

资源请求

ExecutorResourceRequest
- resourceName:String - amount:Long - discoveryScript:String - vendor:String + allowedExecutorResources:HashSet

TaskResourceRequest
- resourceName: String - amount: Double

ExecutorResourceRequests
+ _executorResources:HashMap
+ def requests: Map[String, ExecutorResourceRequest] + def memory(amount: String): this.type + def memoryOverhead(amount: String): this.type + def pysparkMemory(amount: String): this.type + def cores(amount: Int): this.type + def resource(resourceName: String,amount: Long,discoveryScript: String = "", vendor: String = ""): this.type + def addRequest(ereq: ExecutorResourceRequest): this.type + toString:String

TaskResourceRequests
+ _taskResources:HashMap
+ def toString: String + def requests: Map[String, TaskResourceRequest] + def cpus(amount: Int): this.type + def resource(rName: String, amount: Double): this.type + def addRequest(treq: TaskResourceRequest): this.type

ResourceRequest
- id: ResourceID - amount: Int - discoveryScript: Option[String] - vendor: Option[String]

ResourceRequirement
- resourceName: String - amount: Int - numParts: Int

资源分配

ResourceAllocation
- id: ResourceID - addresses: Seq[String] + def toResourceInformation: ResourceInformation

ResourceAllocator
+ addressAvailabilityMap
+ def resourceName: String + def resourceAddresses: Seq[String] + def slotsPerAddress: Int + def availableAdrrs: Seq[String] + def assignedAdrrs: Seq[String] + def acquire(addr: Seq[String]): Unit + def release(addr: Seq[String]): Unit

资源

ResourceID
- componentName:String - resourceName: String - def confPrefix: String + def amountConf: String + def discoveryScriptConf: String + def vendorConf: String

ResourceInformation
- name: String - addresses: Array[String] + def toString: String + def toJson(): JValue + def equals(obj: Any): Boolean + def hashCode(): Int

ResourceInformation(Object)
+ exampleJson
+ def parseJson(json: String): ResourceInformation + def parseJson(json: JValue): ResourceInformation

伴生

ResourceInformationJson
- name: String - addresses: Seq[String] + def toJsonValue: JValue + def toResourceInformation: ResourceInformation

ResourceProfile
+ _id=ResourceProfile.getNextProfileId + _taskResources:HashMap + _executorResources:HashMap + def id: Int = _id + def taskResources: Map[String, TaskResourceRequest] + def executorResources: Map[String, ExecutorResourceRequest] + def taskResourcesJMap: JMap[String, TaskResourceRequest] + def executorResourcesJMap: JMap[String, ExecutorResourceRequest] + def reset(): Unit + def require(requests: ExecutorResourceRequests): this.type + def require(requests: TaskResourceRequests): this.type + def toString(): String

ResourceProfile(Object)
+ UNKNOWN_RESOURCE_PROFILE_ID=-1 + DEFAULT_RESOURCE_PROFILE_ID + CPUS="cpus" + CORES="cores" + MEMORY="memory" + OVERHEAD_MEM="memoryOverhead" + PYSARK_MEM="pyspark.memory" + nextProfileId + defaultProfileRefnew
+ def getNextProfileId: Int + def resetDefaultProfile(conf: SparkConf): Unit + def getOrCreateDefaultProfile(conf: SparkConf): ResourceProfile + def addDefaultTaskResources(rprof: ResourceProfile, conf: SparkConf): Unit + def addDefaultExecutorResources(rprof: ResourceProfile, conf: SparkConf): Unit

ResourceUtils
+ DISCOVERY_SCRIPT + VENDOR + AMOUNT + GPU + FPGA + RESOURCE_PREFIX + RESOURCE_DOT + def parseResourceRequest(sparkConf: SparkConf, resourceId: ResourceID): ResourceRequest + def listResourceIds(sparkConf: SparkConf, componentName: String): Seq[ResourceID] + def parseAllResourceRequests(sparkConf: SparkConf,componentName: String): Seq[ResourceRequest] + def resourcesMeetRequirements(resourcesFree: Map[String, Int],resourceRequirements: Seq[ResourceRequirement]): Boolean + def withResourcesJson[T](resourcesFile: String)(extract: String => Seq[T]): Seq[T] + def parseAllocatedFromJsonFile(resourcesFile: String): Seq[ResourceAllocation] + def parseAllocatedOrDiscoverResources(sparkConf: SparkConf,componentName: String, resourcesFileOpt: Option[String]): Seq[ResourceAllocation] + def parseResourceRequirements(sparkConf: SparkConf, componentName:String:Seq[ResourceRequirement] + def assertResourceAllocationMeetsRequest(allocation: ResourceAllocation, request: ResourceRequest): Unit + def assertAllResourceAllocationsMeetRequests(allocations: Seq[ResourceAllocation], requests: Seq[ResourceRequest]): Unit + def logResourceInfo(componentName: String, resources: Map[String, ResourceInformation]) + def getOrDiscoverAllResources(sparkConf: SparkConf,componentName: String, resourcesFileOpt: Option[String]): Map[String, ResourceInformation] + def discoverResource(resourceRequest: ResourceRequest): ResourceInformation