“A systematic approach to building cloud special purpose systems”

P. E. Golosov, Candidate of Technical Sciences, Dean, Russian Presidential Academy of National Economy and Public Administration (RPANEPA), Moscow, e-mail: pgolosov@gmail.com

M. Gostev, Doctor of Technical Sciences, Leading Research Scientist, A.A. Kharkevich Institute for Information Transmission of the Russian Academy of Sciences (IIT RAS), Moscow, e-mail: igostev@gmail.com

“Системный подход к построению облачных систем специального назначения”

Голосов П.Е., к.т.н. Декан, Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации (РАНГХиГС), Москва

Гостев И. М., д.т.н., ведущий научный сотрудник, А.А. Институт проблем передачи информации имени А.А. Харкевича РАН, Москва (ИППИ РАН)

Признаки специализированной системы

- распределенные (облачные) ресурсы, к которым возможен полный и независимый доступ;
- задачи возможно распараллеливать, ветви задач не связаны по данным;
- в систему поступают задачи нескольких типов (различные потоки), интенсивность поступления каждого описывается одним из трех распределений;
- каждая задача имеет ограничения по времени выполнения;
- вычислительные ресурсы эквиваленты, но имеют разную производительность для каждого типа задач;
- к системе предъявляются требования со стороны ее владельца, ориентированные на различные режимы - полнота загрузки, отдача от предоставления ресурсов и иные.

«Вызовы» базовой модели

- управление выполнением отдельного задания осуществляется независимо от других заданий и состояния всей системы в целом;
- приоритеты, варианты расписания для отдельных заданий и их параметры не учитываются;
- отсутствие глобальных координирующих механизмов приводит как к увеличению времени выполнения отдельных заданий, так и к неполному использованию ресурсов;
- у пользователей отсутствуют сведения о сроках получения решения или сервисных возможностях, что ведет к потере качества обслуживания.

Предоставляемая пользователю возможность управления ожиданиями от быстродействия системы относительно каждого из заданий повышает уровень качества обслуживания отдельного задания, но может оказать отрицательное воздействие на выполнение других заданий.

Классические принципы управления заданиями

Два варианта классических планировщиков построены:

- на минимизации времени отклика (реального времени),
- на максимизации общей загрузки ресурсов (разделения времени).

Традиционная цель:

- улучшение состояния системы в целом, могут не поддерживаться механизмы требований отдельных потребителей.

Определение специальных распределенных (облачным) вычислительных системам

В работах [*] такой специализированный класс задач именуется как citu-задания, citu-задачи; от английского термина: computation - intensive - task - under - uncertainty. А системы называются специализированными вычислительными системами (СВС). Выполнение всех citu-задач в них должно происходить в режиме реального времени. Они допускают распараллеливание по данным и являются ресурсоёмкими. Их решения должны быть получены как можно быстрее и, в отдельности и в совокупности. Там же [*] вводится понятие директивный срок окончания (ДСО) – это установленный диспетчером момент календарного времени, до наступления которого citu- задача должна быть решена. Мы будем использовать термин – директивное время вычислений (ДВВ), которое определяется, как один из параметров сгенерированной задачи и означает, что задача должна быть выполнена за время, не превышающее значение этого параметра с момента её генерации.

* Malashenko Yu.E., Nazarova I.A. Controlling Computationally Intensive Heterogeneous Computational Tasks with Directive Deadlines // J. Computer and Systems Sciences International. 2012. V. 51. No 5. P. 628–635.

Виды специальных распределенных вычислительных систем

Существует необходимость решения ряда специальных задач:

- поиск информации по некоторому шаблону в распределенных базах данных за некоторое минимальное время (в том числе и в Интернете);
- поиск объекта на фото(видео) местности с летательного объекта в реальном времени;
- решение криптографических задач.
- поиск хеш-функций в задачах построения блокчейнов на основе стратегии доказательства работы;
- управление производственными процессами при изготовлении детаей и сборку из них узлов;
- поиск фрагментов ДНК;
- другие аналогичные задачи.

Некоторые требования к специальным распределенным (облачным) вычислительным системам

- все задачи должны выполняться в рамках заданного времени выполнения для каждой задачи (обеспечение ДВВ);
- необходима возможность выполнения одной задачи параллельно на нескольких вычислителях (например, подзадач, независимых по данным);
- функционирование системы при множестве входных потоков подзадач с различными законами их поступления (равномерный, экспоненциальный, пуассоновский и т. д.);
- загруженность ресурсов должна быть максимальной;
- в случае нахождения решения задачи в одной из подзадач, выполнение всех остальных подзадач (ветвей) должно быть прекращено;
- в случае отсутствия решения некоторой задачи при заданных условиях, она должна быть автоматически перезапущена с изменёнными начальными условиями;
- при отказе одного из вычислителей при решении некоторой подзадачи, она должна повторно быть выполнена.

В терминах теории массового обслуживания

Многоканальная сеть массового обслуживания с обратными связями с:

• множеством входных потоков задач с произвольными законами поступления;

• множеством обслуживающих приборов (серверов);

• дисциплина обслуживания постоянна для каждого типа задач, то есть время обслуживания каждой задачи определено при её генерации;

• все задачи после генерации сразу разделяются на некоторое произвольное число подзадач равной длины (длина определяется типом задачи).

• при разбиении задачи на подзадачи подразумевается, что решение существует в одной из

подзадач (случайно выбранной), в остальных время выполнения фиксировано и равно T/ n, где T максимальное время выполнения задачи, а n – число фрагментов (ветвей) деления задачи;

• время обслуживания успешных подзадач определяется по равновероятному закону от 0 до T/n;

В терминах теории массового обслуживания (2)

Данная сеть имеет две обратные связи:

- первая - повторное поступление всей задачи на выполнение в случае отсутствия решения при некоторых входных параметрах (то есть поступление её в систему как новую задачу того же типа);
- вторая обратная связь возникает при отказе сервера (отказы задаются на основе равновероятного закона). В этом случае (именно) данная подзадача поступает на выполнение в общую очередь подзадач. Эта обратная связь существует для всех серверов.

Количество подзадач в каждой новой сгенерированной задаче может варьироваться в зависимости от длины очереди.
Дисциплина обслуживания задач в очереди может быть выбрана любая, из следующих: FIFO, LIFO, по приоритетам (определенным по некоторым параметрам самих задач). Приоритеты могут изменяться в зависимости от изменения состояний системы.

В терминах теории массового обслуживания (3)

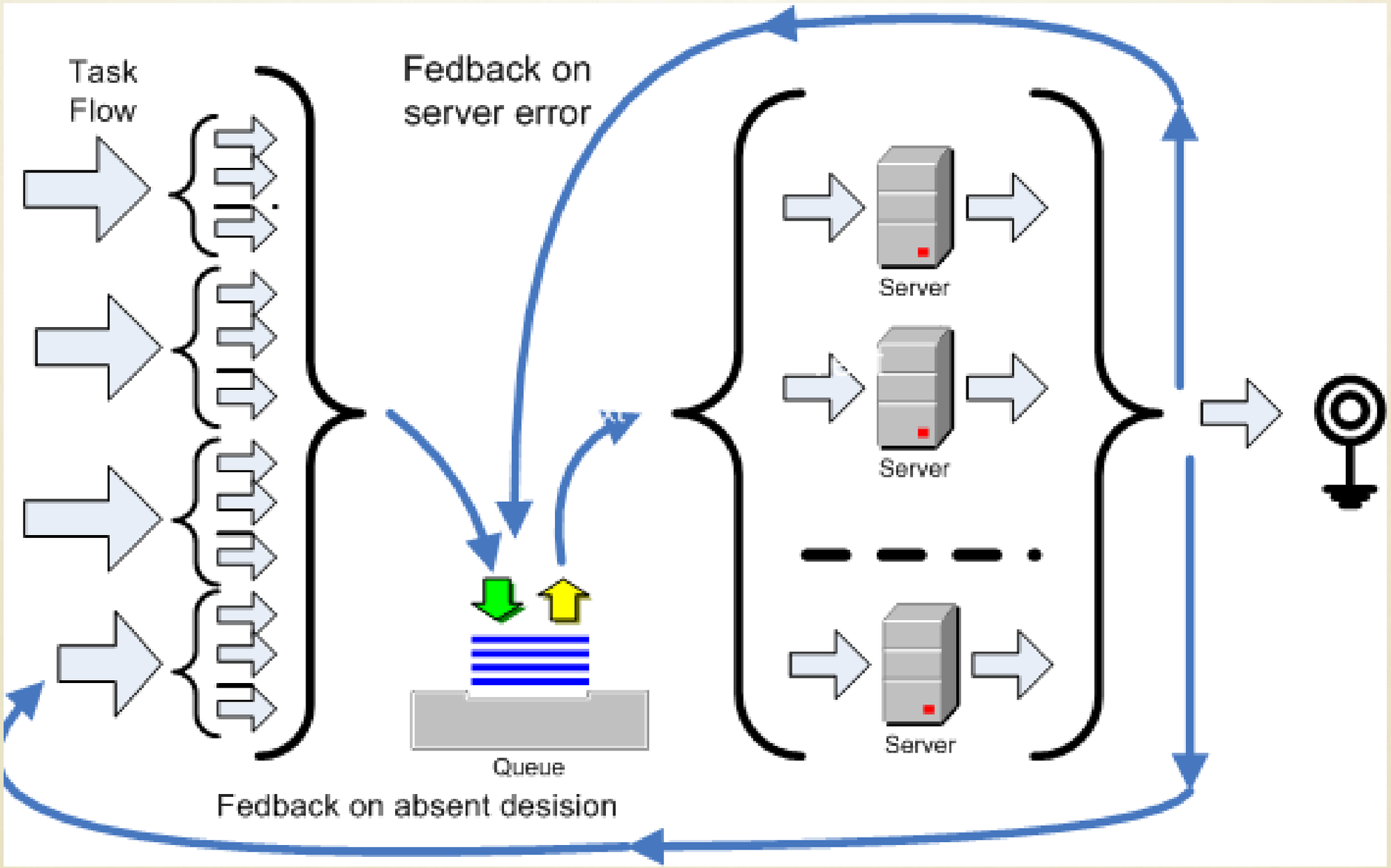
Основная цель построения такой сети массового обслуживания - решение особых классов задач в облачной СВС с максимальной эффективностью использования оборудования и гарантированным временем выполнения задач каждого типа.

