



عنوان مقاله: تنظیمات پیکربندی حافظه YARN و MapReduce

نویسنده مقاله: امیرعلی میرزاییگی

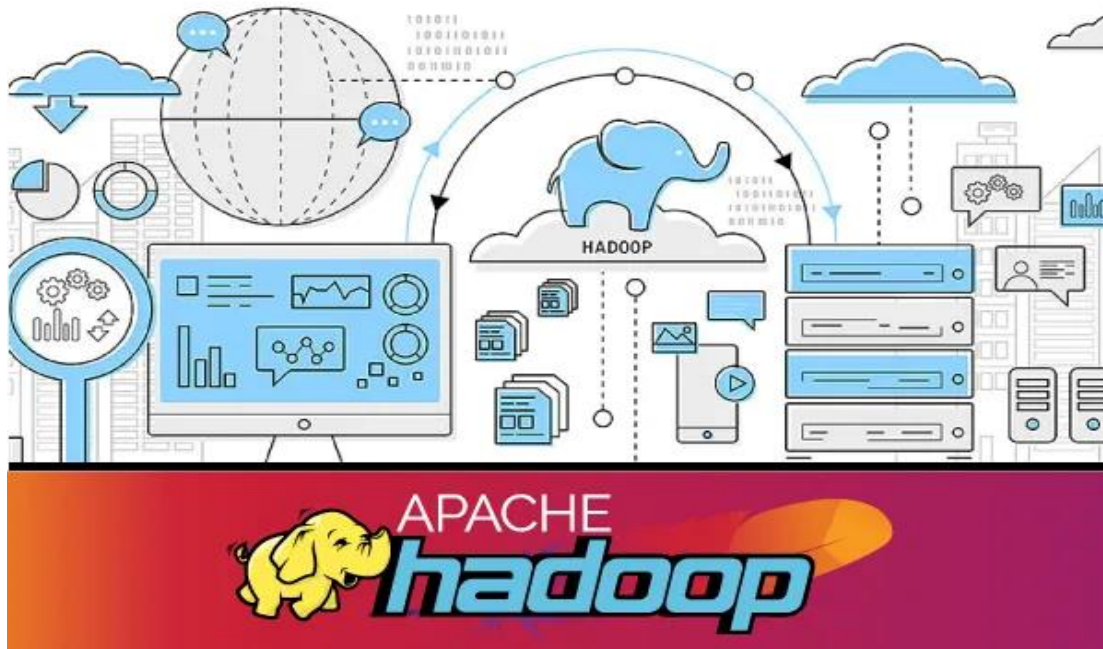
تاریخ انتشار: ۲۱ آذر ۱۴۰۲

منبع: <https://nikamooz.com/yarn-and-mapreduce-memory-configuration-settings/>

تنظیمات پیکربندی حافظه YARN و MapReduce در کلاستر Hadoop اهمیت فراوانی دارد و لازم است فرآیند کانفیگ آن، به درستی انجام شود. در این مقاله، ابتدا Apache Hadoop و اجرای اصلی آن را معرفی و سپس به بررسی نحوه تنظیم پیکربندی حافظه YARN و MapReduce پرداخته خواهد شد.

### چيست Apache Hadoop

آپاچی هودوپ (Apache Hadoop) یک فریمورک متن باز است که برای فضای ذخیره‌سازی توزیع‌شده و پردازش مجموعه داده‌های وسیع طراحی شده است و بخشی از اکوسیستم Apache به حساب می‌آید. Apache Hadoop یک پلتفرم مقیاس‌پذیر، قابل اکتفا و متحمل خطا (Fault-Tolerant) برای کلان داده‌ها محسوب می‌شود.



## اجزای اصلی Apache Hadoop

مهم‌ترین کامپوننت‌های Apache Hadoop به شرح زیر است:

### (HDFS) Hadoop Distributed File System

HDFS یکی از پراهمیت‌ترین اجزای اکوسیستم Hadoop محسوب می‌شود و وظیفه آن، ذخیره‌سازی داده‌های وسیع ساختاریافته و بدون ساختار در گره‌های (Nodes) مختلف است. MetaData حاصل، در قالب فایل‌های Log نگهداری می‌شوند. در عمل، HDFS فایل‌های بزرگ را به بلوک‌های کوچک‌تر تفکیک کرده و آن‌ها را به گره‌های موجود در یک کلاستر Hadoop توزیع می‌کند. چنین مشخصه‌ای، پردازش موازی و تحمل خطا را امکان‌پذیر خواهد کرد.

