hamagraphloader

+GraphLoaderTextReader
+GraphLoaderVertex
+HamaGraphLoader

HamaGraphLoader

+main(args: String[])
-printUsage()
-createJob(args: String[],conf Configuration): GraphJob

uses

uses   *uses*

uses

GraphLoaderTextReader

lastVertexId: String=null
adjacents: ArrayList

+parseVertex(key: LongWritable,value: Text,vertex: Vertex): boolean

GraphLoaderVertex

+compute(messages: Iterator)

---
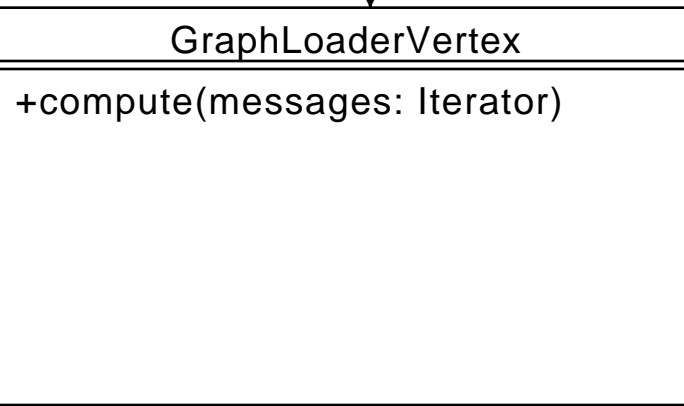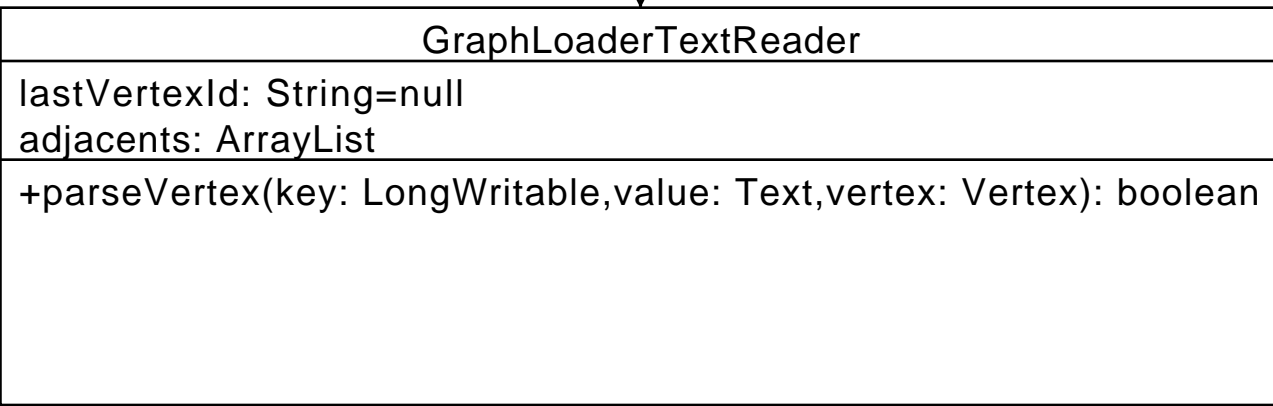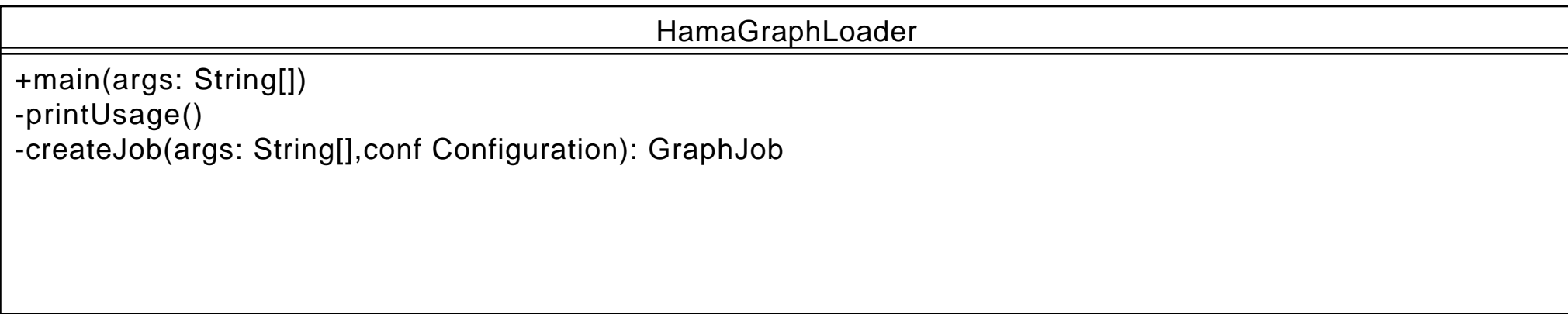
main.java.org.apache.hama.graph

