SkyForm AIP - 应用任务管理调度平台
功能白皮书

1. 概述

SkyForm 应用任务管理调度平台（简称 SkyForm AIP） 是由天云软件自发研制的高性能、高可靠、高可扩的人工智能、高性能计算、大数据应用管理平台，具有自主可控知识产权，核心技术不依赖于国外开源社区，产品成熟，已经广泛用于规模生产环境。

SkyForm AIP 是为人工智能框架、高性能计算、大数据等应用专门设计的企业级资源和任务调度和用户访问系统，注重大规模集群高性能计算、分布式深度学习、机器学习、数据分析等任务管理，使用户在使用大集群和异构硬件的时候达到像用本地系统一样的简单和透明，同时又让系统管理员能够有效地监控和管理集群上所有的资源，使昂贵资源的利用率最大化，从而提高效能、降低成本。

SkyForm AIP 与市场上常见的应用任务管理云平台比较对应用环境的支持有以下明显的优势和先进性：

（1） 应用和服务直接在物理机上运行，不受容器对特殊硬件支持的限制，大大降低系统复杂度和系统软件开销。监控各个任务的状态和资源使用情况，集中统计在任务信息中显示，以便用户及时掌握学习任务的状况。其它平台要不以容器为主，不支持特殊 AI 加速硬件，要不只支持物理机，对容器的支持不够。

（2） 采用 OS 的用户系统来控制用户认证和用户权限，避免系统管理多套用户认证和权限系统，和不必要的数据库开销。

（3） 利用共享文件系统实现数据的持久性和内建的检测机制实现系统服务（如调度器）的 HA，避免依赖于第三方容错软件，降低排错难度、无需数据库管理和清理工作。

(4) 支持分布式多种任务异构资源的集中调度管理，其它的资源调度器对多种任务的资源每次调度一种，当一种任务所需资源不足时其它作业占着资源等待，造成资源浪费。SkyForm AIP 把所有应用的异构组件作为单一作业，直到所有任务所需资源都满足时才启动，以保证昂贵资源利用的最大化。多任务异构资源的统一调度是 SkyForm AIP 的独特调度能力，保障应用性能和资源利用最大化，其它平台没有这种能力。

例子：TensorFlow 的 master 和 Worker 需要 GPU，而 PS 不需要，K8S、MESOS、YARN 等资源管理器会先分配 master，再分配 ps，最后分配 worker。当 master 和 ps 分配后 worker 的资源不足时，会让 master 和 ps 等待，直到 worker 所需资源分配到为止，当集群忙是时，这种占着资源（master，ps）等待的情况会市场发生，造成集群利用率下降，集群越忙，这种情况发生的概率越大。SkyForm AIP 的调度是把 master, ps 和 worker 作为一个作业进行调度，所有任务所需的资源要不都给，要不都不给，不会造成一部分等另一部分的现象。

(5) 特殊调度策略：根据实际资源使用阈值的调度、大作业资源预留、小作业回填等，基于容器的资源管理软件（如 K8S，MESOS）不具备这样完整的大型生产环境需要的高级调度策略支持。

(6) 与主流并行平台 MPI（如 Intel MPI）深度集成。MPI 作业在 K8S 这样的容器平台上很难集成，若是用高速互联（如 InfiniBand）则 K8S 基本不支持。

(7) 调度速度可达每秒 5000 个作业，吞吐量优于所有其它资源管理器，处于国际最领先水平。

(8) 目前支持 500,000 核以上的大集群，达到业界领先水平。

2. SkyForm AIP 关键技术

2.1. 调度策略

2.4.1 对分布式多种任务异构资源的集中调度管理

其它的资源调度器对多种任务的资源每次调度一种，当一种任务所需资源不足时其它作业占着资源等待，造成资源浪费。SkyForm AIP 把整个学习框架作为单一作业，直到所有任务所需资源都满足时才启动，以保证昂贵资源利用的最大化。多任务异构资源的统一调度是 SkyForm AIP 的独特调度能力，保障应用性能和资源利用最大化，其它平台没有这种能力。

2.4.2 伸缩资源主动分配

应用（作业）在一开始可以告诉调度器所需最小和最大资源的值，调度器会根据调度策略和可用资源尽量满足应用的需求。如果不能满足最大资源需求，在应用运行的过程中若有冗余资源可用，调度器会主动把这些资源分配给作业直到作业所需最大资源得到满足。这种主动分配的调度有益于提高像深度学习一类资源饥渴型应用的性能。