### MapReduce

MapReduce یک مدل برنامه‌نویسی و موتور پردازش است که برای محاسبات توزیع‌شده روی مجموعه داده‌های گسترده مورد استفاده قرار می‌گیرد. MapReduce به توسعه‌دهندگان امکان نوشتن برنامه‌هایی را می‌دهد که حجم وسیعی از داده‌ها را به صورت موازی (Parallel) روی کلاستر Hadoop پردازش کند. این پردازش به دو فاز، Map (برای پردازش) و Reduce (برای خلاصه‌سازی) تقسیم می‌شود.

### YARN (Yet Another Resource Negotiator)

YARN یک لایه مدیریت منابع برای Hadoop به حساب می‌آید و به کمک آن، چندین اپلیکیشن می‌توانند منابع را روی یک خوشه کلاستر به اشتراک بگذارند. YARN، کار نظارت (Monitoring) و زمان‌بندی (Scheduling) توابع MapReduce و مدیریت منابع را از هم جدا می‌کند و به سایر اپلیکیشن‌های محاسبات توزیع‌شده، امکان اجرا در کنار MapReduce را می‌دهد.

### Hadoop Common

Hadoop Common مواردی همچون ابزارها (Utilities)، کتابخانه‌ها (Libraries) و API هایی را دربرمی‌گیرد که سایر ماژول‌های Hadoop را پشتیبانی می‌کند. این کامپوننت، ابزارها و API هایی را شامل می‌شود که برای تسک‌های رایج، از جمله محاسبات توزیع‌شده، امنیت و مدیریت داده‌ها، به کار می‌روند.

با این دید مقدماتی، در ادامه تنظیمات پیکربندی حافظه YARN و MapReduce و مفاهیم مربوطه پرداخته خواهد شد.

## پیکربندی حافظه YARN و MapReduce چگونه است؟

YARN تمام منابع موجود در هر ماشین را به صورت Cluster در نظر می‌گیرد، سپس YARN با تخصیص Container ها، ظرفیت پردازش را برای هر برنامه فراهم می‌کند. Container ها، واحد اصلی ظرفیت پردازش در YARN و یک Encapsulation از عناصر (CPU، Ram، ...) هستند.

در یک کلاستر Hadoop، لازم است یک حد تعادل بین استفاده از حافظه (RAM)، هسته‌های پردازنده (CPU cores) و دیسک‌ها وجود داشته باشد تا پردازش توسط هیچ یک از این منابع کلاستر محدود نشود.

به عنوان یک توصیه کلی، اجازه دادن به دو Container در CPU و Disk، بهترین راه تعادل برای استفاده از کلاستر را فراهم می‌کند.

هنگام تعیین پیکربندی حافظه YARN و MapReduce برای یک Cluster Node، از منابع سخت‌افزاری موجود شروع کنید. به طور خاص، به مقادیر زیر در هر node توجه کنید:

- مقدار RAM
- CPU (CPU Cores)
- تعداد Disk

لازم است تمام RAM موجود برای YARN و MapReduce به صورت رزرو شده در نظر گرفته شود. Reserved Memory RAM مورد نیاز فرآیندهای سیستم و سایر فرآیندهای Hadoop مانند HBase است.

Reserved Memory = Reserved for stack memory + Reserved for HBase memory (If HBase is on the same node)

از جدول زیر برای تعیین Reserved Memory در هر Node استفاده کنید:

توصیه‌های Reserved Memory

Total Memory per Node	Recommended Reserved System Memory	Recommended Reserved HBase Memory
4 GB	1 GB	1 GB
8 GB	2 GB	1 GB
16 GB	2 GB	2 GB
24 GB	4 GB	4 GB
48 GB	6 GB	8 GB
64 GB	8 GB	8 GB
72 GB	8 GB	8 GB
96 GB	12 GB	16 GB
128 GB	24 GB	24 GB
256 GB	32 GB	32 GB
512 GB	64 GB	64 GB

تعیین تعداد حداکثر Container های مجاز در هر node، براساس فرمول زیر است:

$$\text{Containers} = \text{minimum of } (2 * \text{CORES}, 1.8 * \text{DISKS}, (\text{Total available RAM}) / \text{MIN\_CONTAINER\_SIZE})$$

جایی که MIN\_CONTAINER\_SIZE حداقل اندازه کانتینر (در RAM) است. این مقدار، به مقدار RAM available بستگی دارد.

در Node های حافظه کوچکتر، حداقل اندازه Container نیز باید کوچکتر باشد.