В разработанной модели возможно динамическое изменение таких параметров системы как:

- количества подзадач для каждого типа и их приоритетов и для гарантирования их директивного времени выполнения;
- весовых коэффициентов для одного из типов задач для оптимизации их скорейшего выполнения возможно в ущерб остальным (блокчейн-модель);
- приоритетов задач при их генерации с целью выполнения некоторого количества задач в комплекте (совокупность некоторого количества задач разного типа).

Исходя из того факта, что поступление подзадач в обратных связях не подчиняется экспоненциальному закону, а имеет равновероятное распределение, данную систему можно отнести к классу G/G/n/∞. Из всех вышеприведенных условий следует, что построение аналитического описания такой системы не представляется возможным.

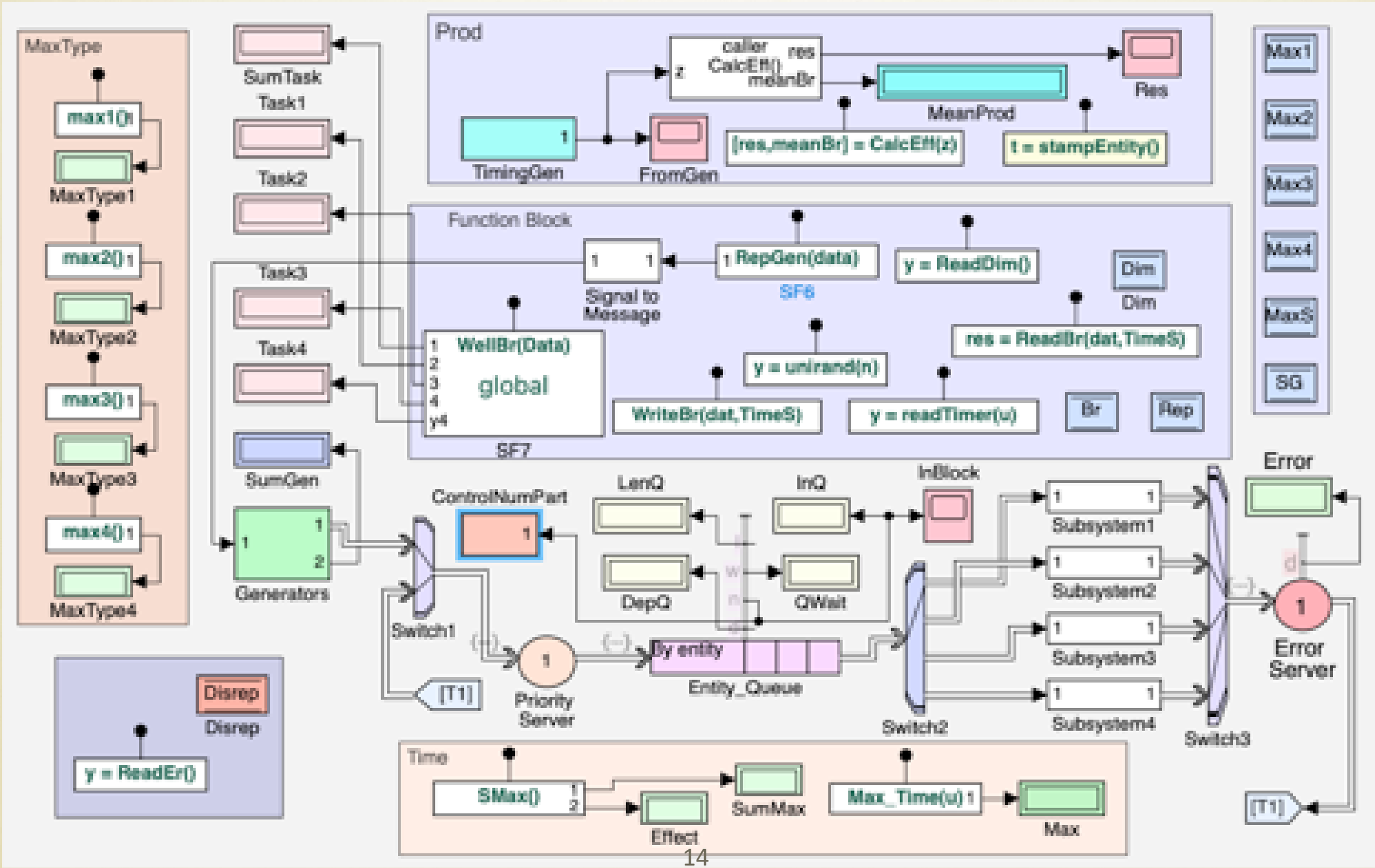
В терминах теории массового обслуживания (4)