这种调度算法也是其它资源管理软件缺乏的。

2.4.3 系统还支持全面的调度策略，主要有：

a) 先进先出：根据作业递交的先后时间顺序分发作业。

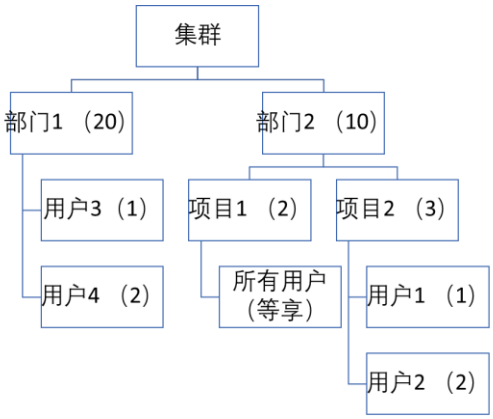
b) 优先级：作业根据优先级递交到不同优先级的队列中，调度系统先分发优先级高的队列中的作业，当高优先级队列中没有等待作业时，才分发下一个优先级队列中的作业。

c) 轮循：当同一队列中有多个用户的作业时，调度系统为每个用户分发一个作业，等队列中所有的用户的第一个作业分发后，再分发每个用户的第二个作业，每个用户的第三个作业，等等。

d) 独占：用户递交作业时可指定为独占作业。独占作业是指每个主机上只能运行这一指定的作业。若有一个独占作业分发到一台主机上，主机将不接受其它普通作业。若主机

上已有其他普通作业，则独占作业不会分发到该主机上。独占作业一般用于需占用大量资源的作业，以防与其他作业在同一主机上发生冲突。

e) 公平分享：当一个集群由多个部门和多个共享的时候，可定义各个部门、用户组、和个用户的使用份额，如下图所示部门、组和用户间的关系，数字表示相邻同级单位间的相对份额（数字大小没有实际意义，相邻单位间的数值比值决定其份额，如 10：20 和 1：2 有同样的效果。下图中部门 1 分配 2/3 的集群资源，部门 2 分配 1/3）。调度系统根据这样的配置来决定作业的优先级以保证各单位间的资源分配份额。



当某个单位所分的资源不被使用时，其他部门会根据配置的份额共享这部分资源，而后这个单位有作业等待时，其份额内被占的资源会在其他单位作业结束后被分回来。公平分享在一个长的时间段内保证各部门资源的配额，又防止配额不用时闲置浪费的现象。

公平分享可以有多层，如上图中部门下面是项目，项目中有用户。公平分享算法确保每一层内各层间的资源份额。

f) 抢占：高优先级作业通过抢占 CPU 核、GPU 以及其他资源使低优先级作业暂停（释放 CPU）或重调度（释放 GPU 等其他资源）的方式提前运行。高优先级作业运行结束后，低优先级作业继续或重运行

g) 并行作业资源自动预留：在繁忙的集群系统中，往往空出来的资源比较小，小作业就容易拿到资源而先走，这样即使大作业优先级高，也会因没有大块资源空出而长期等待。

调度系统可以配置使高优先级并行作业自动将空出的小块资源保留一段时间不被小作业所占，等保留的资源足够时运行。

h) 基于资源阀值的调度：由于作业所用资源难以实现预估，为防止资源不足，尤其是内存不足导致作业失败，可以定义资源的阈值来控制作业调度。对每一个资源可以定义两个阈值（上下水位），第一个下水位用于停止调度，第二个上水位用户停止（杀掉或挂起）已在运行的作业。资源阈值可设在主机层或/和队列层。

i) 资源平衡方式：资源平衡可以有两种方式：减少资源碎片（Packing）或负载平衡（Spreading）。减少资源碎片将作业尽量往最少的主机上调度，以便留下大块资源给大作业用。负载平衡是将作业尽量分布开，以保证作业运行性能和降低主机功耗。

j) 异构系统：允许将不同架构的主机、不同型号和性能的主机、不同操作系统和版本的主机放到一个集群里，通过“host type”参数进行配置。每种不同种类的主机可以定义一个 CPU 的性能值。在递交作业时可以指定这些参数配合使用。

k) 定时作业作业：定时作业与普通作业一样可由所有调度策略调度和作业定义（如环境变量、资源需求等）。

在作业定义中可指定运行用户名、运行时间点、作业命令行、作业最长运行时间（若超出此时间限制，作业会被自动杀掉）、启动超时（若由于在规定的时间里资源不足作业无法启动，最长等待的时间）、覆盖（下一个作业启动时上一个作业未完成是继续运行还是杀掉以前的作业）、失败重新运行最多次数等参数。

