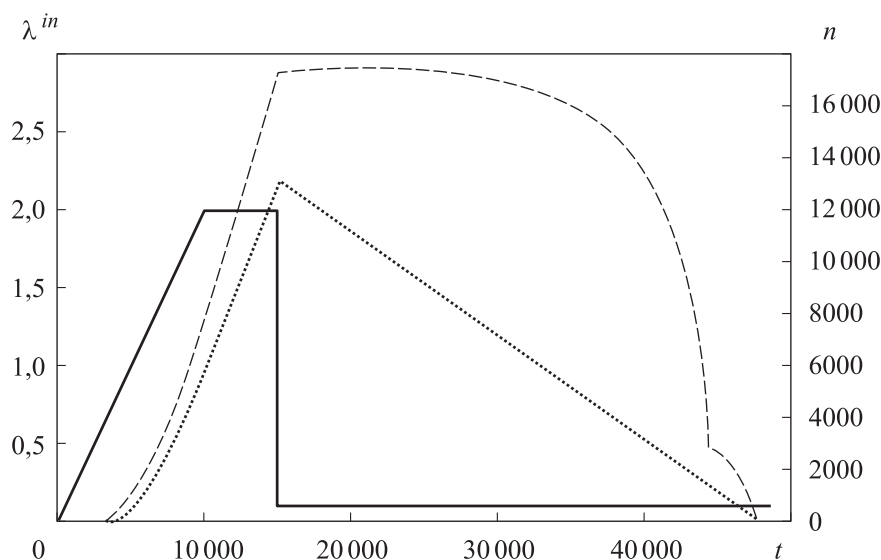


Рис. 2. Результат моделирования очереди планировщика задач:  
сплошная —  $\lambda^{in}$ ; штриховая —  $n$ ; штрихпунктирная —  $n_q$

**Сложная система очередей.** Как было отмечено, для распределенных систем обработки информации характерно наличие сложной системы очередей. В такой системе наиболее опасным является самовозбуждение, возникающее из-за наличия сложных дисциплин обработки заявок в очередях и приводящее к полному блокированию работы системы. Рассмотрим подход к анализу устойчивости такой системы. Пусть имеется модель информационной системы, состоящая из  $k$  очередей, соединенных между собой некоторым произвольным образом (рис. 3). Обозначим  $n_i$  — число заявок в  $i$ -й очереди;  $q_{ij}$  — интенсивность потока заявок, поступающих с выхода  $i$ -й очереди на вход  $j$ -й, либо порождаемых  $i$ -й очередью по отношению к  $j$ -й;  $\lambda_j^{in}$  — интенсивность потока внешних заявок, поступающих на вход  $j$ -й очереди, зависящего как от времени  $t$ , так и от размера других очередей;  $\lambda_j^{out}$  — интенсивность потока результатов, выходящих из  $j$ -й очереди во внешнюю среду. Будем считать, что  $q_{ij}$  и  $\lambda_i^{out}$  являются функциями  $n_i$ .

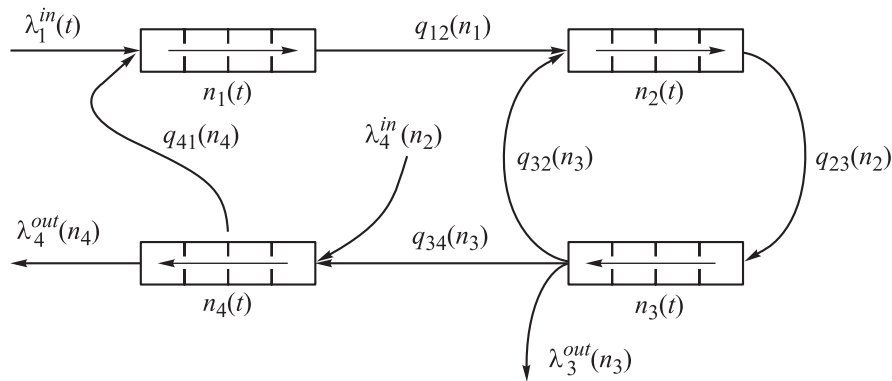


Рис. 3. Сложная система очередей

В отличие от физических систем в информационной системе уравнения равновесия в потоках информации могут не соблюдаться. Одно сообщение, поступающее на вход узла, на выходе может отправляться в виде копий нескольким получателям. Введем весовые коэффициенты  $\rho_{ij}$ , обозначающие, во сколько раз увеличивается единица потока сообщений при передаче с выхода  $i$ -го узла на вход  $j$ -го.