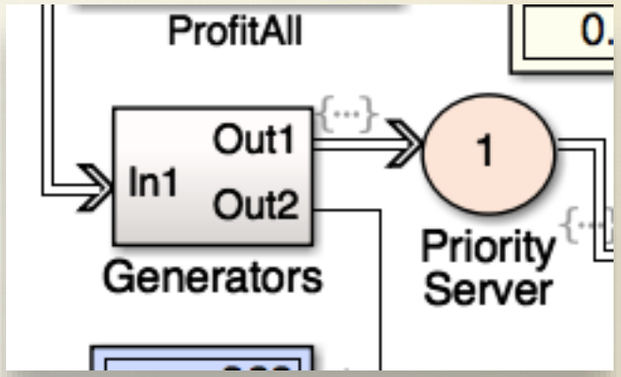
Принципы управления

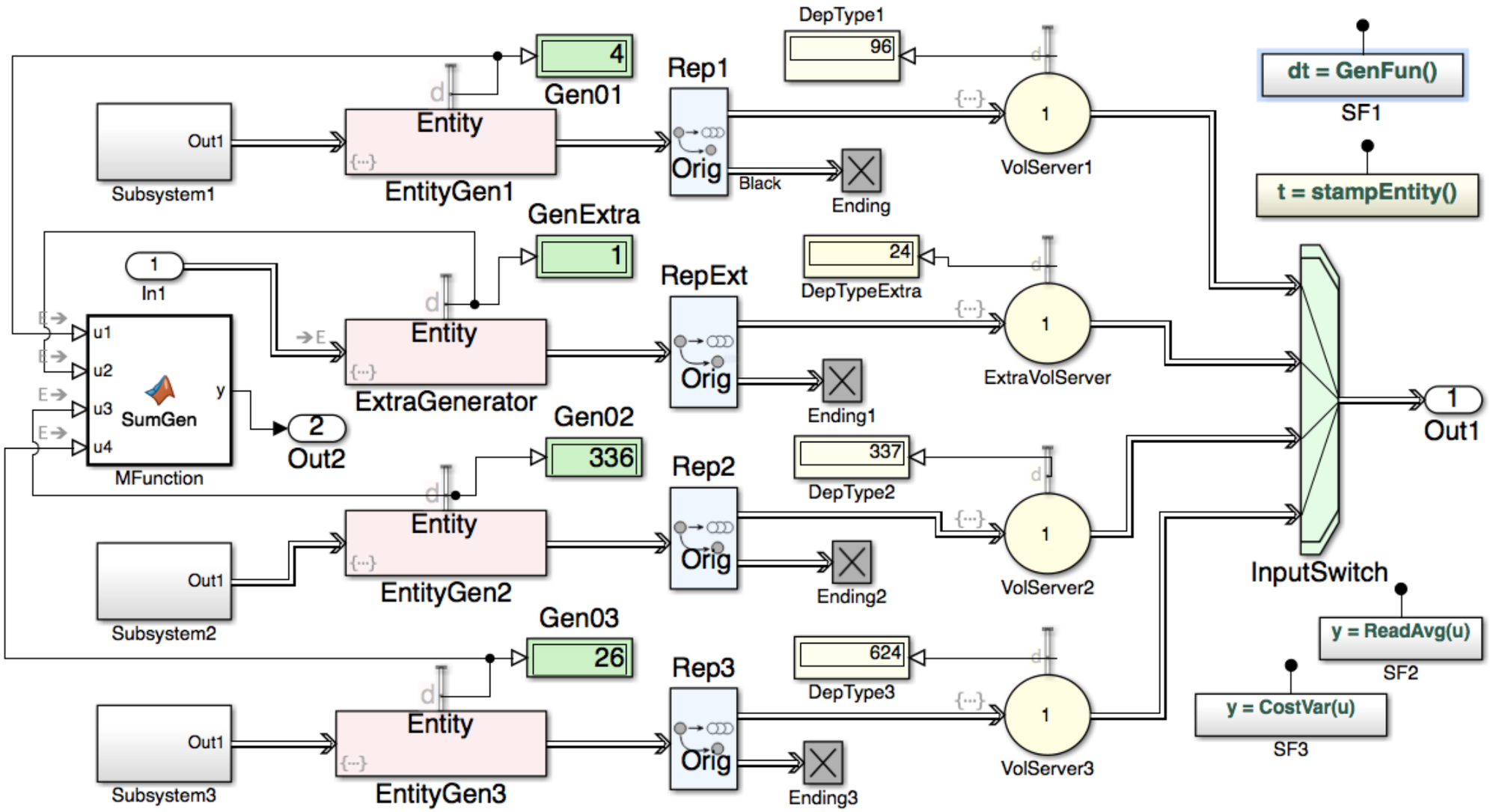
- Ряд предшествующих наших работ показал, что количество подзадач, на которое разбивается задача в процессе её выполнения, может служить одним из способов управления функционированием всей системы.
- Другим объектом управления может служить некоторый **параметр подзадачи** (например, приоритет задачи при выполнения, ДВВ и др.), изменяемый в зависимости от некоторого состояния или всей системы, или времени выполнения самой задачи.
- Для организации функционирования такой системы, **механизмы управления задачами были возложены на сами задачи**. Для этого каждой генерируемой задаче назначается ряд параметров, которые автоматически будут определять очерёдность её выполнения.
- Так как механизм управления задачами априори не известен, а прохождение задач в вычислительной системе не определено никакими функциями, то можно назвать такие механизмы **интеллектуальными агентами**, которые, и будут выполнять все вышеуказанные требования к вычислительной системе и определять производительность и эффективность её работы.

Архитектура системы

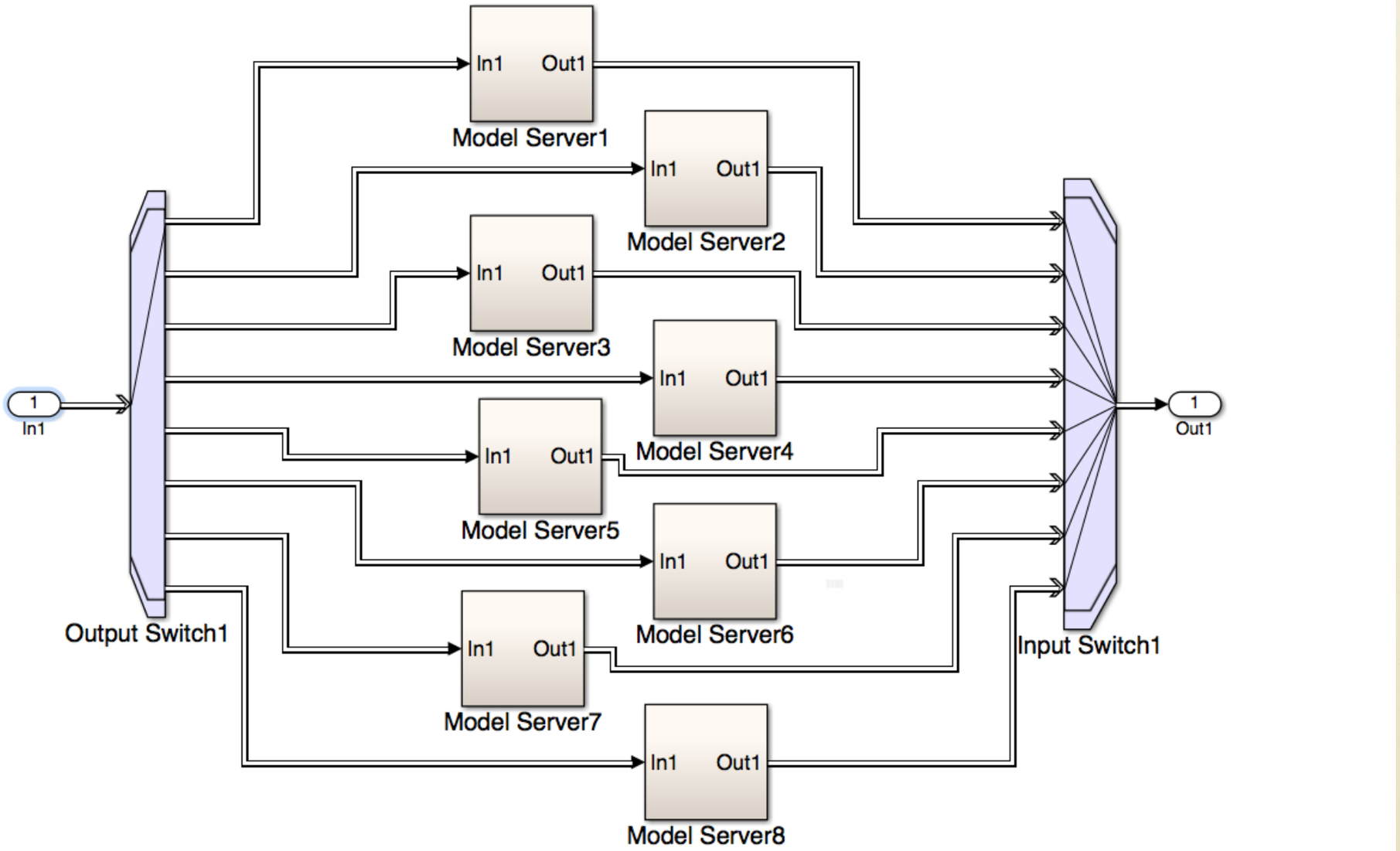


Блок генераторов





Серверная группа (SubSystem)