l) 优先级抢占：高优先级作业可以暂停低优先级作业获得作业资源。高优先作业运行结束后，低优先级作业可以自动恢复。

(2) 业务调度

用户的作业递交到队列中，系统可以设多个队列，每个队列一般可与业务挂钩，合理分配资源，避免作业冲突。队列可设置以下参数：

a) 指定用户：只有指定的用户或用户组可以使用此对列。

b) 总作业槽限制：对业务可用资源的限制。

c) 每个用户总作业槽限制：每个用户做多可用的作业槽数。

d) 主机或主机偏爱：可以限制队列可用的主机或主机组名，也可在分发作业时优先考虑一些主机。

e) 作业预处理/后处理：定制的作业前后处理程序，以便为作业运行做准备，检查作业运行必备条件，和在作业完成后做清理工作。

f) 作业操作定制（暂停、恢复、结束）：对作业的操作在队列层做统一的定制化。如暂停作业默认的操作是暂停作业所有进程，可以定制成暂停作业时叫作业杀掉，然后重新放入队列中。

h) 作业资源限制，可以根据主机、项目、用户的任意组合指定作业槽的限制。

i) CPU 绑定：将作业绑定到 CPU 的核以优化运行性能。

j) GPU 使用检测：若作业使用未经分配的 GPU，使用 GPU 的进程将会被强行终止。

k) 允许/禁止交互作业：交互作业可以让用户在终端上看到应用的屏幕输出，队列可以根据需要允许或禁止用户递交交互作业。

l) 作业自动运行：当作业运行的主机故障时，系统会自动在其它符合条件的主机上自动重启作业。

m) 作业运行时间窗：有些业务只需要在一定的时间窗里运行，如 regression 优先级低，一般可利用晚间和周末的时间运行，则可定义晚间和周末运行的时间窗。

2.2. 系统监控

SkyForm AIP 在每个集群主机上的服务以及插入的资源传感器 RESS 同时也提供对主机和作业的监控服务，监控的内容包括：

- 主机系统负载和状态
- 通过资源传感器 RESS 监控的特殊资源
- 作业状态
- 任务进程和资源使用，包括 CPU、GPU、内存，交换区、线程数、网络 IO、和存储 IO
- 任务状态

监控数据载入 Elasticsearch 数据库里，可以通过像 Kibana、Grafana 这类标准图形监控分析软件产生图标和进行数据分析。

在 Kibana 和 Grafana 里定制图标也可以在监控用的 Web 门户 SkyForm Portal 里展示。

2.3. 用户权限管理

SkyForm AIP 使用操作系统中的用户管理系统（LDAP，AD，NIS 等），自动同步用户信息和用户组信息，使用操作系统里的用户名和密码对用户进行认证。每个用户组可以指定一个组管理员，组管理员可以访问组内用户的任务。

SkyForm AIP 用户组定义的例子：

```
usergroups:
  - name: g1
    members: u001 u002
    administrator: u001
  - name: g2
    members: g1 u004
    administrator: u004
  - name: g3
    members: '@system'
    fairshare: '[default,1]'
    administrator: cadmin
```

组管理员

组可以包含其他子组

组成员从操作系统里获得

组成员平等资源

3. 集成深度学习框架

3.1. TensorFlow

TensorFlow 是 Google 开源的机器学习和深度神经网络库，执行和伸缩性好。灵活的架构能够运行在个人电脑，服务器集群和移动设备上的单个或多个 CPU，GPU 或 TPU 上。TensorFlow 社区活跃开放，追随者众多，是 GitHub 关注度最高的深度学习项目。

TensorFlow 能够支持广泛的应用，比如 Google 搜索，Android 应用商店推荐，语音处理，图像识别，机器翻译，视频目标检测，增强学习等。TensorFlow 开发了可视化工具 TensorBoard，既可以显示神经网络结构，又可以显示训练和测试过程中各层参数的变化情况，用于更好地理解，调试和优化网络。TensorFlow 还开发了机器学习模型 serving 的高性能开源库，可以将训练好的模型快速部署服务上线，并支持模型热更新和自动模型版本管理。TensorFlow 的不足之处主要有速度比其他框架慢，内存资源占用多，静态图框架调试困难，版本更新快、兼容性问题多，很多的接口更新或者丢弃等。