جدول زیر، مقادیر توصیه شده را نشان می دهد:

مقدار RAM برای هر Node	حداقل اندازه Container
کمتر از GIB4	256 MB
بین GIB4 تا GIB8	512 MB
بین GIB8 تا GIB24	1024 MB
بیشتر از GIB24	2048 MB

محاسبه نهایی برای تعیین مقدار RAM در هر Container:

$$\text{RAM-per-Container} = \text{maximum of } (\text{MIN\_CONTAINER\_SIZE}, (\text{Total Available RAM}) / \text{Containers})$$

با این محاسبات، می توان YARN و MapReduce را تنظیم کرد:

تنظیمات مربوط به تخصیص منابع سخت حافظه (RAM) برای YARN و MapReduce براساس مشخصات سخت افزاری، به صورت زیر است:

Configuration File	Configuration Setting	Value Calculation
yarn-site.xml	yarn.nodemanager.resource.memory-mb	= Containers * RAM-per-Container
yarn-site.xml	yarn.scheduler.minimum-allocation-mb	= RAM-per-Container
yarn-site.xml	yarn.scheduler.maximum-allocation-mb	= Containers * RAM-per-Container
mapred-site.xml	mapreduce.map.memory.mb	= RAM-per-Container
mapred-site.xml	mapreduce.reduce.memory.mb	= 2 * RAM-per-Container
mapred-site.xml	mapreduce.map.java.opts	= 0.8 * RAM-per-Container
mapred-site.xml	mapreduce.reduce.java.opts	= 0.8 * 2 * RAM-per-Container
yarn-site.xml (check)	yarn.app.mapreduce.am.resource.mb	= 2 * RAM-per-Container
yarn-site.xml (check)	yarn.app.mapreduce.am.command-opts	= 0.8 * 2 * RAM-per-Container

توجه شود که پس از نصب، هر دو yarn-site.xml و mapred-site.xml در پوشه etc/hadoop/conf/ قرار دارند.

### مثال

کلاستر داریم شامل:

- **CPU:** ۱۲ cores
- **RAM:** ۴۸ GIB
- **Disk:** ۱۲
- **Reserved Memory** = ۶ GB reserved for system memory + (if HBase) 8 GB for HBase
- **Min Container size** = ۲ GB

اگر HBase نداریم:

# Containers = minimum of (2\*12, 1.8\* 12, (48-6)/2) = minimum of (24, 21.6, 21) = 21

RAM-per-Container = maximum of (2, (48-6)/21) = maximum of (2, 2) = 2

Configuration	Value Calculation
yarn.nodemanager.resource.memory-mb	= 21 * 2 = 42*1024 MB
yarn.scheduler.minimum-allocation-mb	= 2*1024 MB
yarn.scheduler.maximum-allocation-mb	= 21 * 2 = 42*1024 MB
mapreduce.map.memory.mb	= 2*1024 MB
mapreduce.reduce.memory.mb	= 2 * 2 = 4*1024 MB
mapreduce.map.java.opts	= 0.8 * 2 = 1.6*1024 MB
mapreduce.reduce.java.opts	= 0.8 * 2 * 2 = 3.2*1024 MB
yarn.app.mapreduce.am.resource.mb	= 2 * 2 = 4*1024 MB
yarn.app.mapreduce.am.command-opts	= 0.8 * 2 * 2 = 3.2*1024 MB



## اگر Hbase وجود داشته باشد:

Containers = minimum of ( $2 * 12$ ,  $1.8 * 12$ ,  $(48 - 6 - 8) / 2$ ) = minimum of (24, 21.6, 17) = 17

RAM-per-Container = maximum of (2,  $(48 - 6 - 8) / 17$ ) = maximum of (2, 2) = 2

Configuration	Value Calculation
yarn.nodemanager.resource.memory-mb	= 17 * 2 = 34*1024 MB
yarn.scheduler.minimum-allocation-mb	= 2*1024 MB
yarn.scheduler.maximum-allocation-mb	= 17 * 2 = 34*1024 MB
mapreduce.map.memory.mb	= 2*1024 MB
mapreduce.reduce.memory.mb	= 2 * 2 = 4*1024 MB
mapreduce.map.java.opts	= 0.8 * 2 = 1.6*1024 MB
mapreduce.reduce.java.opts	= 0.8 * 2 * 2 = 3.2*1024 MB
yarn.app.mapreduce.am.resource.mb	= 2 * 2 = 4*1024 MB
yarn.app.mapreduce.am.command-opts	= 0.8 * 2 * 2 = 3.2*1024 MB

## مروری بر تنظیمات پیکربندی حافظه YARN و MapReduce

MapReduce و YARN دو کامپوننت پراهمیت از Apache Hadoop تلقی می‌شوند. در این مطلب، تنظیمات پیکربندی حافظه YARN و MapReduce شرح داده شد. شما می‌توانید با به‌کارگیری YARN به‌عنوان فریمورک مدیریت منابع و MapReduce به منظور پردازش مجموعه داده‌های گسترده به‌صورت موازی (در Node های مختلف)، از این اجزا به بهترین شکل بهره‌مند شوید.