

Моделирование потока запросов на чтение и запись данных

Докладчик: Морозов Сергей Валерьевич

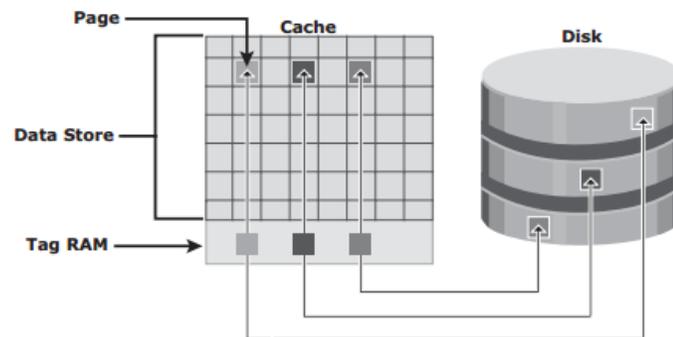
usd.icq@gmail.com

Научный руководитель: д. ф.-м. н., проф. Нестеров В.М.

Санкт-Петербургский Государственный Университет, Математико-Механический
факультет, Кафедра Системного программирования, 2014 год

Актуальность

- Магнитные жесткие диски
 - Доступ к данным - несколько миллисекунд
- Твердотельные накопители
 - Доступ к данным – менее миллисекунды
- Магнитные диски + высокоэффективная кэш-память



Структура cache



HDD диск



SSD диск

Цель работы

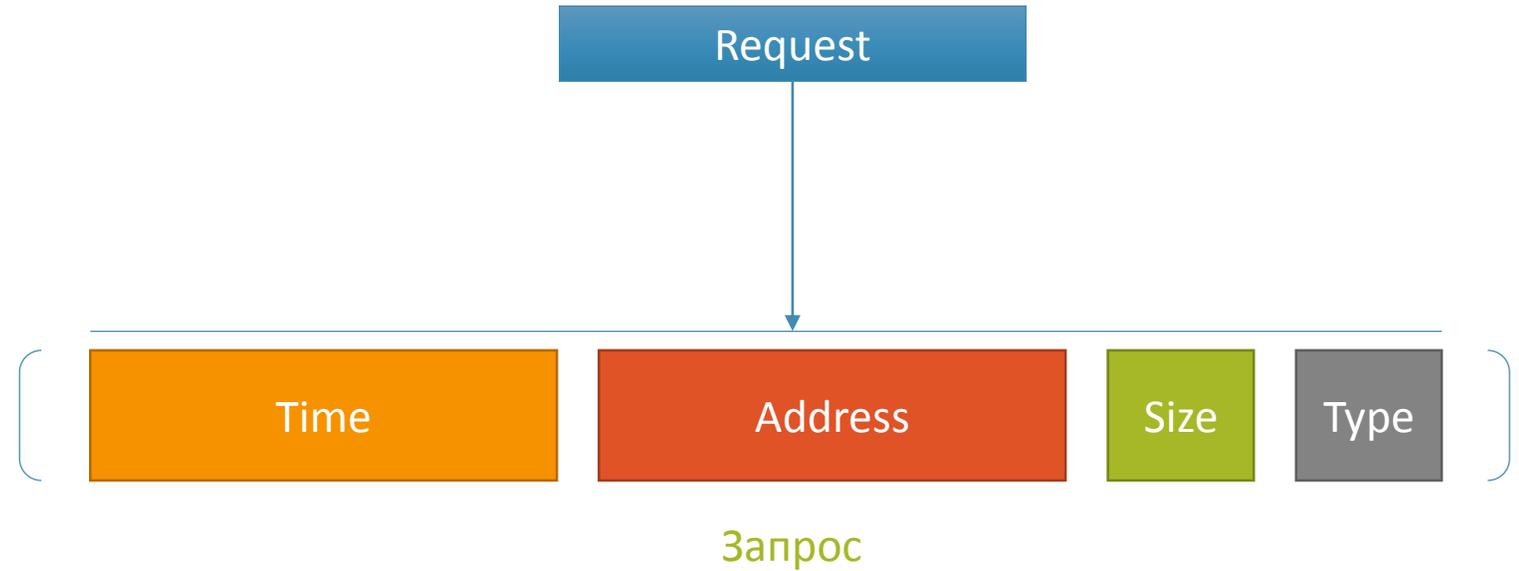
- Создание тестового пакета, моделирующего нагрузку на кэш-память запросами на чтение и запись данных
 - Изучить наиболее типичные потоки запросов и алгоритмы генерации нагрузки на кэш-память.
 - Реализовать известные алгоритмы генерации нагрузки на кэш-память с возможностью гибкого определения входных параметров.
 - Написать легко расширяемый для включения генерации разных потоков запросов тестовый пакет.