Модель сервера

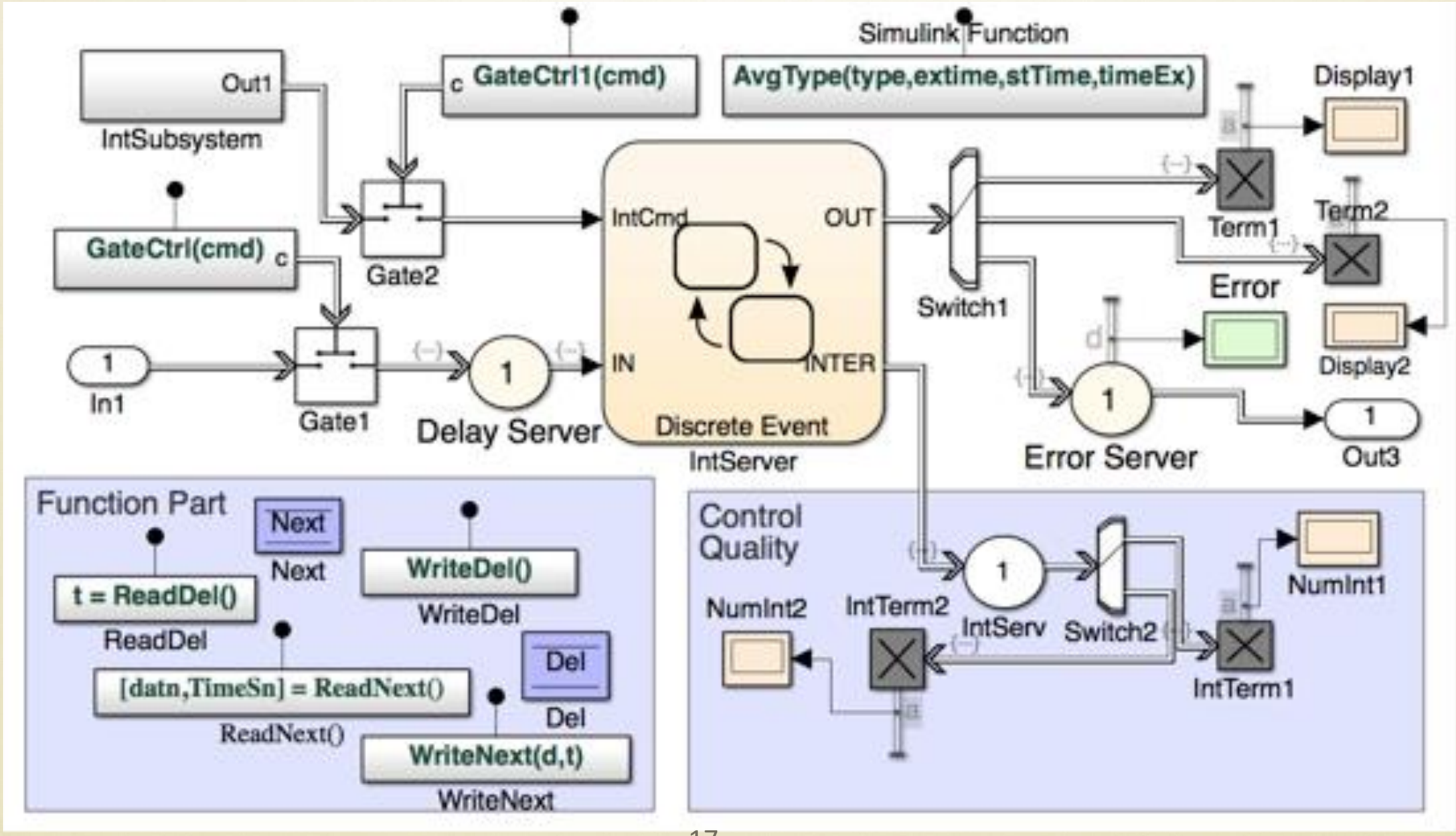


Chart Server

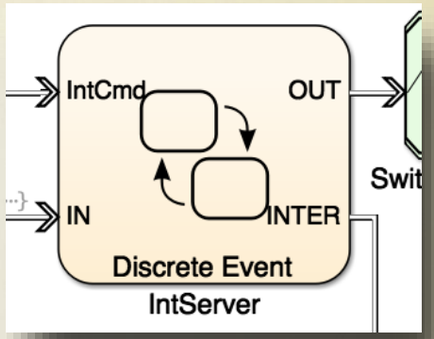
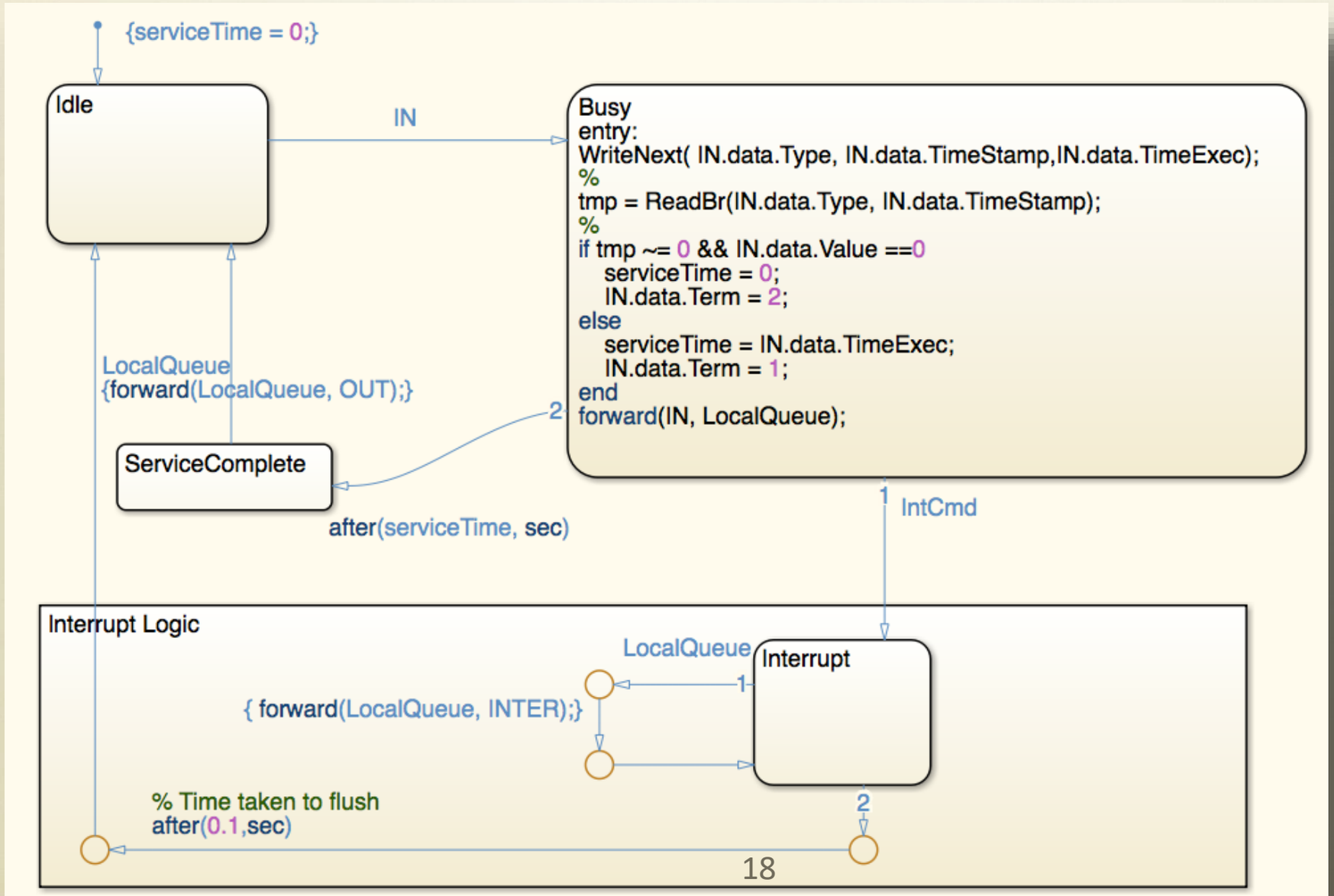
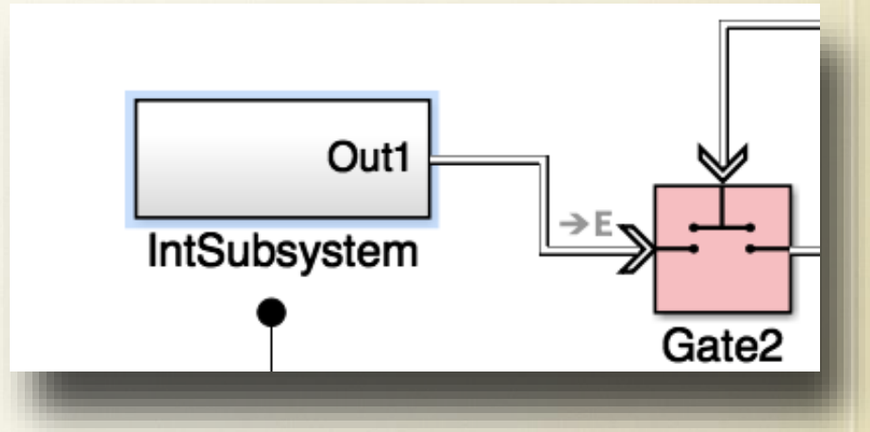


