

Aprendizaje Automático sobre Grandes Volúmenes de Datos

Clase 9

Pablo Ariel Duboue, PhD

Universidad Nacional de Córdoba,
Facultad de Matemática, Astronomía y Física



Material de lectura

- Clase pasada:
 - Capítulo 2 y 4 del Owen et al. (2012)
- Ésta clase:
 - MapReduce: Simplified Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat
 - OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004
 - The Google File System by Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
 - 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003.

Preguntas

- k-Means y features de tipo enumeración
 - ¿Cómo definir el centroide para features categoriales (e.g., "frío, calor, templado")?
 - En ese caso el centroide no necesita tener la misma representación que los elementos normales
 - Distribución de probabilidad sobre cada categoría
 - Hay dos funciones de distancia: distancia entre instancias y distancia entre instancia y centroide
- Evaluación de sistemas de recomendación
 - Evaluar sistemas de recomendación es todavía un problema abierto
 - Se utilizan técnicas donde se mantiene un porcentaje de preferencias oculto y después se usa Precision/Recall para evaluar
 - Técnica "injusta" y depende de cada partición

Recordatorio

- El sitio Web de la materia es <http://aprendizajengrande.net>
 - Allí está el material del curso (filminas, audio)
- Leer la cuenta de Twitter <https://twitter.com/aprendengrande> es obligatorio antes de venir a clase
 - Allí encontrarán anuncios como cambios de aula, etc
 - No necesitan tener cuenta de Twitter para ver los anuncios, simplemente visiten la página
- Suscribirse a la lista de mail en aprendizajengrande@librelist.com es optativo
 - Si están suscriptos a la lista no necesitan ver Twitter
- Feedback para alumnos de posgrado es obligatorio y firmado, incluyan si son alumnos de grado, posgrado u oyentes
 - El "resumen" de la clase puede ser tan sencillo como un listado del título de los temas tratados

C mputo Distribuido

- MapReduce
- Teorema CAP
- Operaciones Matriciales Distribuidas
- Descenso por el Gradiente (B squeda distribuida)
- Algoritmos actualizables, Colas, shared memory
- Paralelizando Algoritmos de Aprendizaje Autom tico

Modelo de Cluster de Máquinas

- Máquinas autónomas con espacio de disco local
 - Sin memoria compartida
- Con una topología de red jerárquica
 - Máquinas en un mismo switch
 - Máquinas en un mismo rack
- Máquinas de poca confiabilidad
 - Fallas en el sistema son algo no solo probable si no esperable

MapReduce

- Modelo de c  puto que simplifica el uso de clusters
- Computa una funci  n $f(\{(k_{in}, v_{in})\}) \rightarrow \{(k_{out}, list(v_{out}))\}$
 - $map(k_{in}, v_{in}) \rightarrow list(k_{out}, v_{int})$
 - $reduce(k_{out}, list(v_{int})) \rightarrow list(v_{out})$

Teorema CAP

- Cuando estamos en un ambiente distribuido, de estas tres caracter  sticas s  lo puedes escoger dos:
 - Consistencia
 - Disponibilidad
 - Tolerancia a particiones de la red

Operaciones Matriciales Distribuidas

- Distribuci n de matrices a nodos
- Matriz por Vector
- (Matriz por Matriz)
- (Inversi n Matricial)

Descenso por el Gradiente

- Descenso por el Gradiente
 - Parallelized stochastic gradient descent por M Zinkevich, M Weimer, L Li, AJ Smola en NIPS 2010
- Búsqueda distribuida
 - (Algoritmos genéticos)

Otros Modelos

- Sistemas por Colas
- Memoria Compartida

Paralelizando Algoritmos

- Algoritmos Actualizables
 - Naive Bayes
- Random Forests
- (Regresi n Log stica)

Objetivo

- Simplificar el acceso al cómputo de gran volumen de datos para programadores sin experiencia en cómputo distribuido
- Simplificar la tolerancia a fallas
- Simplificar la alocaación de recursos (máquinas, disco y red)

Modelo de Programaci  n

- Computa una funci  n $f(\{(k_{in}, v_{in})\}) \rightarrow \{(k_{out}, list(v_{out}))\}$
 - $map(k_{in}, v_{in}) \rightarrow list(k_{out}, v_{int})$
 - Para cada par (clave, valor) de entrada, produce una lista de pares de otros claves y valores intermedios
 - $reduce(k_{out}, list(v_{int})) \rightarrow list(v_{out})$
 - Acumular los valores intermedios seg  n su clave de salida
 - Generar la salida combinada

