



Práctica 2

Sistemas distribuidos

Maite Bernaus, David Fernández
FEHCA DE ENTREGA 29/05/2019

Índice

1.	Enlace al repositorio GitHub	2
2.	Decisiones de diseño	2
	Diseño general.....	2
	Comentarios adicionales	3
	Aspectos a destacar de la asignatura	3
3.	Diagrama de la implementación	4
4.	Validación	5

1. Enlace al repositorio GitHub

<https://github.com/Hilnur/SDPRact2>

2. Decisiones de diseño

Diseño general

El sistema implementado está formado por un nodo de control ("maestro") y distintos nodos ("esclavos").

El objetivo de la práctica es que los nodos "esclavos" generen un número aleatorio entre 0 y 1000 y lo envíen a los otros nodos "esclavos" generando en cada nodo una lista con todos los números generados por los distintos nodos – pero estas listas han de mantener consistencia entre todos los nodos (que en este caso implica estar en el mismo orden). Para ello, hemos diseñado un sistema de exclusión mutua en el cual los nodos esclavos han de pedir permiso al maestro para poder escribir en la lista de todos los nodos simultáneamente de forma síncrona.

El algoritmo implementado tiene dos fases: organización, y trabajo.

En la fase de organización, el nodo maestro simplemente espera a recibir en su cola un mensaje de cada nodo esclavo con su identificador de cola, y añade estos identificadores a su lista de nodos disponibles. Seguidamente el nodo "maestro" realiza un shuffle de los distintos id recibidos para aleatorizar el orden en que los nodos recibidos recibirán permiso. Esto no es estrictamente necesario, pero lo consideramos útil para asegurar que si los elementos están en orden es por el algoritmo de exclusión mutua, no por la casualidad de que los nodos estén en orden.

En la fase de trabajo, el nodo maestro va dando permiso a los elementos de su lista de identificadores uno a uno, esperando a que el nodo acabe antes de darle permiso al siguiente. Cada vez que un nodo recibe permiso de escritura, envía el valor que se ha de escribir a todos los nodos esclavos como un mensaje de escritura, y una vez esta operación se ha realizado, avisa al nodo maestro de que su operación ha terminado. Esto permite sincronizar las operaciones de "escritura".

Para la comunicación entre nodos se utiliza el servicio de colas RabbitMQ con diferentes tipos de mensajes:

- Para el nodo central se ha creado una cola de **peticiones** a la cual enviarán mensajes los esclavos:
 - Durante la fase de organización, cada esclavo envía su identificador de cola a la cola de peticiones para que el nodo maestro sepa que el nodo está activo y esperando permiso de operación, y a donde tendrá que enviar este permiso.
 - Durante la fase de trabajo, cuando un nodo haya terminado de escribir en todos los nodos, envía un mensaje 'done' al maestro para indicar que ha terminado de escribir, con lo cual el maestro puede dar permiso de escritura a otro nodo.
- Cada uno de los nodos esclavo crea su propia cola individual, que está unida a un exchange de tipo fanout. Los diferentes tipos de mensaje que se envían a esta cola son:
 - 'result': mensaje de escritura. Estos mensajes son enviados por los nodos esclavos a todos los nodos esclavos usando un exchange de tipo fanout simultáneamente. Cuando el nodo recibe un mensaje de tipo result, añade su contenido a su lista de elementos.
 - 'permission': mensaje de permiso de escritura. Estos mensajes son enviados de forma individual por el nodo maestro a la cola del nodo esclavo. Indican que el nodo receptor tiene permiso para realizar la "escritura" (enviar un result a los demás).

- 'finish': mensaje de finalización. El nodo maestro lo envía a todos los nodos esclavos cuando todos los nodos han terminado su tarea, para que cierren sus conexiones, purguen sus colas, y devuelvan sus resultados.

