

# 独立组网的5G核心网实现优化



## 简介

以 NSA (非独立架构) 还是以 SA (独立架构) 开启 5G 网络之门, 一直是通信业界目光聚焦的焦点。采用独立架构模式组网, 不仅可体现 5G 的技术优势, 更能为用户带来多样化的服务类型。但新的模式也带来新的挑战, 通信业界需要设计构建全新的网络架构与之相对应。由中国移动牵头联合全球 26 家运营商及网络设备商提出的基于服务的架构 (Service Based Architecture, SBA) 就是适应了这一趋势, 不仅成为 5G 核心技术之一, 同时也是构建5G独立架构组网方案的关键技术。

SBA 架构面向原生云 (Native Cloud) 设计, 借鉴了互联网领域中面向服务的架构 (Service-Oriented Architecture, SOA)、微服务架构等成熟理念, 并结合通信网现状、特点和发展趋势, 以软件服务的概念重构 5G 核心网, 对 5G 核心网控制面各网络功能实施服务化定义, 同时提供一系列的基于服务的接口 (Service Based Interface, SBI), 具备灵活化、开放化以及智能化的特点, 并可根据需求进行灵活部署和扩展。

本白皮书是中国移动与英特尔公司共同起草的《独立组网的 5G 核心网实现优化》的简版, 描述了 5G 核心网中 SBA 架构的基本组成, 以及基于该架构的一系列实现方法, 并归纳各自优缺点。同时基于这些实现方法, 本文提出多个 SBA 架构方案实现原型, 并给出这些方案原型在不同场景下的性能测试数据和相关的技术优化方案。最后, 结合上述的方法与测试结果, 本文建议了一种高性能 SBA 架构实施方案, 并指出了长期演进的方向。

## 基于服务的架构 (SBA)

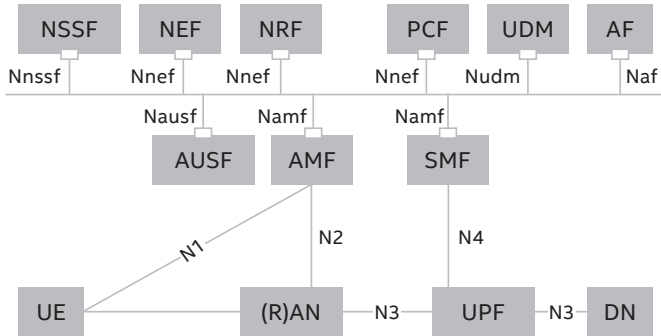
SBA 架构包含了“网络功能服务化”和“基于服务的接口”两大元素, 前者是指其将传统网元设备的功能, 例如会话管理、移动管理、策略控制等, 以软件的方式定义为若干个网络功能, 例如 AMF、SMF、UDM、NRF 等, 这些网络功能间的通信与交互通过服务调用的方式来实现。

后者是指每个网络功能对外都具备通用化、基于服务的接口 (SBI), 可以被经过授权的网络功能或服务灵活调用。这些接口使用了一系列通用化协议, 包括使用 TCP 协议和 HTTP/2 协议, 以 JSON 作为序列化协议, 以 OpenAPI 作为接口描述语言 (Interface

## 目录

简介.....	1
基于服务的架构 (SBA) .....	1
5G 核心网中 SBA 结构的实现方式..	2
集成式 SBA 架构实现.....	2
分布式 SBA 架构实现.....	2
部署方案原型.....	3
集成式容器/虚拟机部署方案 .....	3
分布式容器/虚拟机部署方案 .....	3
性能特征与测试结果.....	3
测试环境搭建 .....	3
测试方案与结果.....	3
优化方案.....	4
总结.....	4
参考文献.....	4

Definition Language, IDL), 允许在不访问源代码或底层服务文档的情况下寻址和调用服务功能。随着技术与标准的日益完善, 服务化接口正逐渐取代 3GPP 中定义的上一代接口, 如 Diameter, SCTP 等, 成为下一代移动通信系统的核心技术之一。



图一 3GPP 技术规范 (TS 23.501) 建议的 5G 系统架构

### 5G 核心网中 SBA 的实现方式

在整体功能上, SBA 可分为两大类组件。一类为接口组件, 其提供了多种功能, 包括 HTTP 服务器功能、收发 HTTP 请求/响应功能, 以及解析 API 端点中内置 HTTP 请求的功能; 另一类为逻辑组件, 其可以处理来自 5G 核心网应用以及特定网络功能 (例如 SMF, UDM, AMF) 的请求。在本文中, 我们设想了以下两种在 5G 核心网控制面中实现 SBA 的方式:

- 集成式实现: 接口和逻辑组件可以整体集成到应用中, 并驻留在同一物理服务器中, 且是同一虚拟机或容器的一部分。
- 分布式实现: 接口和逻辑组件分布在各自独立的物理服务器中, 并通过专用接口相互连接。

