BroadcastManager

- isDriver: Boolean
- conf: SparkConf
- securityManager: SecurityManager
+ isDriver:Boolean
+ conf:SparkConf
+ securityManager:SecurityManager
+ initialized:Boolean
+ broadcastFactory:BroadcastFactory
+ nextBroadcastId:AtomicLong
+ cachedValues:SynchronizedMap

+ def initialize(): Unit
+ def stop(): Unit
+ def newBroadcast[T: ClassTag](value_ : T, isLocal: Boolean): Broadcast[T]
+ def unbroadcast(id: Long, removeFromDriver: Boolean, blocking: Boolean): Unit

---

Broadcast

- id: Long
+ _isValid:Boolean
+ _destroySite:String

+ def value: T
+ def unpersist(): Unit
+ def unpersist(blocking: Boolean): Unit
+ def destroy(): Unit
+ def destroy(blocking: Boolean): Unit
+ def isValid: Boolean
+ def getValue(): T
+ def doUnpersist(blocking: Boolean): Unit
+ def doDestroy(blocking: Boolean): Unit
+ def assertValid(): Unit
+ def toString: String

BroadcastFactory

+ def initialize(isDriver: Boolean, conf: SparkConf, securityMgr: SecurityManager): Unit
+ def newBroadcast[T: ClassTag](value: T, isLocal: Boolean, id: Long): Broadcast[T]
+ def unbroadcast(id: Long, removeFromDriver: Boolean, blocking: Boolean): Unit
+ def stop(): Unit

TorrentBroadcastFactory

+ def initialize(isDriver: Boolean, conf: SparkConf,securityMgr: SecurityManager): Unit
+ def stop(): Unit
+ def newBroadcast[T: ClassTag](value_ : T, isLocal: Boolean, id: Long): Broadcast[T]
+ def unbroadcast(id: Long, removeFromDriver: Boolean, blocking: Boolean): Unit

---

TorrentBroadcast

+ _value:SoftReference
+ obj: T
+ id:Long
+ compressionCodec:Option[CompressionCodec]
+ blockSize:int
+ broadcastId:BroadcastBlockId(id)
+ numBlocks:int
+ checksumEnabled:Boolean
+ checksums:Array[int]

+ def setConf(conf: SparkConf): Unit
+ def getValue()
+ def calcChecksum(block: ByteBuffer): Int
+ def writeBlocks(value: T): Int
+ def readBlocks(): Array[BlockData]
+ def doUnpersist(blocking: Boolean): Unit
+ def doDestroy(blocking: Boolean): Unit
+ def writeObject(out: ObjectOutputStream): Unit
+ def readBroadcastBlock(): T
+ def releaseBlockManagerLock(blockId: BlockId): Unit

TorrentBroadcast(Object)

+ torrentBroadcastLock:KeyLock[BroadcastBlockId]

+ def blockifyObject[T: ClassTag](obj: T,blockSize: Int,serializer: Serializer, compressionCodec: Option[CompressionCodec]): Array[ByteBuffer]
+ def unBlockifyObject[T: ClassTag](blocks: Array[InputStream],serializer: Serializer,compressionCodec: Option[CompressionCodec]): T
+ def unpersist(id: Long, removeFromDriver: Boolean, blocking: Boolean): Unit

伴生