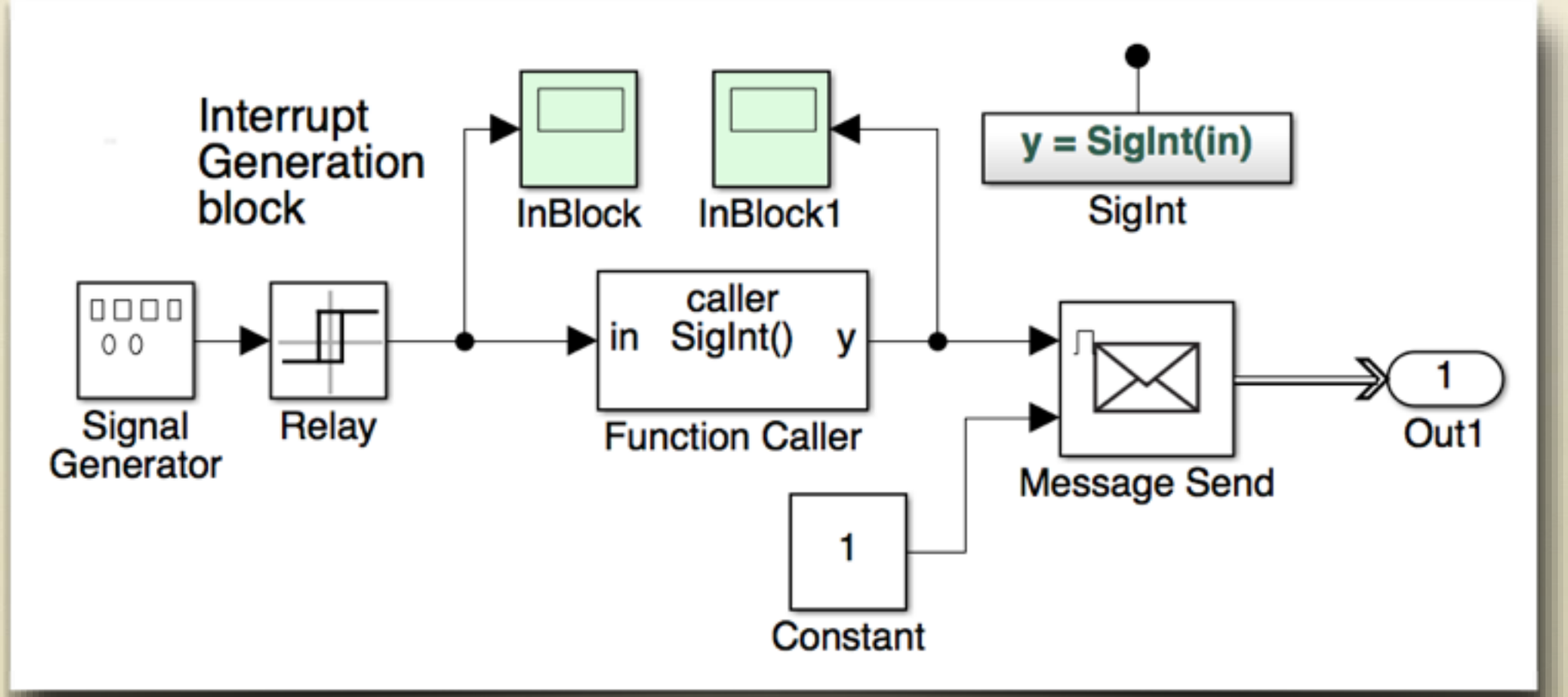
Chart Server специально разработан для реализации асинхронного механизма прерываний при успешном окончании выполнении подзадачи в другом сервере. Основан на автомате конечных состояний.



Подсистема прерываний

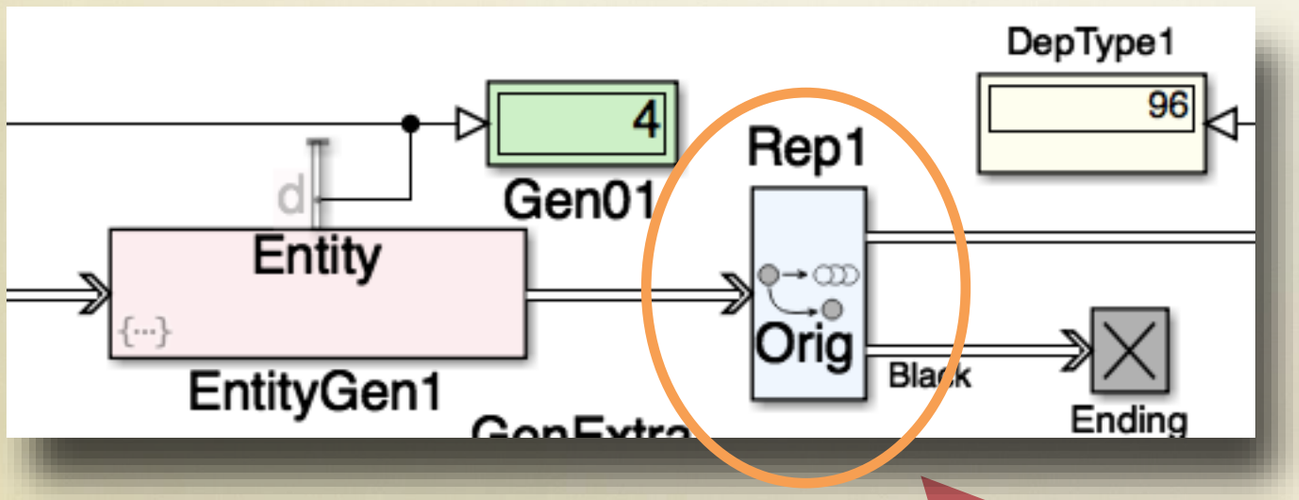
Подсистема разработана для реализации системы прерывания работы серверов Entity Server. Подсистема генерирует прерывание работы сервера в том случае, если в другой ветви задача была выполнена.

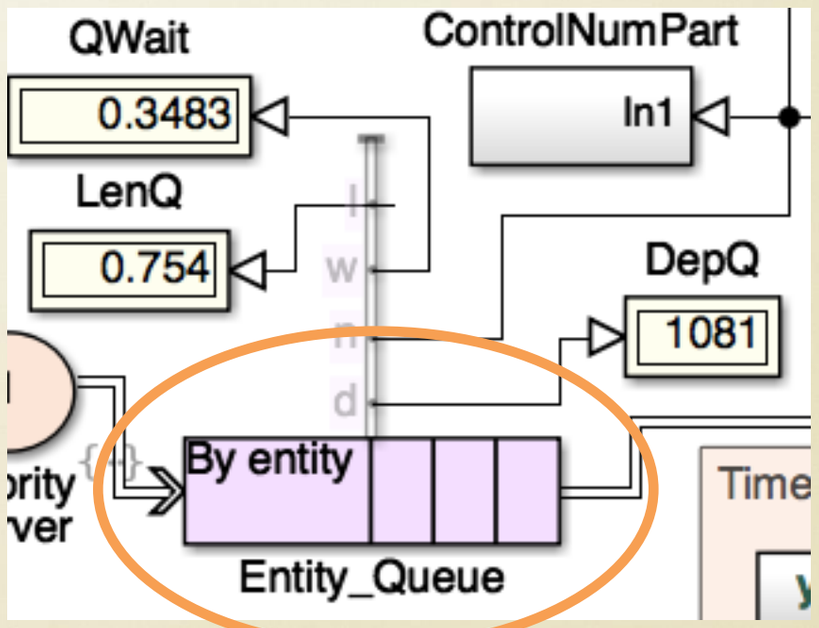




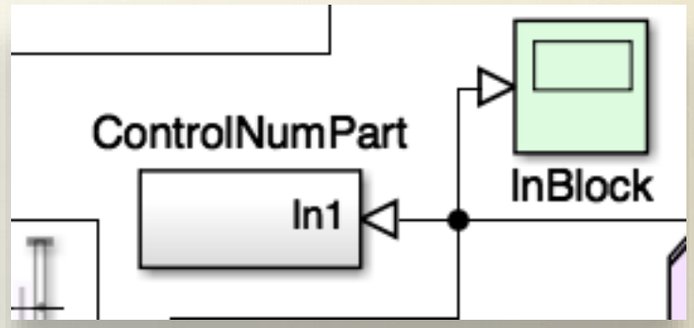
Sporadic Control #1 (ИА1)

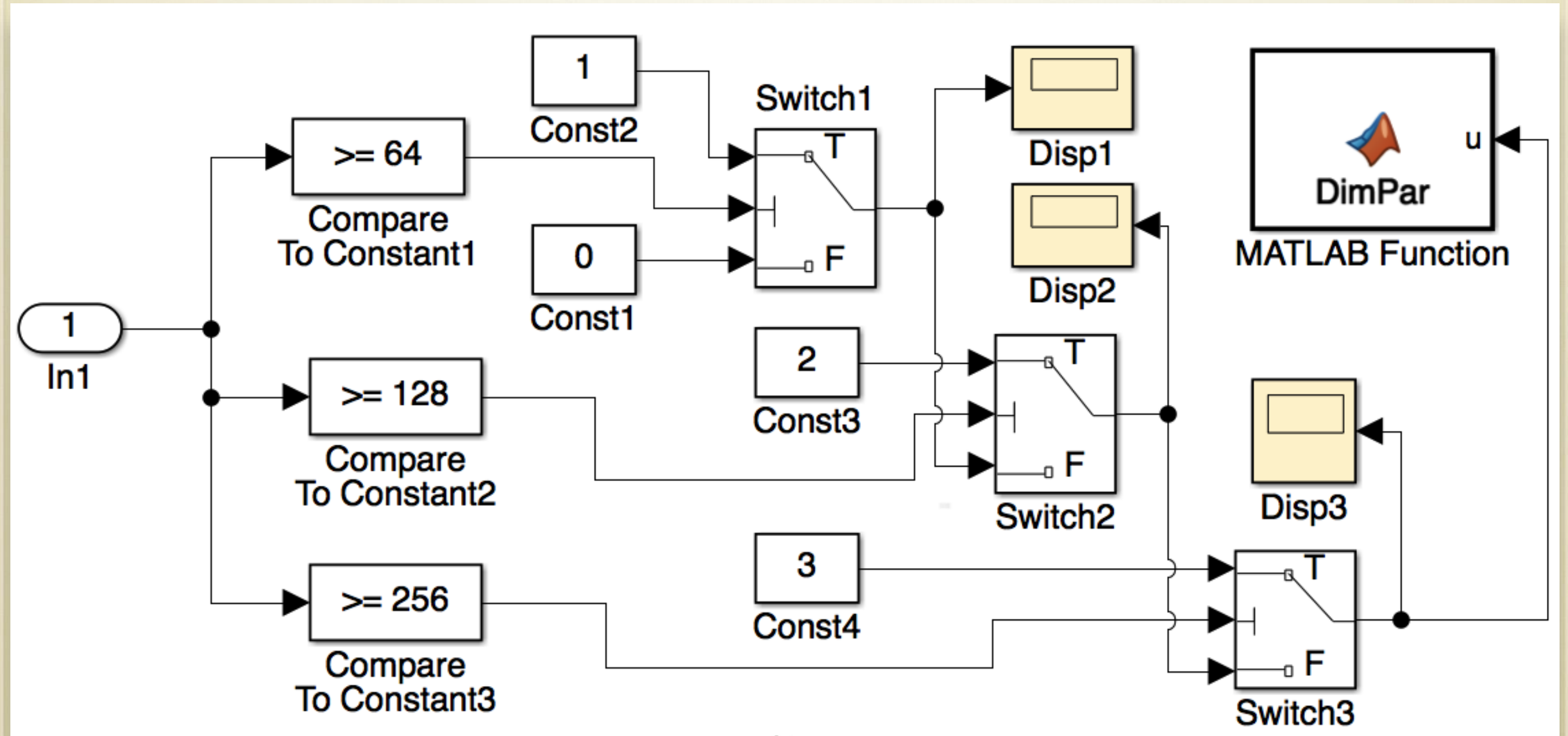
Для реализации спорадического управления использованы два интеллектуальных агента. Первый основан на контроле длины очереди задач (Entity_Queue). Его алгоритм расчитывает количество подзадач, на которые делится задача. Это зависит от длины очереди. Чем больше очередь, тем на меньшее число подзадач делятся поступающие задачи.
Количество подзадач определяется репликаторами Rep1-Rep3 на основании работы подсистемы ControlNumPart.