3.2. Caffe

Caffe 是一款出现时间较早十分知名的深度学习框架，由伯克利 AI 研究所和社区贡献者开发。Caffe 的使用比较简单，无需编写代码即可进行模型训练，运行速度快，同时还有十分成熟的社区。Caffe 维护了一个 Model Zoo，许多论文作者会将最前沿的模型发布到这里，模型与相应优化都是以文本形式而非代码形式给出，其他用户可以轻松稳定复现前沿模型。Caffe 广泛应用于机器视觉，但不适用于文本，声音和时间序列数据等其他类型的深度学习应用。Caffe 原生支持 CPU/GPU 的单机和分布式模式，不支持多机分布式模式，依赖第三方开发的版本（比如英特尔开发的基于 MPI 的多机版本）。Caffe 的不足之处还包括更新放缓，框架设计带来的灵活性缺失和扩展困难，不提供商业支持等。

3.3. PyTorch

PyTorch 是 Facebook 开源的深度学习框架，能够在强大的 GPU 加速基础上实现张量和动态神经网络。PyTorch 与机器学习第一大语言 Python 深度结合，平滑地与 Python 数据科学堆栈结合，接口易于使用。PyTorch 不需要预定义神经网络图，而是提供了一个框架，可以自由地定义和更改神经网络的结构，甚至在运行时动态修正模型结构而不影响其他计算，降低了调试的难度。PyTorch 易于构建新颖甚至复杂的神经网络，支持动态图的灵活性非常适合学术研究开发新模型。PyTorch 支持多机分布式模式，但是没有采用 TensorFlow 和 MXNet 的 PS-Worker 模式，依赖于 TCP 或 MPI 或 Facebook

孵化项目 gloo，只有 gloo 支持 GPU。PyTorch 的不足之处主要有框架比较新，2017 年 1 月才开源，现在最新版本发布为 0.4.1，使用者较少，强大的社区有待形成。

3.4. MXNet

MXNet 是灵活且高效的深度学习库，Apache 孵化器项目，中立，完全靠社区推动，也被 Amazon 选为 AWS 主要支持的深度学习平台。MXNet 平台特性与 TensorFlow 最相近，有完整的多语言前端，应用场景从分布式训练到移动端部署都覆盖，整个系统全部模块化，适合快速开发，同时又具有轻量级，速度快，内存占用小的优势。MXNet 的不足之处主要有缺乏完善高质量的文档，版本更新快、兼容性问题，社区规模较小且松散（主要开发者背景不同，由 “民间” 开发维护），缺乏商业应用等。

4. 集成机器学习开发环境

4.1. Jupyter Notebooks

Jupyter Notebooks 是数据科学/机器学习社区内一款非常流行的开源 web 编辑器，适用于 Python 程序的开发， 调试及运行。

它提供了一个环境，用户无需离开这个环境，就可以在其中编写代码、运行代码、查看输出、可视化数据并查看结果。因此，这是一款可执行端到端的数据科学工作流程的便捷工具，其中包括数据清理、统计建模、构建和训练机器学习模型、可视化数据等等。

以下常用的 python 算法库可以通过 Jutyper notebooks 来调用和运行：

4.1.1 Scikit-learn （通用算法库）

Scikit-learn 是开源的 Python 机器学习库，是一个完整的机器学习流程框架，提供了大量用于数据挖掘和分析的工具，包括数据预处理、交叉验证、算法与可视化算法等一系列接口。Scikit-learn 的基本功能主要被分为六个部分：分类，回归，聚类，数据降维，模型选择和数据预处理。Scikit-learn 定位于通用传统机器学习库，几乎覆盖了机器学习所有的主流算法，有很多高质量模型易于复用，但相对保守，只做机器学习领域的扩展，只采用经过广泛验证的经典算法。Scikit-learn 倾向于使用者自行对数据处理，而以 TensorFlow 为代表的深度学习库会自动从数据中抽取有效特征。Scikit-learn 的模块高度抽象化，例如一个分类算法可以用几行代码完成，这种抽象化限制了使用

者的自由度，但是大大降低了机器学习的使用门槛。Scikit-learn 主要适合中小型的实用机器学习项目，数据量不大且需要使用者手动对数据进行处理，这类项目通常只需单机环境，在 CPU 上就可以完成，对硬件要求低。

4.1.2. Keras

Keras 是用 Python 编写的高级神经网络 API，能够以 TensorFlow，CNTK，或者 Theano 作为后端运行，可以说是站在巨人肩膀上的设计。Keras 把用户体验放在首位，提供一致且简单的 API，易学好懂，可以实现简单而快速的模型设计，用户友好，高度模块化，易扩展，同时支持卷积神经网络和循环神经网络，可以在 CPU 和 GPU 上无缝运行。Keras 由 Google 软件工程师开发，作为高层神经网络 API 而不是单独的深度学习框架，Keras 发展迅速，有可能成为用于开发神经网络的标准 Python API。Keras 的不足之处主要有速度慢，作为中间层比单独使用 TensorFlow，CNTK 或者 Theano 要慢；为了扩展性好，大多数用 Python 实现，在性能和内存管理方面缺乏效率。