«extends»

VertexInputReader

+parseVertex(key: KEYIN,value: VALUEIN,vertex: Vertex)

«extends»

Edge

-destinationVertexID: VERTEX_ID
-cost: EDGE_VALUE_TYPE
+getDestinationVertexID(): VERTEX_ID
+getValue(): EDGE_VALUE_TYPE
+toString()

BSP

BSPJobContext

extends

BSPJob

extends

GraphJob

+some static Strings
+setVertexClass(cls: Class)
+getVertexClass(): Class
+setVertexIDClass(cls: Class)
+setVertexClassValue(cls: Class)
+setEdgeValueClass(cls: Class)
+setAggregatorClass(cls: Class)
+setVertexInputReader(cls: Class)
+setPartitioner(cls: Class)
+setConbinerClass(cls: Class)
+setMaxIteration(maxIteration: int)
+submit()

extends

«Interface»
VertexInterface

+setup(conf: Configuration)
+getVertexID(): V
+getNumVertices(): long
+compute(messages: Iterator<M>)
+getEdges(): List<Edge>
+sendMessage(e: Edge,msg: M)
+sendMessage(destinationVertexID: V,msg:M)
+sendMessagesToNeighbors(msg: M)
+getSuperstepCount(): long
+voteToHalt()
+setValue(value: M)
+getValue(): M

implements

uses

Vertex

runner: GraphJobRunner
-vertexID: V
-value: M
-edges: List<Edge>
-voteToHalt: boolean
+setup(conf: Configuration)
+getVertexID(): V
+getNumVertices(): long
+compute(messages: Iterator<M>)
+getEdges(): List<Edge>
+sendMessage(e: Edge,msg: M)
+sendMessage(destinationVertexID: V,msg:M)
+sendMessagesToNeighbors(msg: M)
+getSuperstepCount(): long
+voteToHalt()
+setValue(value: M)
+getValue(): M
+getConf(): Configuration
+getDestinationPeerName(edge: Edge)
+getDestinationPeerName(vertexID: V)
+setEdges(list: List<Edge>)
+addEdge(edge: Edge)
+setVertexID(vertexID: V)
+getMaxIterations(): int
+getLastAggregatedValue(index: int): IntWritable
+getNumLastAggregatedVertices(index: int): IntWritable
+getNumPeers(): int
+getPeer(): BSPPeer
+getPartitioner(): Partitioner
+setActive()
+isHalted(): boolean
+hashCode(): int
+equals(obj Object): boolean
+toString(): String

contains an object "runner"