Comentarios adicionales

- Para simplificar el algoritmo, en lugar de escribir por su cuenta en su propia lista, los nodos esclavos simplemente envían sus results a todos los nodos, incluyéndose a sí mismos. De esta forma, no es necesario hacer control de duplicaciones.
- Hemos puesto un conjunto de prints que nos permitan ver la correcta evolución de los algoritmos utilizando el log de IBM Functions, ya que debugar un programa distribuido es complicado y queríamos facilitarnos la tarea.
- El diseño es un tanto aparatoso, con un doble bucle, dobles funciones anidadas dentro de la función master, y variables nonlocal. Nos hemos visto forzados a esto por esquema de funcionamiento de Rabbit, donde las funciones de consumición que incluyen espera activa no devuelven los mensajes obtenidos de la cola, sino que cada mensaje se trata con una función independiente cada vez que se recibe un mensaje.
- Debido al diseño realizado no se pueden crear más de 19 nodos esclavos, ya que cada nodo crea su propia conexión y el máximo de conexiones en nuestra cuenta de rabbit es de 20 (y de hecho, no estamos enteramente seguros de por qué el programa funciona con 19 esclavos, ya que pywren debería tomar una conexión, haciendo 1 master + 19 slaves +1 pywren = 21 conexiones, pero nuestras pruebas han dado resultados correctos con 19 esclavos).
- El programa ya incluye las credenciales para nuestra propia cola, pero puede usarse fácilmente con otra cola simplemente reemplazando la variable url en las funciones server y mapper.

Aspectos a destacar de la asignatura

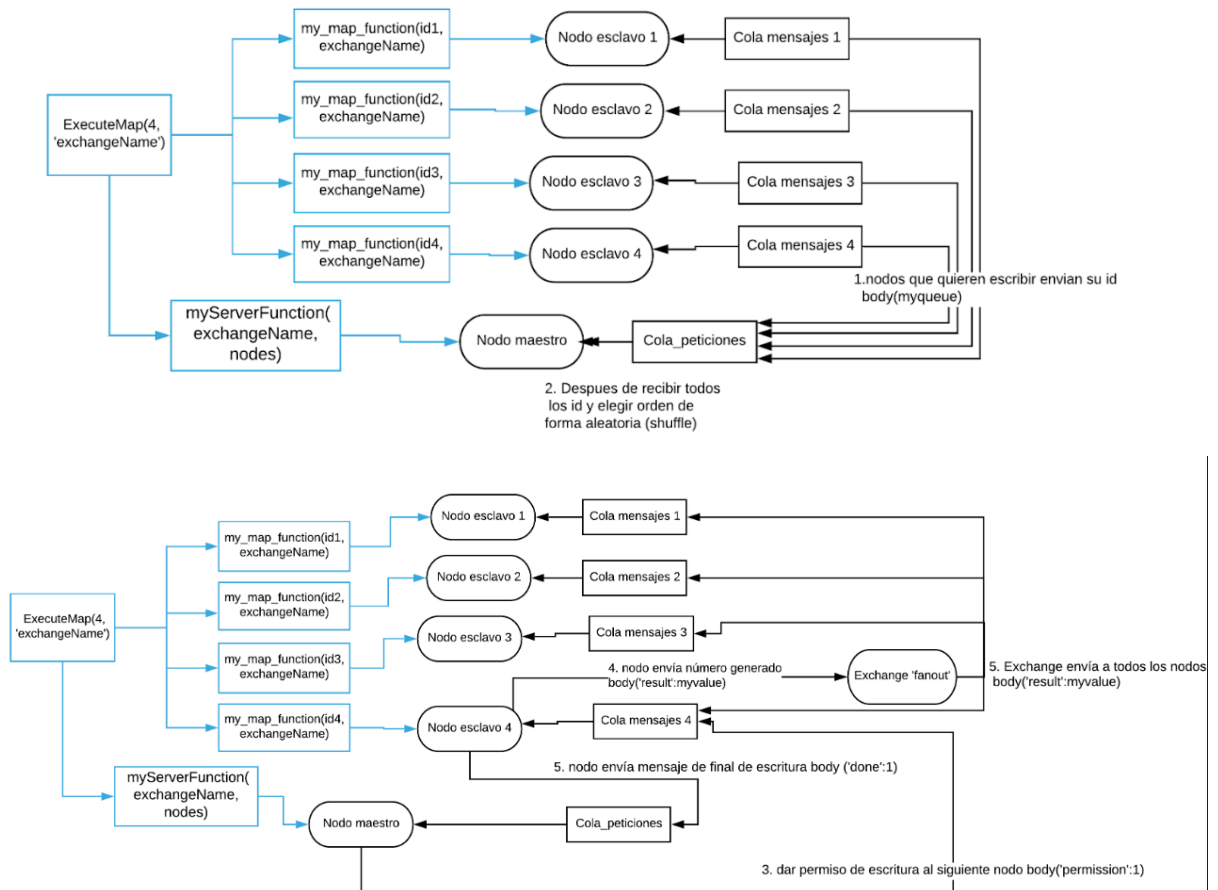
La sincronización se consigue gracias al sistema de mensajes permission-done que crea un esquema de exclusión mutua, de forma que solo un nodo pueda tener permiso de escritura en un momento determinado.