4.1.3. MLlib （分布式算法库）

MLlib 是 Spark 对常用机器学习算法的开源分布式实现库，目标是使实用的机器学习算法可扩展并容易使用。Spark 是一个专门针对大量数据处理的通用的快速引擎，其基于内存的计算模型天生擅长机器学习算法的迭代计算，所以 Spark 是在大数据训练样本下的分布式机器学习理想平台，适用于工程化的实践项目。MLlib 是 Spark 的可以扩展的机器学习库，包括分类，回归，聚类，协同过滤，降维和关联分析等算法。MLlib 提供多种语言支持（Python，R，Scala，Java），对于处理大规模数据速度快。但是 MLlib 每个类别的算法不够丰富，实现的都是些基本算法，如果要把基本算法改进为想要的模式，学习门槛高，需要花费大量时间和精力。机器学习算法的单机和并行化版本的实现是完全不同的，Scikit-learn 的单机算法并不能简单的移植到 Spark MLlib。

4.2. RapidMiner Studio

RapidMiner Studio 是一款世界领先的数据挖掘图形化工具，免费提供数据挖掘技术和库。在一个非常大的程度上有着先进技术，特点是图形用户界面的互动原型。

RapidMiner Studio 是可以进行机器学习、数据挖掘、文本挖掘、预测性分析和商业分析的、具有拖拽功能的图形化工具。可以让分析师可以轻松地设计从混合到建模到部署的预测性分析流程，也可以让企业机构通过使用预测性分析来优化业务，从而获取竞争优势。提供了企业所需的高级分析功能，它可以用于提高市场回应率、降低客户流失、检测机械故障、计划预测性维护以及检测错误等

5. 集成 CAE 仿真软件和 EDA 软件

SkyForm AIP 与 CAE 仿真软件和 EDA 的集成有三种方式：

（1）用户通过 SkyForm AIP 门户可以填写应用参数后直接递交 CAE 应用作业。SkyForm AIP 的门户里已有的作业递交页面包括：ANSYS，FLUENT，ABAQUS，NASTRAN，LS-DYNA，-STAR-CCM+，OptiStruct，SIMPACK 等。

（2）用户在厂商的图形应用里（如 ANSYS WorkBench、Ansys HFSS、FLUENT Launcher 等）里直接递交仿真作业。由于 SkyForm AIP 提供与 LSF 兼容的部分命令，用户选择 LSF 作为后台任务管理，即可将作业直接递交给 SkyForm AIP。

（3）通过命令行递交作业，或在有集成的应用命令行里直接调用 AIP 命令递交作业，如 Synopsys SiliconSmart (通过 Synposys CDPL)。

6. 支持 MPI 作业

在 MPICH2，MVAPICH2 和 Intel MPI 等使用 Hydra 的 MPI 环境中，Hydra 自动调用 SkyForm AIP 来启动 MPI 远程任务。这类 MPI 可直接运行以下作业定义提交作业：

```
{
  "Command": "mpirun mympi",
  "Interactive": false,
  "MinNumSlots": 8
}
```

在以上示例中，递交一个 8 路并行的 MPI 作业。mpirun 会检测到运行环境为 SkyForm AIP，将 SkyForm AIP 调度后设置的环境变量转换为 MPI 作业调度至其上的主机列表，然后调用 SkyForm AIP 的远程任务工具（而不是 SSH）来启动远程 MPI 任务。

当作业被 kill 时，mpirun 也会将 kill 动作传送给所有 MPI 任务。

对于其他类型的 MPI 作业，可用包装脚本的方式实现以上的集成工作。其他不用 Hydra 的 MPI 可以通过脚本集成。

7. 支持大数据应用

SkyForm AIP 支持 Spark 单机模式和集群模式。

8. 支持交互式图形作业

进入系统，在应用中选择远程图形终端的图标；进入任务监控页面后，点击“远程桌面”，系统为用户动态生成图形终端桌面，用户可以在桌面中运行交互式图形应用。
远程图形终端支持多种 3D 远程可视化技术，包括 AWS 的 NICE DCV 和 TurboVNC 等。

9. 用户门户

用户通过 web 门户登录可以

- 启动应用任务，监控应用任务运行情况，暂停、恢复、杀掉、重启任务。
- 对服务器上的文件进行访问和操作，上传下载文件
- 启动和访问远程桌面
- 使用远程文字终端