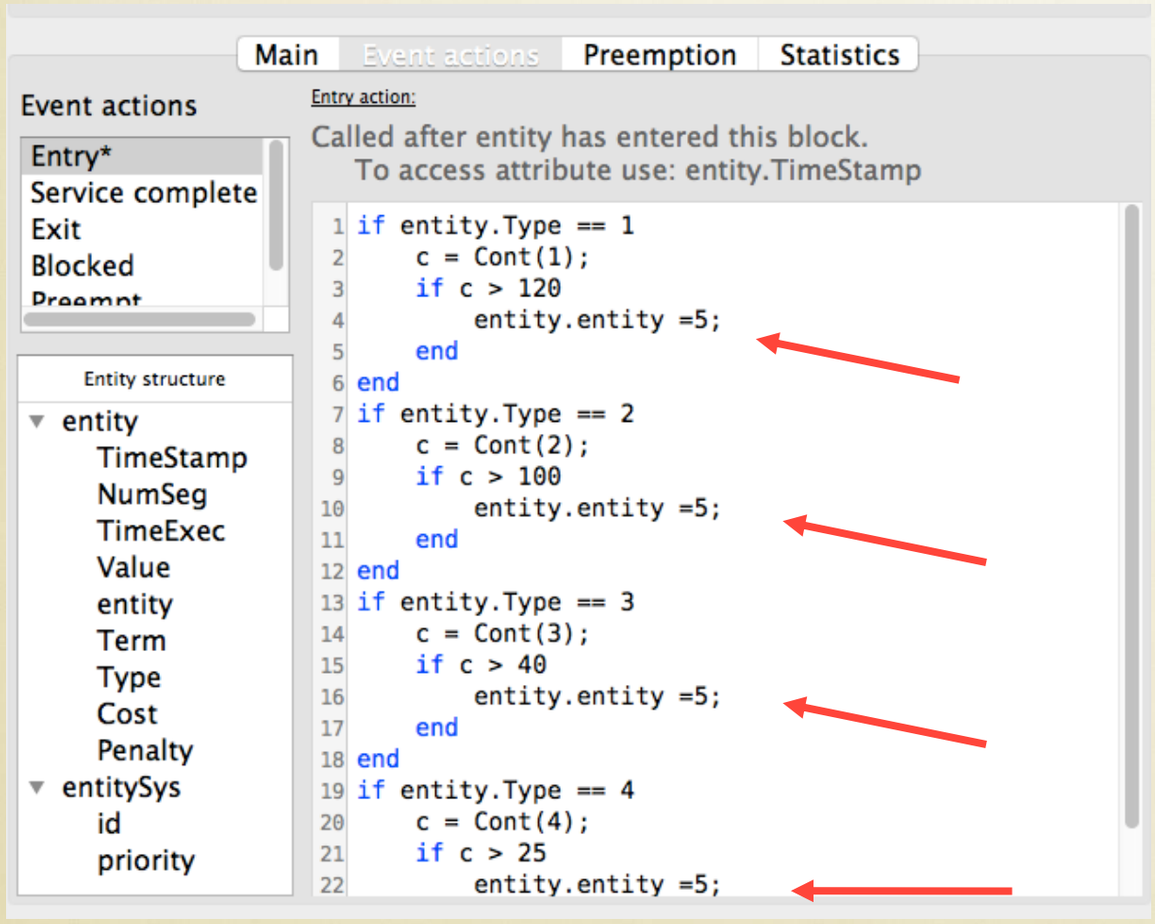
Блок ControlNumPart

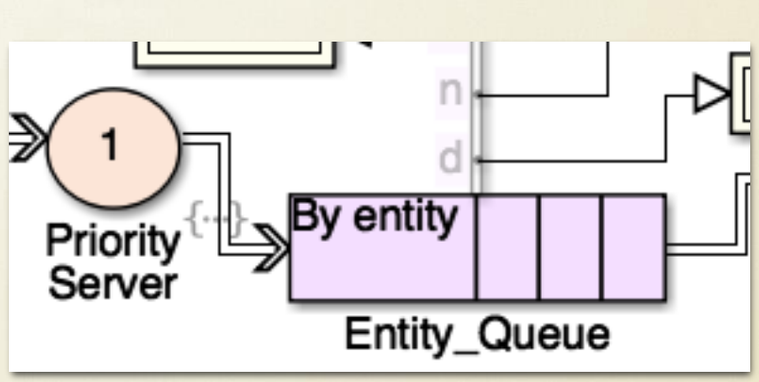




Sporadic Control #2 (ИА2)

Второй интеллектуальный агент реализован в виде кода встроенного в Priority Server и представляет механизм спорадического контроля на основании изменения приоритета задач. При возрастании среднего времени нахождения подзадачи в очереди, система увеличивает её приоритет, обеспечивая быстрое продвижение на выполнение.





Повышение приоритета подзадачи при возрастании среднего времени ожидания подзадачи в очереди.

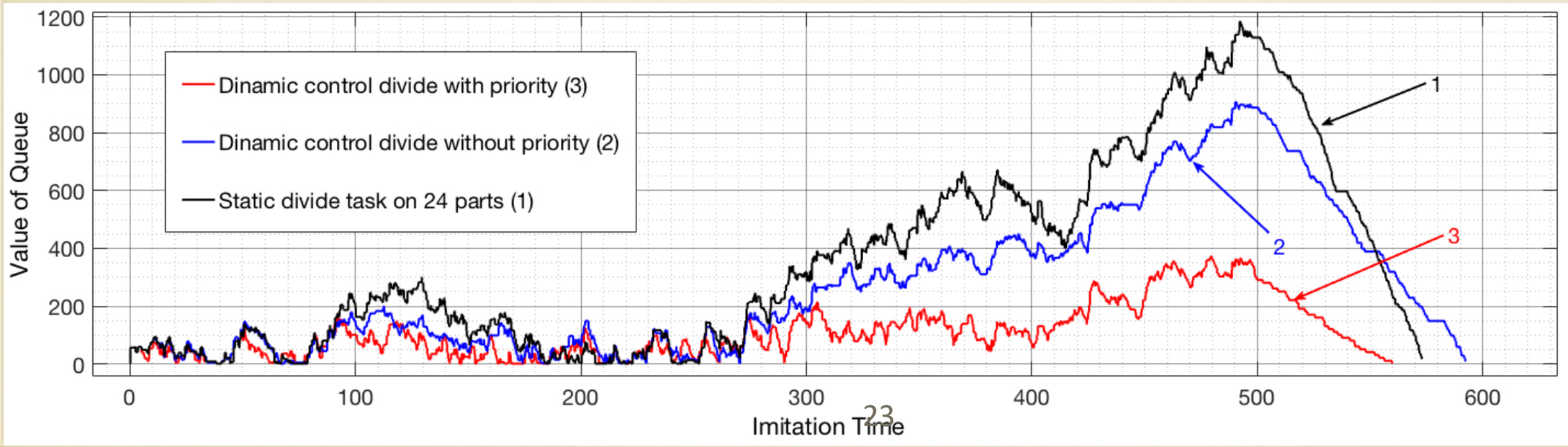
Функция Cont() вычисляет среднее время нахождения подзадач в очереди.