Существующие решения

- Подсистема кэширования СХД AVRORA
- Oracle's Sun Unified Storage Simulator
 - Виртуальная система хранения данных
- GigaNet Systems VirtualSAN
 - Система тестирования оптоволоконной сети
- SimSANs
 - Проектирование и симуляция систем хранения данных

Моделируемые данные

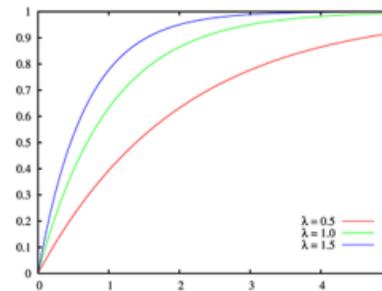
- Запрос
 - Время запроса
 - Адрес
 - Размер запроса
 - Тип запроса (R/W)



Компоненты запроса: время

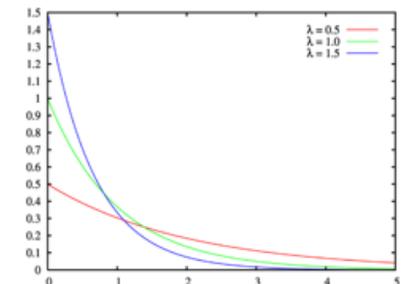
- Время транзакции:
 - $t_1 = 0, \Delta t_0 = 0$
 - $t_i = t_{i-1} + \Delta t_i, i > 1$
- $\Delta t_i, i > 1$ распределено в соответствии с экспоненциальным законом
- Параметр экспоненциального распределения $\lambda > 0$ называется параметром скорости (rate parameter)

$$F_X(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$



Функция экспоненциального распределения и ее график

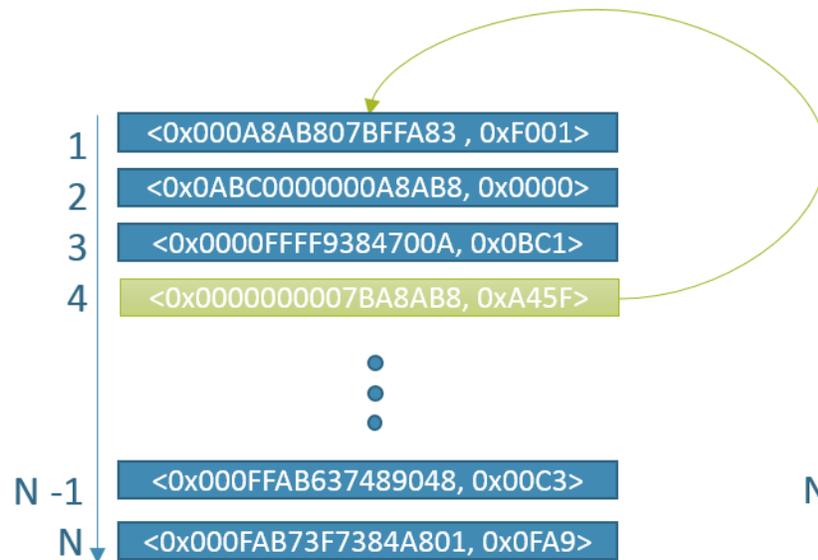
$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$