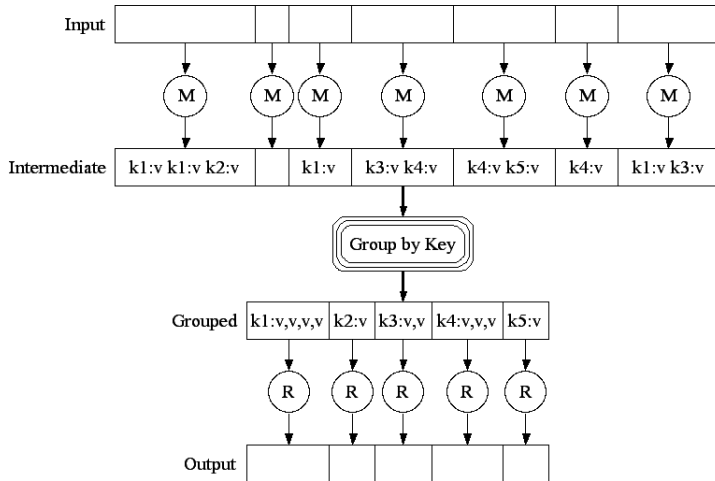
Ejemplo: Contar Palabras

- `map(input_key: DocName, input_value: DocText):`
 - for each word `w` in `input_value`:
 - `EmitIntermediate(w, 1)`
- `reduce(output_key: Word, intermediate_values: List[Int]):`
 - `int result = 0`
 - for each `v` in `intermediate_values`:
 - `result += v`
 - `Emit(result)`

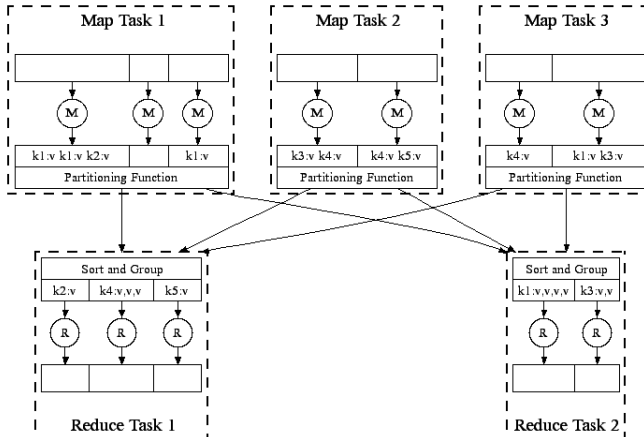
Ejemplos de Uso en Google

- distributed grep
- distributed sort
- web link-graph reversal
- term-vector per host
- web access log stats
- inverted index construction
- document clustering
- machine learning
- statistical machine translation

Ejecución



En Paralelo



(fuente: <http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0008.html>)

Pipelining

- Las máquinas pueden ser re-usadas para realizar tareas Map y Reduce
- A medida que las tareas Map terminan, las máquinas pueden ser utilizadas para hacer tareas Reduce
- Load Balancing dinámico
- Ejemplo de <http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0010.html>

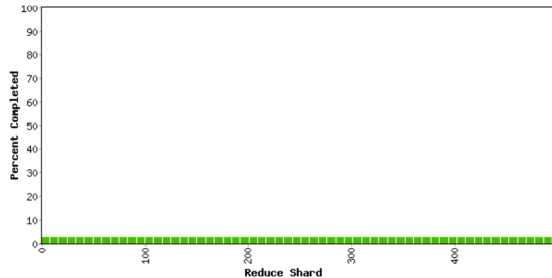
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 00 min 18 sec

323 workers; 0 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	0	323	878934.6	1314.4	717.0
Shuffle	500	0	323	717.0	0.0	0.0
Reduce	500	0	0	0.0	0.0	0.0



Counters

Variable	Minute
Mapped (MB/s)	72.5
Shuffle (MB/s)	0.0
Output (MB/s)	0.0
doc-index-hits	145825686
docs-indexed	506631
dups-in-index-merge	0
mr-operator-calls	508192
mr-operator-calls	506631