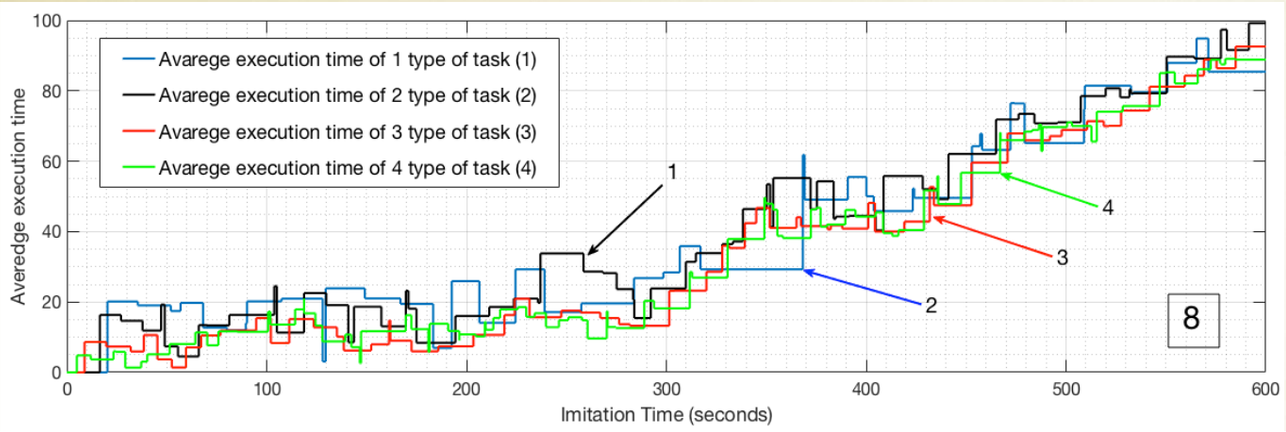
Пример результатов моделирования

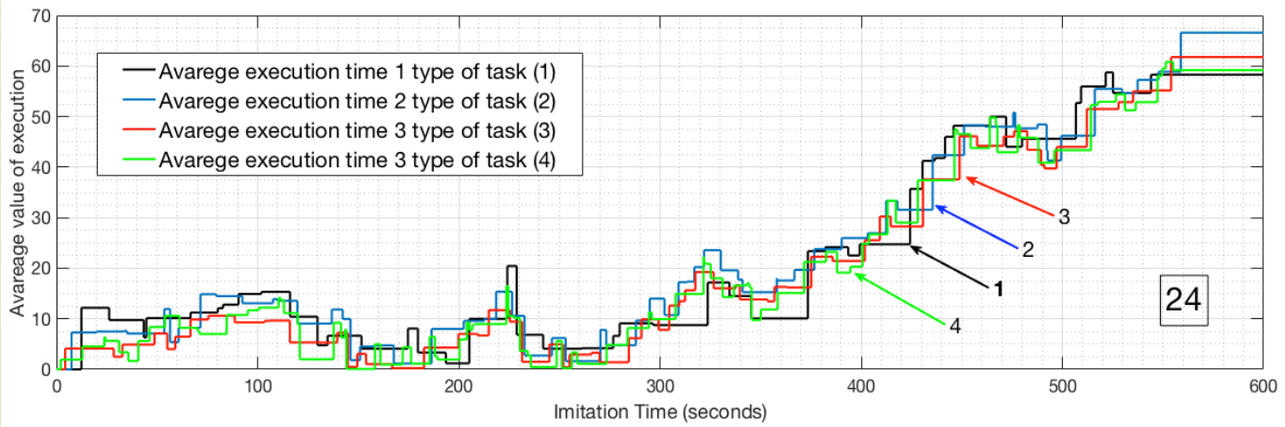
Таблица1 Экспоненциальный закон входного потока

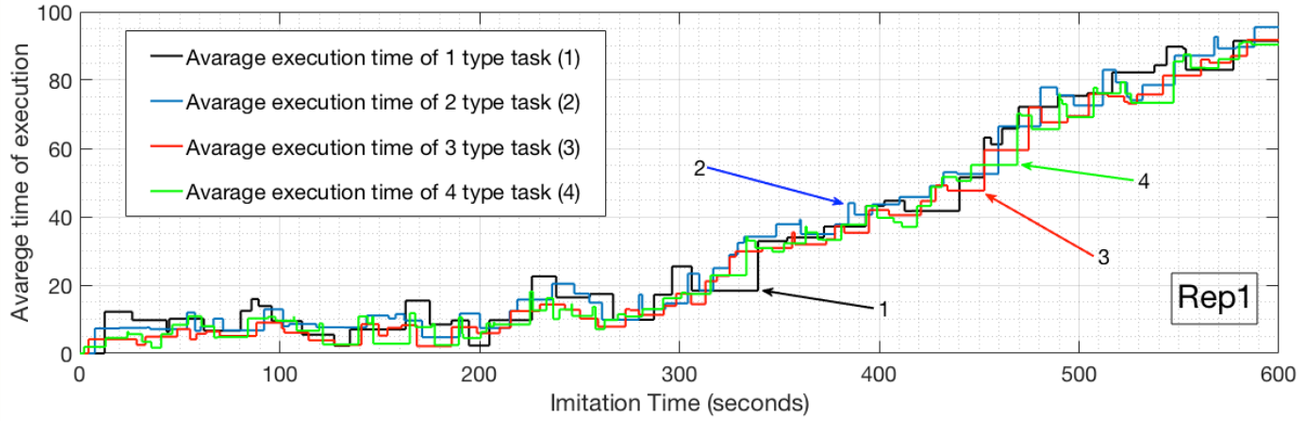
Rep	Q_{max}	LenQ	WaitQ	Avg_1	Avg_2	Avg_3	Avg_4
1	3	0.07	0	110.6	122.8	58.46	31.8
4	231	75.46	27.64	106.6	99.36	78.67	97.17
8	56.3	199	38.42	127.1	123,6	124.2	68.9
16	859	270.7	24.82	84.42	65.47	91.36	90.98
24	1186	35.8	21.34	71.38	64.02	77.21	76.46
var1	907	270	20.87	91.46	71.36	96.14	95.94
var2	373	110	7.468	74.93	46.43	8.074	2.74

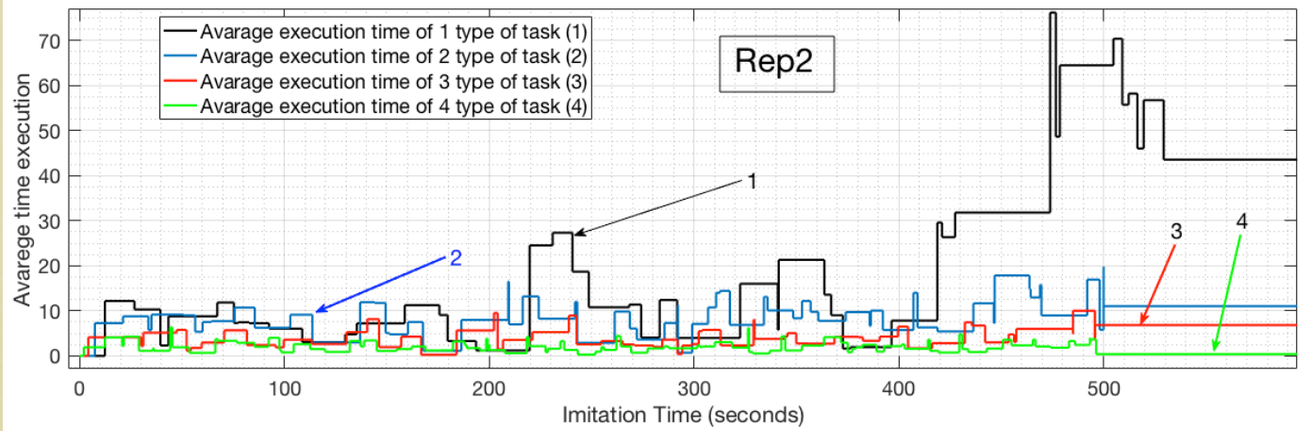
В таблице показаны результаты моделирования. Времена выполнения всех четырёх типов задач были неизменны и составляли 160, 128, 64 и 32 условных единиц. При экспоненциальном законе поступления задач в системы были выбраны следующие значения λ = 12, 10, 6, 2;