С учетом этого для  $i$ -й очереди можно записать уравнение равновесия

$$\dot{n}_i = \lambda_i^{in}(t, \{n\}) - \lambda_i^{out}(n_i) - \sum_{j=1}^k q_{ij}(n_i) + \sum_{j=1}^k \rho_{ji} q_{ji}(n_j).$$

Обозначим  $r_i = \dot{n}_i$  и перепишем уравнение равновесия в векторной форме:

$$\{r\} = \{\Lambda^{in}(\{n\})\} - \{\Lambda^{out}(\{n\})\} - \{s(\{n\})\}, \quad (5)$$

где  $\{\Lambda^{in}\{n\}\}$  — вектор-столбец интенсивностей внешней нагрузки;  $\{\Lambda^{out}\{n\}\}$  — вектор-столбец интенсивностей результатов;  $\{s\}$  — вектор-столбец внутренних интенсивностей.

Рассмотрим критическое состояние системы, когда размер всех очередей остается постоянным, т. е.  $\{r\} = 0$ . Пусть по какой-то причине вектор  $\{n\}$  получает произвольное малое приращение длины очередей  $\{\delta n\}$  и  $\{n^*\} = \{n\} + \{\delta n\}$ . В результате действия этого приращения на систему скорости изменения размера очередей  $\{r\}$  также получат приращение  $\{r^*\} = \{r\} + \{\delta r\} = \{\delta r\}$ .

Найдем значение этого приращения. Линеаризуем правую часть уравнения (5) относительно текущего значения  $\{n\} = \{n_{crit}\}$  и выразим приращение скоростей изменения размеров очередей для малого  $\{\delta n\}$  как

$$\{\delta r\} = \begin{bmatrix} \frac{\partial r_i}{\partial n_j} \end{bmatrix} \{\delta n\}; \quad \{\delta r\} + [A]\{\delta n\} = 0,$$

где

$$[A] = \begin{bmatrix} \frac{\partial s_i}{\partial n_j} \end{bmatrix} + \begin{bmatrix} \frac{\partial \Lambda_i^{out}}{\partial n_j} \end{bmatrix} - \begin{bmatrix} \frac{\partial \Lambda_i^{in}}{\partial n_j} \end{bmatrix}$$

— матрица частных производных скоростей изменения размера очередей по их длинам.

Для полученной матрицы  $[A]$  могут быть найдены собственные числа  $\alpha_i$  и соответствующие им собственные векторы  $\{v_i\}$ ,  $i = 1, \dots, k$ , такие, что  $[A]\{v_i\} = \alpha_i\{v_i\}$ , т. е. в некоторых очередях при увеличении числа элементов относительно критического состояния скорость изменения числа элементов в тех же очередях увеличивается пропорционально этому изменению с коэффициентом пропорциональности  $\alpha_i$ .

Это позволяет выполнить анализ устойчивости всей системы, т. е. поведения системы при ее отклонении от равновесного состояния. Действительно, если для некоторого  $i$   $\alpha_i < 0$  и все элементы  $v_i$  имеют один знак, это означает, что увеличение длины очередей на любое значение, пропорциональное  $v_i$ , может вызвать дальнейший рост размеров этих очередей даже при отсутствии внешней нагрузки и система перейдет во второе аварийное состояние. В случае  $\alpha_i > 0$  размер очередей будет уменьшаться и система потенциально может самостоятельно вернуться в рабочее состояние при снятии внешней нагрузки. Если все собственные числа матрицы  $[A]$  положительны и все элементы  $v_i$  имеют один знак, информационная система является устойчивой, т. е. способной самостоятельно возвращаться в рабочее состояние при снятии внешней нагрузки.

Наличие  $\alpha_i = 0$  показывает, что в системе существуют состояния безразличного равновесия, в которых добавление в очереди произвольного числа элементов, пропорционального  $\{v_i\}$ , не изменяет характера поведения системы, т. е. не приводит ни к уменьшению размера очередей, ни к их неограниченному росту. Это свойство характерно для классических систем массового обслуживания, в которых не учитывается использование ресурсов системы для поддержания очереди.

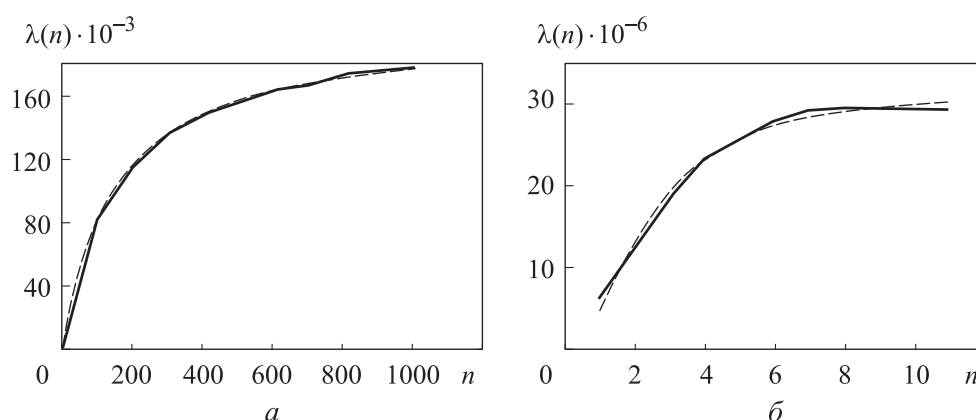
Комплексные значения  $\alpha_i$  свидетельствуют о наличии в системе периодического колебательного процесса, однако поскольку такое решение периодически будет проходить через нуль, эту ситуацию будем считать менее опасной, чем для случая отрицательных действительных собственных значений, так как очереди с отрицательной длиной не имеют практической