GraphJobRunner

+some static Strings
-conf: Configuration
-combiner: Combiner
-partitioner: Partitioner
-vertices: List<Vertex>
-updated: boolean
-globalUpdateCounts: int
-numberVertices: long
-maxIteration: int
-iteration: long
-vertexIDClass: Class
-vertexValueClas: Class
-edgeValueClass: Class
-vertexClass: Class
-aggregatorRunner: AggregatorRunner
-peer: BSPPeer
+setup(peer: BSPPeer)
+bsp(peer: BSPPeer)
+cleanup(peer: BSPPeer)
-doMasterUpdates(peer: BSPPeer)
-doSuperstep(peer: BSPPeer)
-doInitialSuperstep(peer: BSPPeer)
-setupFields(peer: BSPPeer)
-loadVertices(peer: BSPPeer)
-repair(peer: BSPPeer)
-partitionMultiSteps(peer: BSPPeer, splitSize: long): int
-countGlobalVertexCount(peer: BSPPeer)
-parseMessages(peer: BSPPeer): Map
+getNumberVertices(): long
+getNumberIterations(): long
+getMaxIteration(): int
+getPartitioner(): Partitioner
+getLastAggregatedValue(index: int): Writable
+getNumLastAggregatedVertices(index: int): IntWritable
+getPeer(): BSPPeer
+isMasterTask(peer: BSPPeer): boolean
+getMasterTask(peer: BSPPeer): String
+newVertexInstance(vertexClass: Class,conf: Configuration): vertex

uses

uses

Writable

implements

GraphJobMessage

+some static int flags
+VERTEX_CLASS: Class
+VERTEX_ID_CLASS;
+VERTEX_VALUE_CLASS;
+EDGE_VALUE_CLASS;
-flag: int
-map: MapWritable
-vertexId: Writable
-vertexValue: Writable
-vertex: Vertex
-vertices_size: IntWritable
+write(out: DataOutput)
+readFields(in: DataInput)
+getMap(): MapWritable
+getVertexId(): Writable
+getVertexValue(): Writable
+getVertex(): Vertex
+getVerticesSize(): IntWritable
+isMapMessage(): boolean
+isVertexMessage(): boolean
+isRepairMessage(): boolean
+isPartitioningMessage(): boolean
+isVerticesSizeMessage(): boolean
+toString(): String

uses

«Interface»
Aggregator

+aggregate(vertex: Vertex,value: M)
+getValue(): M

setup uses

uses

implements

AggregatorRunner

-aggregators: Aggregator[]
-globalAggregatorResult: Writable[]
-globalAggregatorIncrement: IntWritable[]
-isAbstractAggregator: boolean[]
-aggregatorClassNames: String[]
-aggregatorValueFlag: Text[]
-aggregatorIncrementFlag: Text[]
-masterAggregator: Aggregator[]
-enabled:boolean
-conf: Configuration
+setupAggregators(peer: BSPPeer)
+sendAggregatorValues(peer: BSPPeer,activeVertices: int)
+aggregateVertex(lastValue: M,v: Vertex)
+doMasterAggregation(updateCnt: MapWritable)
+receiveAggregatedValues(peer: BSPPeer,long iteration): booleab
+isEnabled(): boolean
+masterReadAggregatedValue(textIndex: Text,value: M)
+masterReadAggregatedIncrementalValue(textIndex: Text,value: M)
-getNewAggregator(clsName: String): Aggregator
+getLastAggregatedValue(index: int): Writable
+getNumLastAggregatedVertices(index: int): Writable

uses

AbstractAggregator

-timesAggregated: int
+aggregate(vertex: Vertex,value: M)
+aggregate(vertex: Vertex,oldValue: M,newValue: M)
+finalizeAggregator(): M
+getValue(): M
+getTimesAggregated(): IntWritable
+toString(): String

«extends»   «extends»   «extends»   «extends»

AbsDiffAggregator

absoluteDifference: double
+aggregate(vertex: Vertex,
value: DoubleWritable)
+aggregate(vertex: Vertex,
oldValue: DoubleWritable,
newValue: DoubleWritable)
+getValue(): DoubleWritable

SumAggregator

sum: double
+aggregate(vertex: Vertex,
value: DoubleWritable)
+getValue(): DoubleWritable

MinAggregator

min: int
+aggregate(vertex: Vertex,
value: IntWritable)
+getValue(): IntWritable

MaxAggregator

max: int
+aggregate(vertex: Vertex,
value: IntWritable)
+getValue(): IntWritable

«extends»

AverageAggregator

+finalizeAggregation(): DoubleWritable