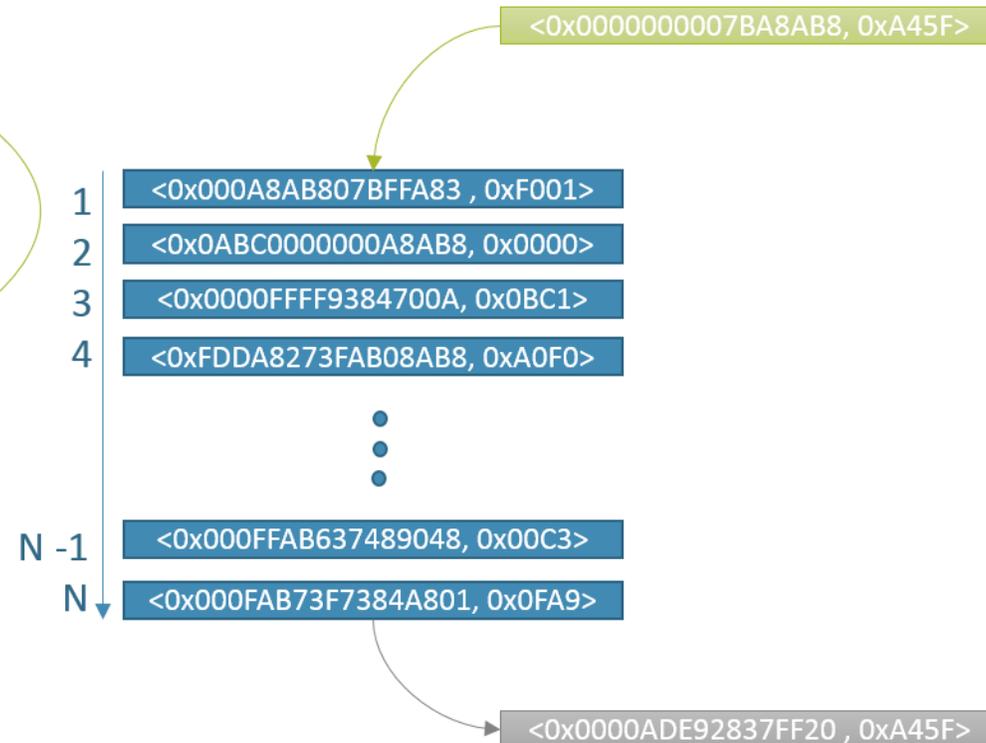
Плотность экспоненциального распределения и ее график

Кэш-память и LRU алгоритм

- Работа кэш-памяти моделируется при помощи LRU (Least Recently Used) алгоритма



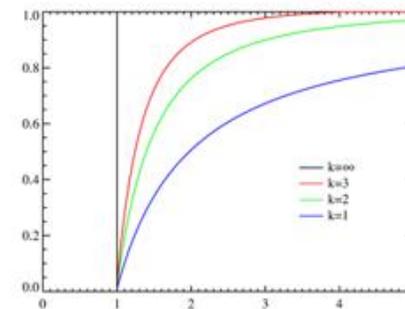
- Стековое расстояние – глубина элемента стека.



Компоненты запроса: адрес

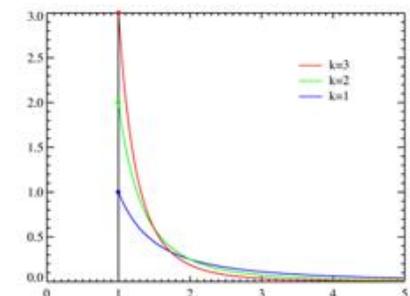
- Адрес транзакции
 - Стековые расстояния распределены в соответствии с распределением Парето
- Параметры распределения Парето
 - k – параметр положения (location parameter). Определяет минимально возможное значение $x \geq k$
 - a – параметр формы (shape parameter). Определяет «хвост» распределения Парето.