реализации. Аналогичная неопределенность возникает в случаях, если компоненты собственного вектора имеют разный знак, поскольку такое решение определяет разнонаправленную тенденцию в разных очередях в зависимости от различных начальных условий и в связи с условием неотрицательности размеров очередей не может быть выполнено. Для анализа собственных значений можно использовать методы определения корней характеристического полинома, рассмотренные в [14].

**Пример анализа информационной системы.** Рассмотрим пример анализа устойчивости информационной системы. Пусть имеется информационная система, обслуживающая входящие пакеты данных, передаваемые через коммуникационную среду. Считаем, что каждый пакет данных имеет фиксированный заголовок длиной  $h$  и переменную часть длиной  $l$ , что частота поступления входных пакетов имеет математическое ожидание  $f$ , следовательно, интенсивность входного потока равна  $f(h + l)$ . Входные пакеты освобождаются от заголовков и попадают в непрерывный буфер, в котором с ними выполняются некоторые действия, завершающиеся отправкой результата также в виде пакета данных но, возможно, уже другой длины. Будем считать, что работа с данными в буфере организована таким образом, что выходные пакеты могут быть сформированы сразу для всего буфера, размер которого в текущий момент времени равен  $n$ , а частота отправки выходных пакетов совпадает с частотой приема входных. Тогда интенсивность отправки выходных данных будет составлять  $f(h + n(t))$ .

Для такой системы  $\dot{n} = f(h + l) - f(h + n) = -f(n - l)$ . Применяя изложенный подход, получаем единственное собственное значение  $\alpha = f$ , следовательно, такая система является устойчивой. В действительности в информационных системах присутствует разнонаправленная динамика поведения. С одной стороны, группировка элементов увеличивает производительность обмена, ограниченную конструктивными особенностями коммуникационной среды, с другой — вызывает рост накладных расходов.

На рис. 4, а приведены результаты измерения предельной производительности дисковой подсистемы для потока записей размером 32 байта с группировкой в зависимости от размера группы. Индекс  $e$  соответствует экспериментальным данным, индекс  $a$  — аппроксимации вида  $\lambda_a(n) = an / (b + n)$ , где методом наименьших квадратов можно получить значения  $a = 205 \cdot 10^3$  записей в секунду,  $b = 153,5$ . Здесь параметр  $a$  соответствует предельной производительности;  $b$  — фиксированным накладным расходам.



**Рис. 4.** Производительность дисковой (а) и сетевой (б) подсистем ввода/вывода: сплошная —  $\lambda_c$ ; штриховая —  $\lambda_a$

Похожую динамику демонстрирует сетевой канал типа Gigabit Ethernet. На рис. 4, б приведены экспериментальные и аппроксимирующие кривые предельной производительности сетевой подсистемы, где для аппроксимирующих кривых использована зависимость вида  $\lambda_a(n) = an^2 / (b + n^2)$ , где  $a = 32,6 \cdot 10^6$  записей в секунду;  $b = 5,8$ .

В обоих случаях  $\partial\lambda / \partial n$  является положительной величиной, зависящей от  $n$ . Поскольку на практике, как правило, имеются ограничения на максимальное число одновременно передаваемых записей  $n_{\max}$ , а значение производной достаточно быстро уменьшается с увеличением  $n$ , в первичных расчетах на устойчивость производную следует принимать равной нулю, поскольку в этом случае она будет принимать наименьшее значение.

Рассмотрим пример неустойчивой информационной системы (первый вариант), в которой порождаются потоки данных (рис. 5). Система состоит из очередей обработки (индекс 1) и обслуживания (индекс 2). В процессе обработки формируется вспомогательный поток сообщений интенсивности  $\lambda_2$  в очередь обслуживания, из которой на обработку могут поступать дополнительные заявки интенсивности  $\lambda_1$ . Данные, выходящие из очереди обслуживания, группируются и отправляются в дисковую подсистему с пиковой производительностью  $\lambda_2^{out}$ . При увеличении длины очереди обработки производительность обработки снижается с коэффициентом пропорциональности  $\gamma_1$ , при этом пиковая производительность очереди обработки составляет  $\lambda_1^{out}$ .



