



Universidad  
Carlos III de Madrid

# TECNOLOGÍAS DEL SECTOR FINANCIERO

## Práctica 1: Latencias

Andoni Alcelay

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS DEL SECTOR FINANCIERO:  
FINTECH

# Introducción

En este documento se explicarán los resultados obtenidos en la práctica 1 realizada con la librería HdrHistogram, para cálculos de latencias en cálculos de operaciones en Java. La práctica ha consistido en 3 ejercicios siendo cada uno más complejo que el anterior acabando por un cálculo multihilo.

## Práctica 1

Este ejercicio consiste en un bucle simple donde se ejecuta una operación 100.000 veces. En cada una de las operaciones se deben mostrar los resultados de esa operación y al final compararlos con el percentil 99 y 99.9, donde se ve que las latencias se disparan.

Aquí se pueden ver los resultados de unas de las ultimas veces que se ejecuta la operación:

Value	Percentile	TotalCount	1/(1-Percentile)
276.48	0.999877929688	99988	8192.00
337.92	0.999890136719	99990	9102.22
356.35	0.999902343750	99991	10240.00
454.66	0.999914550781	99992	11702.86
552.96	0.999926757813	99993	13653.33
622.59	0.999938964844	99994	16384.00
929.79	0.999945068359	99995	18204.44
1064.96	0.999951171875	99996	20480.00
1064.96	0.999957275391	99996	23405.71
1138.69	0.999963378906	99997	27306.67
1138.69	0.999969482422	99997	32768.00
1548.29	0.999972534180	99998	36408.89
1548.29	0.999975585938	99998	40960.00
1548.29	0.999978637695	99998	46811.43
1736.70	0.999981689453	99999	54613.33
1736.70	0.999984741211	99999	65536.00
1736.70	0.999986267090	99999	72817.78
1736.70	0.999987792969	99999	81920.00
1736.70	0.999989318848	99999	93622.86
5144.58	0.999990844727	100000	109226.67
5144.58	1.000000000000	100000	
#[Mean = 52.23, StdDeviation = 34.77]			
#[Max = 5144.58, Total count = 100000]			
#[Buckets = 10, SubBuckets = 256]			

Y aquí al final del todo los resultados del número de veces que se ha ejecutado la operación, el tiempo que le ha llevado y sus percentiles 99 y 99.9:

#[Min = 0.384, Time = 224.99]

99 percentile = 102399

99.9 percentile = 118271

## Práctica 2

Este ejercicio es parecido al anterior. La diferencia es que esta vez se puede observar cómo la máquina virtual va cogiendo ritmo y su optimizador hace que cada vez tarde menos en realizar esa operación. Para eso disponemos, de una operación que se realiza en bucle 200.000 veces. Al final de esas 200.000 vueltas se sacan los resultados de lo que ha tardado en realizar esta operación, y para simular que la maquina “coge temperatura” se repite esto 40 veces

En las primeras ejecuciones se puede ver que los tiempos de las últimas ejecuciones no tienen nada que ver. Los tiempos bajan considerablemente:

Ejecución número 3:

Min = 5120

Max = 175103

Mean = 6237.76896

99 percentile = 24575

99.9 percentile = 33791

Repetitions = 3

Ejecución número 30:

Min = 3584

Max = 120831

Mean = 5589.69856

99 percentile = 24575

99.9 percentile = 35839

Repetitions = 30

Tal y como se ve en los dos ejemplos, el tiempo de ejecución del número 3, la media está en unos 6200, mientras que en la ejecución número 30 ha bajado hasta 5589. Aun así los

percentiles altos siguen siendo muy parecidos. En el resto se ve la bajada de la latencia debido a que la máquina virtual ha optimizado el proceso para esto.

## Práctica 3

Obtener la distribución por percentiles de latencia sobre una operación para todos los hilos de ejecución. Comparar con la latencia acumulada teniendo en cuenta un ratio esperado máximo de llamadas.

Para ello se han creado dos instancias de Histogram, usando la clase ConcurrentHistogram, ya que al haber varios hilos, era necesario que la instancia fuera 'thread-safe'.

En el resultado se puede observar como hay dos outputs, uno donde se ve la latencia de un hilo, y por otro lado la latencia acumulada en todos los hilos. Para un hilo se ve que apenas hay diferencia mientras que para múltiples hilos esta diferencia crece:

Output para un hilo:

Latencia 1 hilo	Latencia acumulada
#[Mean = 10.49000, StdDeviation = 0.88876]	#[Mean = 485.84000, StdDeviation = 275.16819]
#[Max = 16.00000, Total count = 100]	#[Max = 962.00000, Total count = 100]
#[Buckets = 1, SubBuckets = 262144]	#[Buckets = 1, SubBuckets = 262144]

Output para multiples hilos (8 hilos):

Latencia 1 hilo	Latencia acumulada
#[Mean = 69.42500, StdDeviation = 81.77270]	#[Mean = 87.92625, StdDeviation = 75.46584]
#[Max = 623.00000, Total count = 800]	#[Max = 623.00000, Total count = 800]
#[Buckets = 1, SubBuckets = 262144]	#[Buckets = 1, SubBuckets = 262144]