Esta sincronización es lo que proporciona al sistema consistencia haciendo que todos los nodos reciben en el mismo orden los mensajes.

El propio objetivo de la práctica en sí proporciona replicación, ya que la idea es que por tanto gracias al sistema de colas las lista generadas sean iguales (cada uno de los nodos tiene la misma información = replicación de datos).

Por lo que respecta a tolerancia a fallos, debido a falta de tiempo no se ha podido implementar ningún sistema para detectar si falla algún nodo. Sin embargo, en caso de que rabbitMQ disponga de una funcionalidad de timeout (de lo cual no estamos seguros, ya que encontrar documentación actualizada para Rabbit ha resultado más complicado de lo que esperábamos), sería bastante sencillo añadir una funcionalidad al nodo central para detectar si el nodo al que se le ha enviado un mensaje de permiso no responde en demasiado tiempo, y en tal caso simplemente guardar el nodo errado en una lista de nodos caídos para futuro diagnóstico, eliminar su identificador de la lista de nodos de trabajo, y continuar con el siguiente. De esta forma se conseguiría que el programa no se bloqueara y proporcionara resultados casi completos en caso de caída de cualquier nodo excepto el maestro.

3. Diagrama de la implementación



Finalmente, cuando todos los nodos han escrito el nodo maestro envía un mensaje body ("finished":1) al exchange el cual lo envía a todas las colas de mensajes de los distintos nodos esclavos para que finalicen su ejecución.

4. Validación

Para analizar el correcto funcionamiento de la práctica que ha verificado que las listas de los distintos nodos sean la misma corroborando así la consistencia del sistema. Se han realizado las pruebas por triplicado para validar su correcto funcionamiento para 1, 2, 4, 8, 16, 19 nodos donde como se observa en los resultados se consigue sincronización , consistencia y redundancia en el sistema.

REPLICA	NUM NODOS	ID NODO	LISTA
1	1	0	368
2	1	0	681
3	1	0	531

REPLICA	NUM NODOS	ID NODO	LISTA
1	2	0	78, 479
		1	78, 480
2	2	0	532, 88
		1	72, 384
3	2	0	72, 384
		1	72, 385

REPLICA	NUM NODOS	ID NODO	LISTA
1	4	0	678, 601, 827, 206
		1	678, 601, 827, 207
		2	678, 601, 827, 208
		3	678, 601, 827, 209
2	4	0	453, 813, 755, 183
		1	453, 813, 755, 184
		2	453, 813, 755, 185
		3	453, 813, 755, 186
3	4	0	819, 526, 674, 462
		1	819, 526, 674, 463
		2	819, 526, 674, 464
		3	819, 526, 674, 465