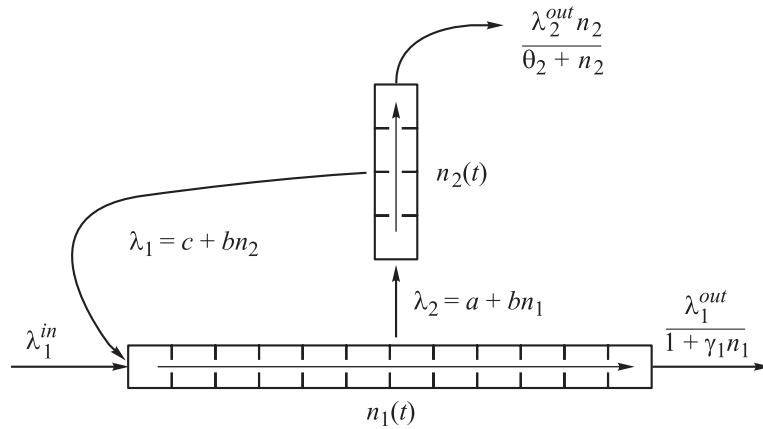


Рис. 5. Неустойчивая информационная система

Для такой системы запишем следующие уравнения равновесия:

$$\begin{aligned} r_1 = \dot{n}_1 &= \lambda_1^{in} - \frac{\lambda_1^{out}}{1 + \gamma_1 n_1} + c + d n_2; \\ r_2 = \dot{n}_2 &= a + b n_1 - \frac{\lambda_2^{out} n_2}{\theta_2 + n_2}. \end{aligned}$$

Дифференцируя их и принимая частные производные выходных потоков равными нулю, получаем следующее уравнение для возможных приращений длин очередей:

$$\begin{Bmatrix} \delta r_1 \\ \delta r_2 \end{Bmatrix} + \begin{bmatrix} 0 & -d \\ -b & 0 \end{bmatrix} \begin{Bmatrix} \delta n_1 \\ \delta n_2 \end{Bmatrix} = 0.$$

Решая характеристическое уравнение на собственные значения вида  $\alpha^2 - bd = 0$ , получаем значения корней  $\alpha_{1,2} = \pm \sqrt{bd}$ . При этом для отрицательного корня соответствующий ему собственный вектор будет иметь компоненты одного знака, что однозначно свидетельствует о неустойчивости такой системы.

Более сложной для анализа является система (второй вариант), основанная на предположении, что возвратный поток из очереди обслуживания в очередь обработки передается с использованием коммуникационной среды и имеет пиковую интенсивность  $\lambda_{12} n_{12}^2 / (1 + \gamma_{12} n_{12}^2)$ , учитывающую группировку элементов очереди. Такой поток будет достаточно быстро выходить на стационарный режим при увеличении длины очереди  $n_2$ , и соответствующий коэффициент матрицы частных произ-

водных будет приниматься равным нулю. В этом случае оба корня уравнения будут нулевыми, что свидетельствует о безразличном характере поведения системы. Для определения действительного характера поведения системы необходимо учитывать характер зависимостей на входе и выходе очередей. В этом случае матрица для определения собственных значений является матрицей переменных коэффициентов вида

$$A = \begin{bmatrix} -a_{11}(n_1) & -a_{12}(n_2) \\ -b & a_{22}(n_2) \end{bmatrix},$$

где все коэффициенты выбраны положительными, откуда следует решение для соответствующего характеристического уравнения вида

$$\alpha_{1,2} = \frac{1}{2} \left( a_{22} - a_{11} \pm \sqrt{(a_{11} + a_{22})^2 + 4ba_{12}} \right),$$

для которого одно из собственных значений получается отрицательным при компонентах соответствующего ему собственного вектора одного знака, следовательно, матрица частных производных не является положительно определенной, а информационная система не является устойчивой.

Рассмотрим третий вариант системы, в котором в очереди обработки используется группировка и интенсивность выходного потока составляет  $\lambda_1^{out} n_1 / (1 + \gamma_1 n)$ .

В этом случае матрица частных производных будет иметь вид

$$A = \begin{bmatrix} a_{11}(n_1) & -a_{12}(n_2) \\ -b & a_{22}(n_2) \end{bmatrix},$$

отличаясь от предыдущего рассмотренного варианта знаком элемента  $a_{11}$ , а решение характеристического уравнения примет вид

$$\alpha_{1,2} = \frac{1}{2} \left( a_{11} + a_{22} \pm \sqrt{(a_{11} - a_{22})^2 + 4ba_{12}} \right),$$

откуда можно получить условие устойчивости  $a_{11}a_{22} > ba_{12}$ , или с учетом выражений для частных производных

$$\frac{\lambda_1^{out}}{(1 + \gamma_1 n_1)^2} \frac{\lambda_2^{out} \theta_2}{(\theta_2 + n_2)^2} > 2b \frac{\lambda_{12} n_2}{(1 + \gamma_{12} n_2^2)^2}.$$

Полученное неравенство означает, что поведение системы в критическом состоянии зависит от числа элементов в каждой очереди, и в зависи-

мости от соотношения между ними система может находиться в устойчивом или неустойчивом состоянии.