$$F(x) = 1 - \left(\frac{k}{x}\right)^a$$



Функция распределения Парето и ее график

$$f(x) = \frac{a k^a}{x^{a+1}}$$



Плотность распределения Парето и ее график

Компоненты запроса: размер и тип

- Размер
 - 512 B
 - 1 KB
 - 2 KB
 - 8 KB
 - 64 KB

Size	Probability of request size:				
	512 B	1 KB	2 KB	8 KB	64 KB
<input checked="" type="checkbox"/> 512 B					
<input type="checkbox"/> 1 KB					
<input checked="" type="checkbox"/> 2 KB					
<input type="checkbox"/> 8 KB	0,392	0	0,422	0	0,186
<input checked="" type="checkbox"/> 64 KB					

Clear

Вероятность появления запроса определенного размера

- Тип
 - Read
 - Write

Type	Probability of request type:	
	Read	Write
<input type="checkbox"/> Read		
<input checked="" type="checkbox"/> Write	0,73234	0,26766

Clear

Вероятность появления запроса определенного типа

Моделирование нагрузки

- Нагрузка представляется в виде последовательности запросов



- Интенсивность нагрузки изменяется во времени
 - Параметры распределений компонентов запроса меняются

Алгоритм изменения интенсивности нагрузки

	0	1	2	3
0	0	$\lambda_{0,1}$	$\lambda_{0,2}$	$\lambda_{0,3}$
1	$\lambda_{1,0}$	0	$\lambda_{1,2}$	$\lambda_{1,3}$
2	$\lambda_{2,0}$	$\lambda_{2,1}$	0	$\lambda_{2,3}$
3	$\lambda_{3,0}$	$\lambda_{3,1}$	$\lambda_{3,2}$	0

Матрица состояний

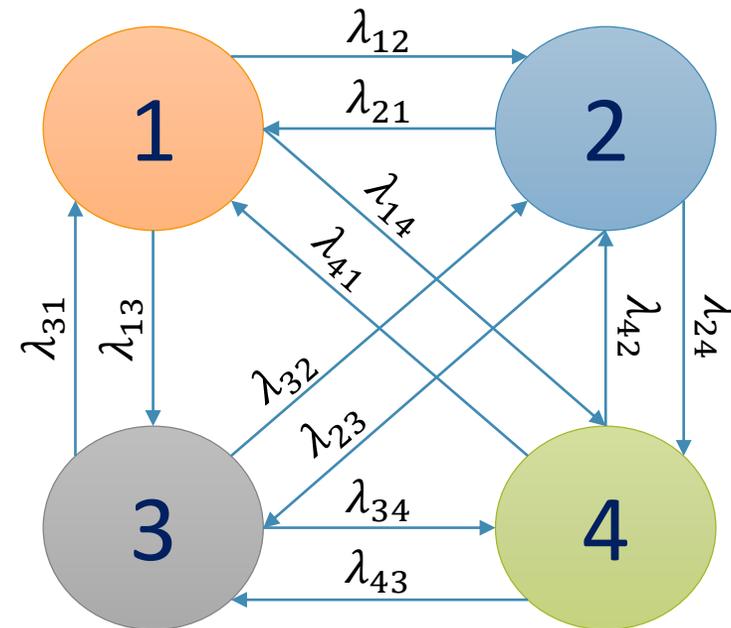


Схема работы алгоритма

Тестовый пакет

I/O Workload Modeling

General

States: 6
Requests: 1000000
Cache type: empty
Cache size: 65535

Matrix of states:

	1	2	3	4	5	6
1		3	2	3	1	0,78
2	7		3	10	0,6	6
3	6	9		5	8,57	9
4	1	5,5	4		0,153	3,0938
5	0,8	0,018	0,74	3		7
6	4	2	6	11,9384	4,832	

Time

Rate parameter of exponential distribution:

1	2	3	4	5	6
20	38,0394	51,393	30	68	13

Address

Location parameter of Pareto distribution:

1	2	3	4	5	6
1	0,9	1,1	2,4	1	0,832

Shape parameter of Pareto distribution:

1	2	3	4	5	6
103	29	72	40	20	38,1722

Size

Probability of request size:

512 B	1 KB	2 KB	8 KB	64 KB
0,73	0	0,1	0	0,17

Type

Probability of request type:

Read	Write
0,3647	0,6353

Generate [] Requests editor



C++ DLL

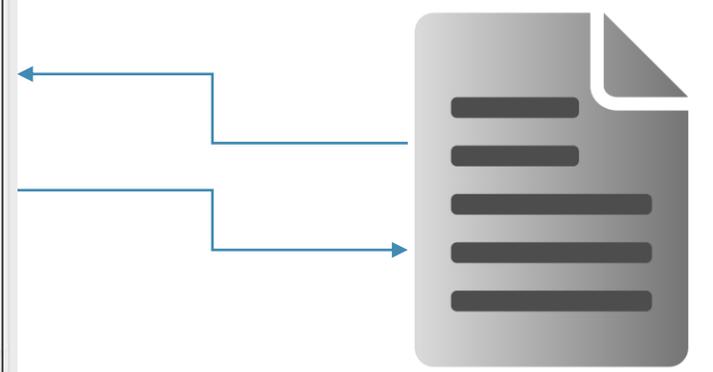
C# Windows Forms GUI

Редактор запросов

Request Editor

Open file C:\Users\Sergey\Documents\Projects\Workload_Modeling\records.txt Mode: read only

	Time	Address	Size	Type
59149	2173,17114424107	51408540B0432038h	0	0
59150	2173,42242578541	4E6A175425D2CE6Dh	0	0
59151	2173,45880361934	624A6EE8CEA731DFh	4	0
59152	2173,47689230979	722DBD56B54FC6CFh	0	0
59153	2173,53169314217	78D7B264E7026CF3h	0	0
59154	2173,71792951065	E0F09F3AC3F7D2D4h	4	0
59155	2174,0707709189	A3E24FACDE11D6E7h	0	0
59156	2174,07980596917	2D35427439E742A0h	0	0
59157	2174,13209868901	AE765FA81DF3728Eh	4	0
59158	2174,13598077643	F5AB9E9E9286E905h	0	1
59159	2174,17189178402	D8ADBCCDD46D30952h	0	0
59160	2174,20492645718	2A6270523B98FD74h	0	0
59161	2174,5062926999	40BB504528D2CAC9h	4	1
59162	2174,52686040466	127C4DDB7C3FE181h	1	0
59163	2174,55800904483	ACB77161F1C00EB5h	0	1
59164	2174,56752165163	4852671B8109BB2Bh	1	0
59165	2174,60971769072	61E30390E728D56Eh	1	0
59166	2174,62219538171	1F06DCE917A239A8h	4	0
59167	2174,6330129001	EE52341141F5DB73h	0	0
59168	2174,63832115866	EE8A7BF0731BE91Eh	1	0



Trace (binary file)

C# Windows Forms Requests Editor

Результаты

- Изучены типичные потоки запросов на чтение и запись данных и алгоритмы генерации нагрузки на кэш-память
- Выбраны наиболее часто встречающиеся статистические модели
- Разработан тестовый пакет
 - Библиотека, реализующая алгоритмы генерации потока запросов
 - Графический интерфейс для удобного практического использования библиотеки

Литература

- “Workload Modeling for Computer Systems Performance Evaluation”, G. Feitelson
- “Calculating Stack Distances Efficiently”, George Almasi, David A. Padua
- “Элементарный курс теории вероятностей и математической статистики”, А. Н. Бородин
- “Information Storage and Management” Second Edition, EMC Education Services

Морозов Сергей Валерьевич

usd.icq@gmail.com