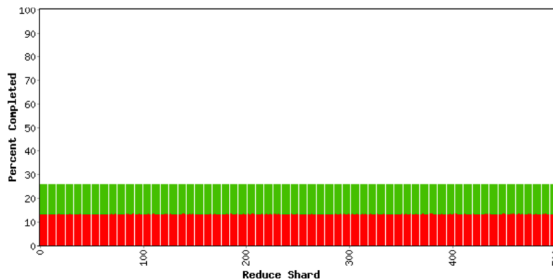
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 05 min 07 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	1857	1707	878934.6	191995.8	113936.6
Shuffle	500	0	500	113936.6	57113.7	57113.7
Reduce	500	0	0	57113.7	0.0	0.0



Counters

Variable	Minute
Mapped (MB/s)	699.1
Shuffle (MB/s)	349.5
Output (MB/s)	0.0
doc-index-hits	5004411944
docs-indexed	17290135
dups-in-index-merge	0
mr-operator-calls	17331371
mr-operator-outouts	17290135

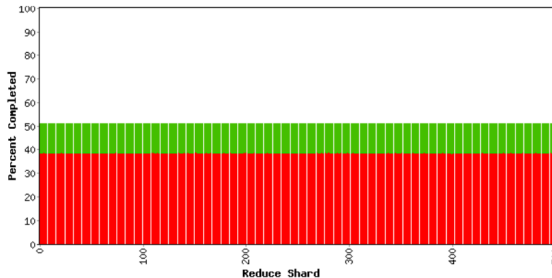
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 10 min 18 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	5354	1707	878934.6	406020.1	241058.2
Shuffle	500	0	500	241058.2	196362.5	196362.5
Reduce	500	0	0	196362.5	0.0	0.0



Counters

Variable	Minute
Mapped (MB/s)	704.4
Shuffle (MB/s)	371.9
Output (MB/s)	0.0
doc-index-hits	5000364228
docs-indexed	17300709
dups-in-index-merge	0
mr-operator-calls	17342493
mr-operator-outputs	17300709

Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 15 min 31 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	8841	1707	878934.6	621608.5	369459.8
Shuffle	500	0	500	369459.8	326986.8	326986.8
Reduce	500	0	0	326986.8	0.0	0.0



Counters

Variable	Minute
Mapped (MB/s)	706.5
Shuffle (MB/s)	419.2
Output (MB/s)	0.0
doc-index-hits	4982870667
docs-indexed	17229926
dups-in-index-merge	0
mr-operator-calls	17272056
mr-operator-outouts	17229926

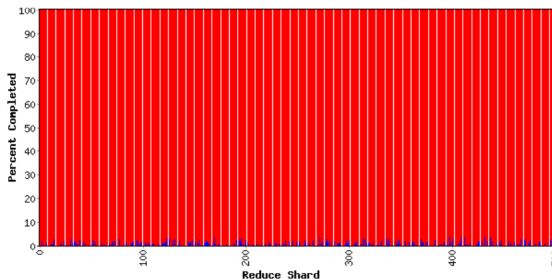
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 29 min 45 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	195	305	523499.2	523389.6	523389.6
Reduce	500	0	195	523389.6	2685.2	2742.6



Counters

Variable	Minute
Mapped (MB/s)	0.3
Shuffle (MB/s)	0.5
Output (MB/s)	45.7
doc-index-hits	2313178
docs-indexed	7936
dups-in-index-merge	0
mr-merge-calls	1954105
mr-merge-outputs	1954105

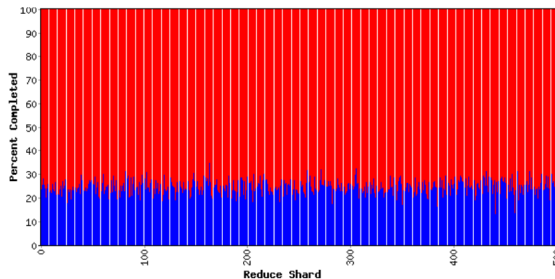
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 31 min 34 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	133837.8	136929.6



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.1
Output (MB/s)	1238.8
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51738599
mr-merge-outputs	51738599

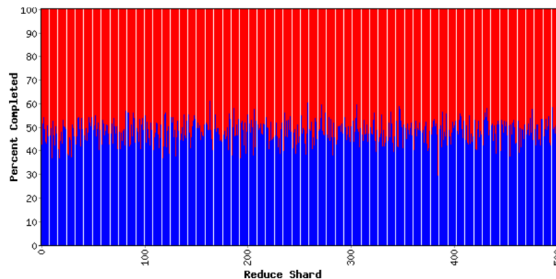
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 33 min 22 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	263283.3	269351.2