#### 集成式 SBA 架构实现

集成式 SBA 架构实现方式的特点, 在于功能自治而不必依赖其他网络功能。基于这一方式, 在集成式 SBA 实现方式中, 对每个控制面网络功能, 例如 AMF, SMF, PCF, AF 等, 都能够提供基于 SBA 的接口和逻辑组件来与其他服务进行交互, 每个网络功能都可实现功能自治。

接口和逻辑组件既可通过函数调用, 进行静态链接或编译, 也可通过多线程的方法, 使用软件队列或共享内存结构进行消息传递, 且消息传递的方式有嵌入式请求处理和基于工作线程的请求处理两种。嵌入式请求处理是将 5G 核心网功能库化, 并集成了 HTTP 服务器处理流程。接口组件依次提供 HTTP 服务器和相关的 API 端点处理消息, 并以库函数的方式调用 5G 核心网控制面处理流程相关的逻辑组件, 这种实现方法在进行功能、服务消息处理时引入了“运行到完成 (Run To Complete, RTC)”模型。

优点	缺点
<ol style="list-style-type: none"> <li>1. 不同的供应商产品, 只要符合 IETF*、3GPP*、5GC*、OpenAPI* 等标准, 就可以方便地进行集成, 互操作性良好;</li> <li>2. 库函数的引入使缓冲区数据包拷贝数量最少, 相关 API 端点解析负载也得以大大降低, 实现性能提升;</li> <li>3. 集成式方案对于网络功能虚拟化 (NFV) 平台更加友好, 易于配置、连接和部署, 而 NFV 正是 5G 时代的核心技术之一。</li> </ol>	<ol style="list-style-type: none"> <li>1. 消息处理采用了 RTC 模型, 可能会因 HTTP 接口耗费大量时间在消息处理上而降低效率, 造成性能隐患;</li> <li>2. 在功能/服务维护期间, 整体网络功能将不可用。</li> </ol>

表一 集成式 SBA 架构实现的优缺点

基于工作线程的请求处理方法可在一定程度上克服嵌入式请求处理方式中可能存在的性能隐患。在这一方法中, 接口和逻辑组件被分离为各自具有可通信存储器映射接口 (如 SW 队列) 的独立应用/进程。这一机制使得接口组件和逻辑组件彼此独立, 并可在不同的工作内核、线程上完成 5G 核心网特定的消息处理, 其中消息可以通过共享内存接口传递, 而消息的元数据则通过内存映射接口传递。

#### 分布式 SBA 架构实现

在这一实现方式中, 部署在电信运营商数据中心/中心局的接口组件可以通过专有接口, 如 UNIX 或 TCP/IP 套接字接口, 以消息代理的方式将消息发送给各个逻辑组件中基于服务的 API 端点。

接口组件的 API 端点会接收所有 HTTP 请求, 并打上相应的逻辑组件目的地标识, 然后在 UNIX 或 TCP / IP 接口上传输请求。各个网络功能, 例如 AMF、SMF 等, 可以作为单个进程, 在同一服务器或不同服务器上启动。这种方法可以将接口组件和逻辑组件在不同的物理服务器上独立部署。

优点	缺点
<ol style="list-style-type: none"> <li>1. 具有专有接口的接口组件可以连接在一起实现完整的控制面解决方案, 而无需每个网络功能都部署对应的接口组件。同时, SMF、AMF 等功能/服务可以与它的接口组件分离, 并且可以彼此独立地升级或更新换代。例如, 接口组件可以在不影响逻辑组件的情况下执行技术升级。</li> </ol>	<ol style="list-style-type: none"> <li>1. 网络功能之间的通信必须基于同样的方法, 这就需要所有共享服务化接口的接口组件来自于同一供应商, 不利于互操作性;</li> <li>2. 网络套接字通信需要在从内核接口到用户应用的每个层面上都进行数据包拷贝, 这将大大提升系统资源占用率;</li> <li>3. 接口组件的 API 端点可能承载较高的解析负载。</li> </ol>