Рассмотрим систему с параметрами  $\lambda_1^{out} = 10^4$ ;  $\gamma_1 = 2$ ;  $\lambda_2^{out} = 2 \cdot 10^5$ ;  $\theta_2 = 100$ ;  $b = 0,1$ ;  $\lambda_{12} = 2 \cdot 10^4$ ;  $\gamma_{12} = 3$ .

Для такой системы на рис. 6 показаны зоны устойчивости (stable) и неустойчивости, уравнение для границы которых в данном случае может быть найдено аналитически и представляет собой рациональную функцию. Поскольку процесс поступления заявок в общем случае является стохастическим, длины очередей в этой ситуации являются случайными величинами. В связи с этим даже при докритических интенсивностях входных сообщений существует ненулевая вероятность попадания в зону неустойчивости, и такую систему следует считать неустойчивой.

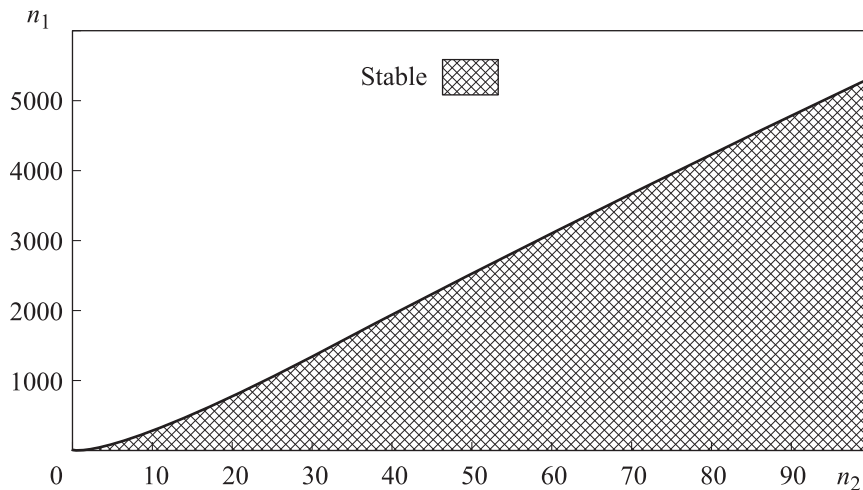


Рис. 6. Зоны устойчивости и неустойчивости для третьего варианта

Для сравнения рассмотрим четвертый вариант системы, в котором возвратный поток не имеет группировки и его интенсивность определяется выражением  $\lambda_{12} / (1 + \gamma_{12}n_2)$ . В этом случае матрица частных производных имеет вид

$$A = \begin{bmatrix} a_{11}(n_1) & a_{12}(n_2) \\ -b & a_{22}(n_2) \end{bmatrix}$$

и из характеристического уравнения могут быть найдены корни

$$\alpha_{1,2} = \frac{1}{2} \left( a_{11} + a_{22} \pm \sqrt{(a_{11} - a_{22})^2 - 4ba_{12}} \right),$$

которые либо при  $a_{11} - a_{22} > 2\sqrt{ba_{12}}$  являются положительными, либо являются комплексными с положительной действительной частью, что указывает на возможное возникновение в очередях системы затухающих колебаний.

Различие в поведении последних двух вариантов системы показано на рис. 7. В обоих случаях система подвергается действию одинаковой кратковременной интенсивной нагрузки, после чего внешняя нагрузка полностью снимается. График 1 описывает неустойчивое поведение системы для третьего варианта, при котором размер очереди обработки увеличивается с течением времени экспоненциально, график 2 — зависимость длины очереди обработки от времени при четвертом варианте реализации системы, при котором размер очереди стабилизируется и перестает расти неограниченно.

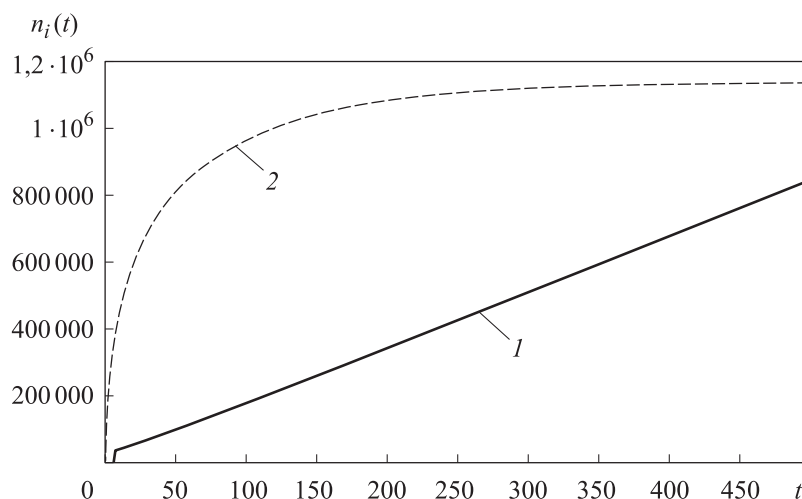


Рис. 7. Результаты моделирования для третьего (1) и четвертого (2) вариантов

Подобный подход можно распространить на системы с произвольным числом узлов обработки и очередей сообщений. Как следует из рассмотренного примера, в ряде случаев информационная система может являться условно устойчивой, когда количественное изменение параметров системы может привести к качественному изменению ее поведения под нагрузкой. Для корректного выполнения подобного расчета на практике необходимо выполнить определение достаточно большого числа параметров влияния текущего состояния узлов информационной системы на производительность их работы. В рассмотренных примерах параметры узла зависели только от состояния соответствующей очереди, однако в действительности для информационной системы, скорее всего, обнаружится более сложная