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	1225.1
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51842100
mr-merge-outputs	51842100

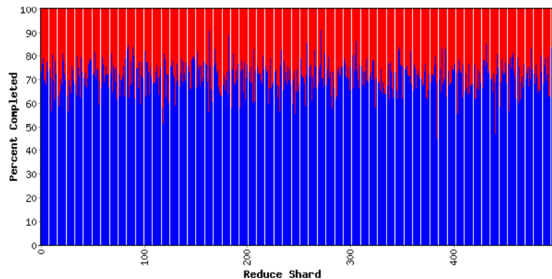
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 35 min 08 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	390447.6	399457.2



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	1222.0
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51640600
mr-merge-outs	51640600

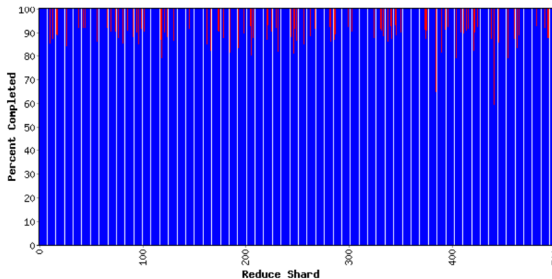
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 37 min 01 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	520468.6	520468.6
Reduce	500	406	94	520468.6	512265.2	514373.3



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	849.5
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mrr-merge-calls	35083350
mrr-merge-outs	35083350

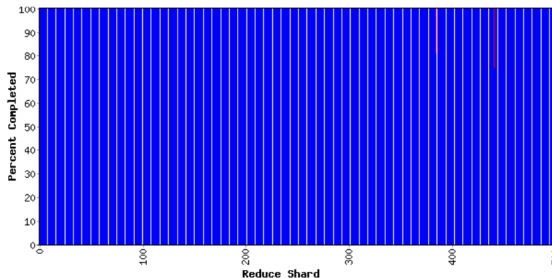
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 38 min 56 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	519781.8	519781.8
Reduce	500	498	2	519781.8	519394.7	519440.7



Counters

Variable	Minute	
Mapped (MB/s)	0.0	
Shuffle (MB/s)	0.0	
Output (MB/s)	9.4	
doc-index-hits	0	1056
docs-indexed	0	3
dups-in-index-merge	0	
mr-merge-calls	394792	3
mr-merge-outputs	394792	3

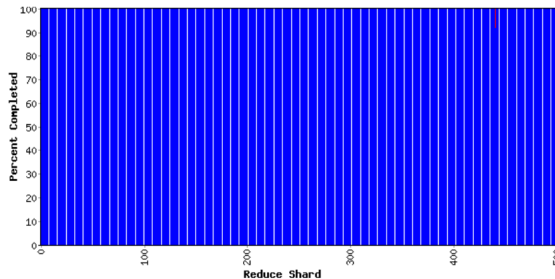
Pipelining

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 40 min 43 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	519774.3	519774.3
Reduce	500	499	1	519774.3	519735.2	519764.0



Counters

Variable	Minute	
Mapped (MB/s)	0.0	
Shuffle (MB/s)	0.0	
Output (MB/s)	1.9	
doc-index-hits	0	105
docs-indexed	0	:
dups-in-index-merge	0	
mr-merge-calls	73442	:
mr-merge-outs	73442	:

Tolerancia a Fallas

- Si el master muere, el sistema cae
- Si un worker muere el sistema se da cuenta via *heartbeats*
 - Re-ejecución de tareas fallidas
 - Re-ejecución de tareas en ejecución
 - Mejora el peor caso (ejecución redundante)
- Semántica en caso de fallas:
 - No presenta problemas para tareas determinísticas

GFS y Localidad de los Datos

- El GFS es un sistema de archivos distribuido
 - Diseñado para trabajar bien con el modelo de cluster que hablamos antes
- Los archivos son divididos en chunks de tamaño fijo y replicados en los nodos
 - Usualmente un nivel de réplica de 3 copias por chunk
- Una buena implementación de MapReduce asigna tareas Map a nodos donde los chunks son locales

Ejemplos

- distributed grep
- distributed sort
- web link-graph reversal
- term-vector per host
- web access log stats
- inverted index construction
- document clustering
- machine learning
- statistical machine translation