Среднее время выполнения

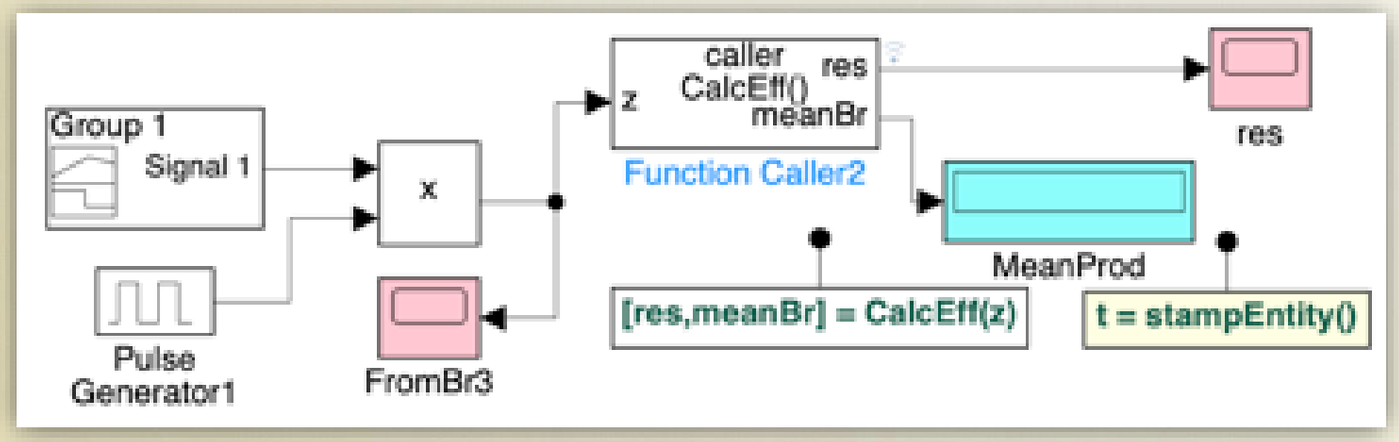
Верхний график показывает поведение среднего времени выполнения при отсутствии регулирования и разбиении задачи на 8 подзадач. Второй график - аналогично, но при разбиении задачи на 24 части.

Второй график снизу показывает поведение среднего времени выполнения при динамическом разбиении задачи, но без управления приоритетами. Самый нижний график показывает поведение среднего времени выполнения при динамическом разбиении и приоритетном спорадическом управлении задачами.

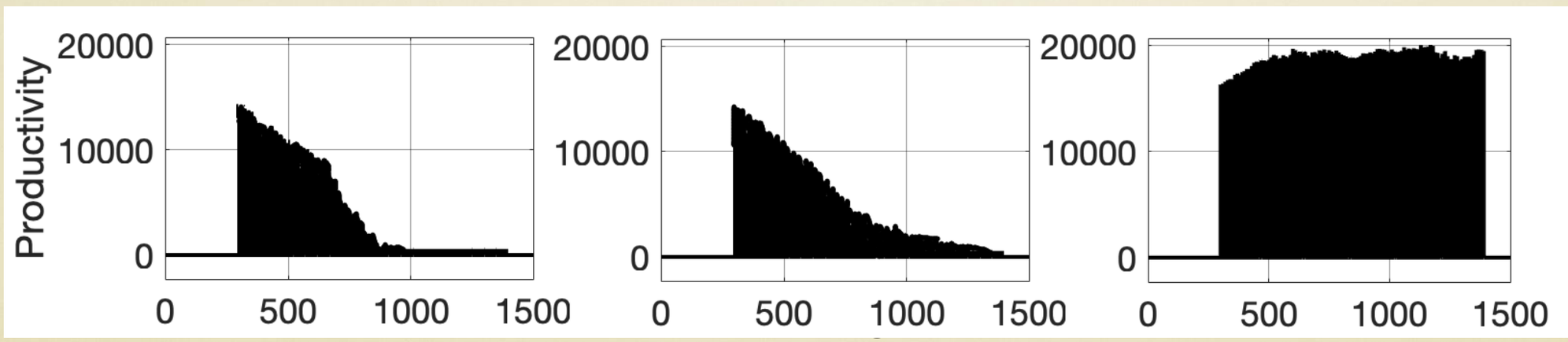
Из этих графиков видно, что только последний случай обеспечивает заданное директивное время выполнения с большим запасом. При этом размеры очереди существенно уменьшились.

Блок оценки эффективности Prod

Для введения численного параметра производительности системы введем некоторые определения. Предположим, что имеется *N* параллельно работающих независимых серверов. Пусть на входе системы имеется поток задач, каждая из которых имеет единичное время обслуживания. Тогда в единицу времени можно обслужить *N* задач, а за промежуток времени *t* число решенных задач будет *Nt.* Таким образом, можно определить среднюю продуктивность системы как сумму ДВВ решенных задач за некоторый промежуток времени.



В данном случае использовался промежуток времени, равный 300 единицам условного времени (далее будем считать их секундами). Для расчетов в модель встроен блок Prod.



Влияние отказов серверов

- при возникновении отказа сервера в процессе выполнении некоторой подзадачи она возвращается в очередь на повторное выполнение;
- серверу, на котором возник отказ, необходимо некоторое время для перезагрузки, в течение которого он заблокирован для приема задач на выполнение;
- после перезагрузки сервер продолжает принимать и обрабатывать поступающие задачи;
- отказы серверов в настоящей модели случаются по закону равномерного распределения в интервале [0, 1]

Экспоненциальное распределение

P	*M*	*S*	*Eff, %*	*InQ*	*LenQ*	*DepQ*	*QWait*	*Tm*	*Err*
0,997	14 762,8	0	100	0	45,5	29 557	2,155	1402	72
0,9	14 715,6	0	100	0	60,1	31 712	2,660	1416	2847
0,8	14 701,8	0	100	0	80,52	33 856	3,388	1433	5857
0,75	14 609,2	0	100	0	103,8	34 469	4,291	1431	7823
0,7	14 312,3	6 (4, 2, 0, 0)	99,4	0	160	31 617	7,299	1456	8652

Результаты с использованием 24 серверов

№	*M*	*S*	*Eff, %*	*InQ*	*LenQ*	*DepQ*	*QWait*	*Tm*	*Err*
1	14 672	0	100	0	80,18	27 822	4,064	1419	87
2	18 597	0	100	0	145,4	29 374	7,098	1439	87

Преимущества такого подхода

Предложенная схема управления прохождением задач в облачных системах имеет ряд преимуществ по сравнению с существующими классическими схемами управления.

- Во-первых, в такой системе отсутствует планировщик, который при больших входных потоках и разного типа задач будет требовать дополнительного времени на расчет оптимальной стратегии обслуживания.
- Во-вторых, как видно из таблиц и рисунков появился большой запас по гарантированному ДВВ, что позволяет дополнительно нагрузить систему, и тем самым повысить её эффективность.
- В-третьих, имеется возможность на основе элементов ИА внесения “волюнтаризма” в управлению системой, когда в силу каких либо причин необходимо в кратчайшие сроки выполнить некоторую задачу, не нарушая работы всей системы.

Преимущества такого подхода (2)

Кроме того:

- Имеется возможность исследовать систему посредством варьирования её параметров и определение её максимальной эффективности. То есть определять её максимальную производительность в различных режимах.
- Имеется возможность настраивать систему на максимальную эффективность посредством изменения системы приоритетов.
- Данная система является весьма гибкой и при незначительных изменениях может быть перестроена на выполнения других задач, например оптимизация работы инструментального цеха или оптимизации логистических операций.
- Большим преимуществом данной системы является отсутствие каких либо целевых функций, на вычисление который требуется некоторое время.
- В таких системах имеется возможность контролирования директивного срока выполнения задач. что является существенным преимуществом таких систем.

Литература

1.Малашенко Ю.Е., Назарова И.А. Модель управления разнородными вычислительными заданиями на основе гарантированных оценок времени выполнения. // Изв. РАН. ТиСУ. 2012. No 4. С. 29-38.

2.Купалов-Яропол к И.К., МалашенкоЮ.Е., НазароваИ.А. и др. Модели и программы для системы управления ресурсоемкими вычислениями. М.: ВЦ РАН, 2013. http://www.ccas.ru/depart/malashen/papper/ronzhin2012preprint.pdf.

3.Голосов П.Е., Гостев И.М. О некоторых имитационных моделях планировщиков операционных систем // Телекоммуникации. 2021 No 6, ст. 10-21.

4.Голосов П.Е., Гостев И.М. Об имитационном моделировании функционирования операционной системы с вытесняющим планированием // Телекоммуникации. 2021. No 8. С. 2–22.

5.Голосов П.Е., Гостев И.М. Имитационное моделирование серверов с прерываниями в больших многопроцессорных системах // Известия вузов. Приборостроение. 2021. Т. 64. No 11. С. 879–886. https://doi.org/10.17586/0021-3454-2021-64-11-879-886.

6.Golosov P.E., Gostev I.M. About one cloud computing simulation model // Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings. 2021. P. 9416100. https://doi.org/10.1109/IEEECONF51389.2021.9416100

7.Golosov P.E., Gostev I.M. Cloud computing simulation model with a sporadic mechanism of parallel problem solving control. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2022, vol. 22, no. 2, pp. (in Russian).