нелинейная зависимость, связанная, к примеру, со снижением производительности общей внутренней шины коммутаторов, организующих обмен между узлами по схеме звезда. Это приведет к усложнению выражений для матрицы  $[A]$  и, что более существенно, к возникновению неоднозначностей как в определении ее коэффициентов, так и в идентификации самих зависимостей. В связи с этим важно, чтобы инженеры, занимающиеся проектированием распределенных информационных систем, придерживались рекомендаций, приведенных далее.

**Заключение.** Подытоживая рассмотренные примеры, а также основываясь на опыте проектирования распределенных систем, можно предложить следующие рекомендации по проектированию устойчивых распределенных систем.

Не допускать ситуацию экспоненциального роста числа заявок в очередях. Если в системе имеется поток сообщений с интенсивностью, возрастающей пропорционально числу элементов в очереди этой системы, возникает потенциальная возможность возврата части этих сообщений в ту же очередь. По возможности следует ограничивать интенсивность этого потока некоторой предельной величиной.

При наличии очередей, выполненных в виде непрерывного буфера в памяти, нельзя допускать ситуацию перемещения в буфере блоков данных, размер которых зависит от числа данных в буфере. Эта задача должна решаться использованием циклического буфера с двумя указателями, начала и конца.

При разработке алгоритмов обработки данных необходимо стремиться обеспечить возможность повышения эффективности системы за счет групповой обработки данных. Например, голосовой кодек opus [15] для IP-телефонии поддерживает единовременное сжатие пакетов звука длительностью 2,5, 5, 10, 20, 40 или 60 мс. При этом увеличение интервала кодирования до 20 мс позволяет не только снизить накладные расходы за счет уменьшения объема заголовков, но и повысить степень сжатия данных при сохранении качества звука. Дальнейшее увеличение интервала кодирования незначительно повышает степень сжатия, однако эффект повышения общей эффективности за счет уменьшения объема заголовков продолжает действовать.

При проектировании систем реального времени могут существовать ограничения, связанные с такими характеристиками, как

– продолжительность нахождения заявок в очереди совместно с временем обработки отдельной заявки — время отклика (латентность) системы на поступление внешних данных;

– неравномерность выходного потока (джиттер) — разброс значений времени отклика системы.

Так, в IP-телефонии превышение на 100 мс времени обработки голосовых данных в сети оператора порога приводит к нарушению восприятия абонентом своего собеседника и вызывает дискомфорт при ведении диалогов. Это значительно ограничивает свободу инженера при разработке архитектуры распределенной системы реального времени. В указанной ситуации с кодеком opus значения интервала кодирования в 40 и 60 мс на практике оказываются неприменимы для кодирования звука в реальном времени, поскольку, складываясь с другими задержками в системе, не позволяют достичь требуемого качества обслуживания абонентов даже при сохранении высокого качества звука. Поэтому при проектировании устойчивых систем реального времени необходимо учитывать следующие рекомендации:

– недопустимо создавать отдельный поток (процесс) выполнения для каждого запроса. Необходимо заранее подготовить рабочие потоки, исходя из оценки имеющихся ресурсов, а распределение заявок между исполнителями осуществлять с помощью системы очередей;

– использовать очереди фиксированного максимального размера, значение которого рассчитывается заранее на основании известных стохастических свойств входных потоков и характеристик алгоритмов обработки;

– не допускать выгрузку содержимого очередей во внешнюю память с увеличенным временем доступа (файлы подкачки);

– не выполнять затратные действия с утрачиваемыми данными, например их сохранение для последующей обработки. Исключение составляет сбор статистики потерь, формирующий поток фиксированной интенсивности;

– при необходимости обеспечения полноты обслуживания предусматривать отдельные режимы функционирования системы при критической нагрузке, повышающие ее производительность путем целенаправленного снижения качества обслуживания. Примером этому может служить реализация режима overload (перегрузка) в стандарте ITU-T G.767 [16]. В этом стандарте предусмотрено использование в обычном режиме работы звукового кодека со скоростью выходного потока 16 Кбит/с, тогда как в режиме перегрузки предусмотрено автоматическое переключение кодека в режим 12,8 Кбит/с и далее 9,6 Кбит/с, обеспечивающие более низкое качество кодирования, но освобождающие ресурсы системы для прохождения критических участков.