REPLICA	NUM NODOS	ID NODO	LISTA
1	8	0	331, 550, 583, 668, 120, 826, 44, 157
		1	331, 550, 583, 668, 120, 826, 44, 157
		2	331, 550, 583, 668, 120, 826, 44, 157
		3	331, 550, 583, 668, 120, 826, 44, 157
		4	331, 550, 583, 668, 120, 826, 44, 157
		5	331, 550, 583, 668, 120, 826, 44, 157
		6	331, 550, 583, 668, 120, 826, 44, 157
		7	331, 550, 583, 668, 120, 826, 44, 157
2	8	0	778, 907, 889, 951, 261, 273, 179, 202
		1	778, 907, 889, 951, 261, 273, 179, 202
		2	778, 907, 889, 951, 261, 273, 179, 202
		3	778, 907, 889, 951, 261, 273, 179, 202
		4	778, 907, 889, 951, 261, 273, 179, 202
		5	778, 907, 889, 951, 261, 273, 179, 202
		6	778, 907, 889, 951, 261, 273, 179, 202
		7	778, 907, 889, 951, 261, 273, 179, 202
3	8	0	513, 594, 674, 761, 205, 971, 25, 639
		1	513, 594, 674, 761, 205, 971, 25, 639
		2	513, 594, 674, 761, 205, 971, 25, 639
		3	513, 594, 674, 761, 205, 971, 25, 639
		4	513, 594, 674, 761, 205, 971, 25, 639
		5	513, 594, 674, 761, 205, 971, 25, 639
		6	513, 594, 674, 761, 205, 971, 25, 639
		7	513, 594, 674, 761, 205, 971, 25, 639

REPLICA	NUM NODOS	ID NODO	LISTA
1	16	0	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		1	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		2	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		3	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		4	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		5	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		6	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		7	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		8	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		9	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		10	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		11	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		12	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		13	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		14	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
		15	984, 28, 180, 935, 844, 291, 880, 465, 715, 870, 219, 91, 234, 299, 194, 133
2	16	0	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		1	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		2	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		3	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		4	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		5	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		6	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		7	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		8	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		9	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		10	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		11	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		12	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		13	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		14	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
		15	273, 707, 955, 982, 229, 765, 895, 369, 166, 932, 220, 40, 353, 427, 901, 144
3	16	0	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		1	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		2	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		3	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		4	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		5	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		6	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		7	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		8	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		9	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		10	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		11	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		12	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		13	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		14	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578
		15	798, 196, 61, 674, 480, 496, 282, 702, 803, 477, 848, 984, 623, 416, 488, 578

REPLICA	NUM NODOS	ID NODO	LISTA
1	19	0	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		1	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		2	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		3	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		4	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		5	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		6	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		7	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		8	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		9	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		10	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		11	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		12	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		13	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		14	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		15	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		16	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		17	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
		18	585, 436, 202, 600, 480, 405, 291, 861, 954, 454, 854, 387, 243, 214, 969, 868, 18, 811, 157
2	19	0	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		1	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		2	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		3	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		4	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		5	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		6	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		7	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227,

			610, 519, 347, 715, 760, 18
		8	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		9	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		10	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		11	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		12	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		13	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		14	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		15	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		16	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		17	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
		18	885, 168, 515, 846, 611, 291, 64, 979, 250, 472, 322, 775, 227, 610, 519, 347, 715, 760, 18
3	19	0	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		1	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		2	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		3	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		4	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		5	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		6	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		7	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		8	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		9	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		10	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		11	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		12	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		13	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		14	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
		15	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900,

		705, 61, 395, 879, 940, 840
	16	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
	17	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840
	18	319, 533, 921, 671, 367, 187, 996, 419, 829, 615, 515, 474, 900, 705, 61, 395, 879, 940, 840