## ЛИТЕРАТУРА

- [1] Кобылянский В.Г. Системы реального времени. Новосибирск, НГТУ, 2015.
- [2] Postel J.B. STD 10 RFC 821. Simple mail transfer protocol.  
DOI: <https://doi.org/10.17487/RFC0821>
- [3] Арнольд В.И. Теория катастроф. Новые идеи в естествознании. М., Изд-во МГУ, 1983.
- [4] Ушаков И.А., ред. Надежность технических систем. М., Радио и связь, 1985.
- [5] Дружинин Г.В. Надежность автоматизированных систем. М., Энергия, 1977.
- [6] Калашников В.В. Анализ устойчивости в задачах массового обслуживания методом пробных функций. *Теория вероятностей и ее применения*, 1977, т. 22, № 1, с. 89–105.
- [7] Василенко Н.В., Макаров В.А. Модели оценки надежности программного обеспечения. *Вестник НовГУ*, 2004, № 28, с. 126–132.
- [8] Канер С., Фолк Дж., Нгуен Е.К. Тестирование программного обеспечения. Киев, ДиаСофт, 2000.
- [9] Матвеев В.А., Басараб М.А., Троицкий И.И. Асимптотические свойства оценки вероятности ошибки тестирования систем информационной безопасности. *Инженерный журнал: наука и инновации*, 2013, № 11.  
DOI: <http://doi.org/10.18698/2308-6033-2013-11-997>
- [10] Ivanov I.P., Basarab M.A., Kolesnikov A.V. Evaluation efficiency forecast modeled self-similar process based on FARIMA-models. *AIP Conf. Proc.*, 2019, vol. 2195, art. 020052. DOI: <https://doi.org/10.1063/1.5140152>
- [11] Лобанов В.Е., Оныкий Б.Н., Станкевичус А.А. Архитектура системы защиты Грид от атак типа «Отказ в обслуживании» и «Распределенный отказ в обслуживании». *Безопасность информационных технологий*, 2010, т. 17, № 3, с. 136–139.
- [12] Семёнов Н.Н. Самовоспламенение и цепные реакции. *Успехи химии*, 1967, т. 36, № 1, с. 3–33.
- [13] Басараб М.А., Иванов И.П., Колесников А.В. Анализ моделей прогнозирования процессов сервера корпоративной сети. *Нелинейный мир*, 2015, т. 13, № 3, с. 18–31.
- [14] Калинина Е.А. Применение алгебраических методов для анализа сложных систем. Дис. ... д-ра физ.-мат. наук. СПб., СПбГУ, 2016.
- [15] Valin J.M., Vos K., Terriberry T. Definition of the opus audio codec. RFC 6716.  
DOI: <https://doi.org/10.17487/RFC6716>
- [16] Digital circuit multiplication equipment using 16 kbit/s LD-CELP, digital speech interpolation and facsimile demodulation/remodulation. *ITU-T Recommendation G.767*. Geneva, 1998.

**Вишняков Игорь Эдуардович** — старший преподаватель кафедры «Теоретическая информатика и компьютерные технологии» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, корп. 1).

**Иванов Игорь Потапович** — д-р техн. наук, профессор, заведующий кафедрой «Теоретическая информатика и компьютерные технологии» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, корп. 1).

**Одинцов Олег Александрович** — канд. техн. наук, доцент кафедры «Прикладная механика» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, корп. 1).

**Просьба ссылаться на эту статью следующим образом:**

Вишняков И.Э., Иванов И.П., Одинцов О.А. Анализ закритической устойчивости распределенных информационных систем реального времени. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2021, № 3 (136), с. 60–83.

DOI: <https://doi.org/10.18698/0236-3933-2021-3-60-83>

**ANALYSIS OF SUPERCRITICAL STABILITY  
OF DISTRIBUTED REAL-TIME INFORMATION SYSTEMS**

**I.E. Vishnyakov**

[vishnyakov@bmstu.ru](mailto:vishnyakov@bmstu.ru)

**I.P. Ivanov**

[ivanov@bmstu.ru](mailto:ivanov@bmstu.ru)

**O.A. Odintsov**

[odintsov@bmstu.ru](mailto:odintsov@bmstu.ru)

**Bauman Moscow State Technical University, Moscow, Russian Federation**

---

**Abstract**

The article describes an approach to the numerical analysis of the stability of distributed information systems under a load that varies over time allowing for a short-term load exceeding critical values. Information systems are considered as imperfect queuing systems in which the intensity of the message flow may depend on the size of the queues of applications. Based on this approach, criteria for the stable and unstable behavior of the system under load are formulated, where the stable behavior of the system is understood as the ability of the system to return to its normal mode of operation independently when the external load drops below the critical value, and unstable one when there is the transition to an emergency state. As examples, the most common types of unstable systems are considered, as well as an example of a distributed system that implements a complex queuing system, which, depending on the characteristics of the service, can be either stable or unstable. The analysis of such a system is carried out, which is confirmed by the results of numerical simulation. Based on the analysis results, zones of stability and instability are determined.

**Keywords**

*Distributed information systems, real-time systems, stability analysis, queuing systems, design of information systems, system analysis*

Features of the implementation of distributed real-time information systems are considered, practical recommendations for the implementation of sustainable information systems are given

Received 27.02.2020

Accepted 10.08.2020

© Author(s), 2021

## REFERENCES

- [1] Kobylanskiy V.G. *Sistemy real'nogo vremeni* [Real time systems]. Novosibirsk, NGTU Publ., 2015.
- [2] Postel J.B. STD 10 RFC 821. Simple mail transfer protocol. DOI: <https://doi.org/10.17487/RFC0821>
- [3] Arnol'd V.I. *Teoriya katastrof. Novye idei v estestvoznanii* [Catastrophe theory. New ideas in natural history]. Moscow, IMGU Publ., 1983.
- [4] Ushakov I.A., ed. *Nadezhnost' tekhnicheskikh system* [Reliability of technical systems]. Moscow, Radio i svyaz' Publ., 1985.
- [5] Druzhinin G.V. *Nadezhnost' avtomatizirovannykh system* [Reliability of automated systems]. Moscow, Energiya Publ., 1977.
- [6] Kalashnikov V.V. Stability analysis of in queuing problems by a method of trial functions. *Theory Probab. Appl.*, 1977, vol. 22, no. 1, pp. 86–103. DOI: <https://doi.org/10.1137/1122007>
- [7] Vasilenko N.V., Makarov V.A. Assessment models of software reliability. *Vestnik NovGU*, 2004, no. 28, pp. 126–132 (in Russ.).
- [8] Kaner C., Falk J., Nguyen H.Q. *Testing computer software*. Wiley, 1999.
- [9] Matveev V.A., Basarab M.A., Troitskiy I.I. Asymptotical properties of testing error probability estimates for information security systems. *Inzhenernyy zhurnal: nauka i innovatsii* [Engineering Journal: Science and Innovation], 2013, no. 11 (in Russ.). DOI: <http://doi.org/10.18698/2308-6033-2013-11-997>
- [10] Ivanov I.P., Basarab M.A., Kolesnikov A.V. Evaluation efficiency forecast modeled self-similar process based on FARIMA-models. *AIP Conf. Proc.*, 2019, vol. 2195, art. 020052. DOI: <https://doi.org/10.1063/1.5140152>
- [11] Lobanov V.E., Onykiy B.N., Stankevichus A.A. A DOS and DDOS attack protection system architecture for the Grid. *Bezopasnost' informatsionnykh tekhnologiy* [IT Security (Russia)], 2010, vol. 17, no. 3, pp. 136–139 (in Russ.).
- [12] Semenov N.N. Spontaneous ignition and chain reactions. *Russ. Chem. Rev.*, 1967, vol. 36, no. 1, pp. 1–17. DOI: <https://doi.org/10.1070/RC1967v036n01ABEH001579>
- [13] Basarab M.A., Ivanov I.P., Kolesnikov A.V. Analysis of the forecasting models of corporate network server processes. *Nelineynyy mir* [Nonlinear World], 2015, vol. 13, no. 3, pp. 18–31 (in Russ.).
- [14] Kalinina E.A. *Primenenie algebraicheskikh metodov dlya analiza slozhnykh sistem*. Dis. d-ra fiz.-mat. nauk [Using algebraic methods for complex systems analysis. Dr. Sc. (Phys.-Math. Diss.)]. St. Petersburg, SPbGU Publ., 2016 (in Russ.).

[15] Valin J.M., Vos K., Terriberry T. Definition of the opus audio codec. RFC 6716. DOI: <https://doi.org/10.17487/RFC6716>

[16] Digital circuit multiplication equipment using 16 kbit/s LD-CELP, digital speech interpolation and facsimile demodulation/remodulation. *ITU-T Recommendation G.767*. Geneva, 1998.

**Vishnyakov I.E.** — Senior Lecturer, Department of Theoretical Informatics and Computer Technologies, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5/1, Moscow, 105005 Russian Federation).

**Ivanov I.P.** — Dr. Sc. (Eng.), Professor, Head of Department of Theoretical Informatics and Computer Technologies, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5/1, Moscow, 105005 Russian Federation).

**Odintsov O.A.** — Cand. Sc. (Eng.), Assoc. Professor, Department of Applied Mechanics, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5/1, Moscow, 105005 Russian Federation).

**Please cite this article in English as:**

Vishnyakov I.E., Ivanov I.P., Odintsov O.A. Analysis of supercritical stability of distributed real-time information systems. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2021, no. 3 (136), pp. 60–83 (in Russ.). DOI: <https://doi.org/10.18698/0236-3933-2021-3-60-83>