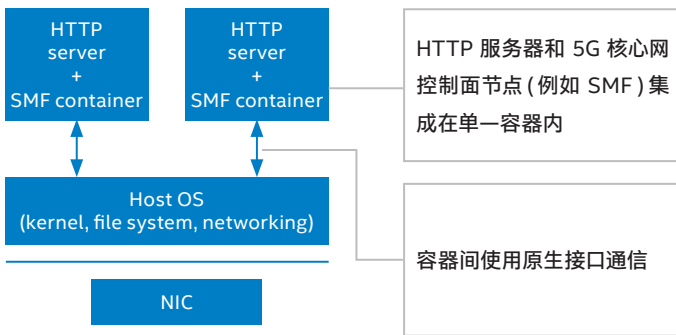
表二 分布式 SBA 架构实现的优缺点

## 部署方案原型

基于上述 2 种 SBA 架构的实例化，可以提出以下几种可行的部署方案原型：

### 集成式容器/虚拟机部署方案

本方案是集成式 SBA 架构的实例化原型，集成了 HTTP 服务器和控制面（如 SMF、AMF 等）的 5G 核心网控制面，可以在虚拟机或容器中进行部署。虚拟机/容器通过网卡上的 SR-IOV 技术来实现性能的提升，并可考虑使用 virtio-net 或自适应虚拟功能（AVF）来实现更强的可扩展功能，从而支持真正的原生云实施。同时由于每个网络功能都具备本地化的 HTTP Web 服务接口功能，因此该部署方案还可实现各 5G 核心网各种控制面功能之间的完全互操作性。



图二 集成式容器 SBA 架构部署方案

### 分布式容器/虚拟机部署方案

本方案是分布式 SBA 架构的实例化原型。方案中的 HTTP 服务器功能在物理上可以与承载 5G 核心网控制面功能分别部署在不同的虚拟机或容器中。HTTP 服务器虚拟机和控制面功能虚拟机之间的接口通过专用的网络套接字接口进行通信。

此外，还有一种在传统分布式方案基础上衍伸而来、具备容器间内存映射接口的分布式容器部署方案。在该方案中，HTTP 服务器容器和承载 5G 核心网控制面功能容器使用内存映射接口进行通信。这可减少数据缓冲区拷贝的数量，从而提高消息请求处理的性能，并获得更低的延迟。

## 性能特征与测试结果

### 测试环境搭建

为对上文所述的几种部署方案原型进行性能验证，本文设计了几类相关的性能测试场景，并将测试目标聚焦在事务吞吐率和处理器负载两个关键指标上。测试方案采用两台独立的基于英特尔® 志强® E5 处理器，并通过 25GbE 以太网交换机互连的物理服务器执行。测试服务器 1 作为待测设备（DUT），运行接口组件和逻辑组件，并可进一步配置为前文所述各类部署方案的场景。在测

试中，测试服务器 2 会向测试服务器 1（待测设备）发起请求并接收响应，来测量相应的请求数量与响应总数等结果，以确定测试服务器 1（待测设备）在不同部署方案场景下的整体性能。

操作系统(服务器/客户端)	Fedora 26
HTTP2/O 服务器	NGINX (Ver 1.12)
后端应用	FastCGI (v2.4.0)
HTTP 客户端	H2load and CURL
JSON 解析器	Rapid JSON (v1.1.0)

表三 用于基准测试的软件工具包配置

### 测试方案与结果

测试方案一：测试服务器 1（待测设备）的一个处理器内核被配置为执行接口组件，而另一个处理器内核被配置为执行逻辑组件。测试包含了两个场景：

1. 集成式容器/虚拟机部署方案，且接口组件和逻辑组件之间建立了基于套接字的通信进程。接口和逻辑组件在各自独立的处理器内核上执行，但执行在同一进程空间中；
2. 分布式容器/虚拟机部署方案，接口和逻辑组件在独立的虚拟机或容器中实例化，并通过专门的基于套接字的通信进程进行交互。

测试方案二：测试服务器 1（待测设备）的一个处理器内核同时执行接口组件和逻辑组件。该测试体现了接口和逻辑组件被紧密集成在一起的系统实现方式，因此，该实现方式中两个组件将被配置在同一处理器内核上执行，并且可通过多核实现可扩展性。

	测试方案一	测试方案二
事务吞吐率(每秒请求数量)	13.5 K	117 K
HTTP2.0 工作内核处理器负载(内核1)	79%	81%
后端应用处理器负载	34%	NA (含在工作内核处理器负载项中)

表四 两种测试方案结果对比:

根据测试结果，下表对 5G 核心网 SBA 架构下的不同实现方案的关键指标进行了对比：

	互通性	性能	延迟
集成式容器/虚拟机部署方案	具备	最高	低
分离式容器/虚拟机部署方案	不具备	低	高
具备容器间内存映射接口的分布式容器部署方案	不具备	中等	中等

表五 5G 核心网 SBA 架构不同实现方案的关键指标对比

## 优化方案

通过对 SBA 架构实际部署和测试中积累的经验进行深入分析和研究,可以看到,进一步对其各个组件开展优化,可使系统获得更好的整体性能。

1. 将 HTTP 和应用程序工作线程分别绑定到独立的处理器内核可更有效地利用本地化数据和处理器的高速缓存。一项基于 NGINX HTTP 服务器的测试显示,将 NGINX 工作内核绑定到特定处理器内核时,可获得更优的性能。
2. 在单个进程中嵌入 HTTP 和消息请求/响应处理过程虽然是良好的设计,但有时可能会限制实现方案的可伸缩性和灵活性。将 HTTP 服务器或后端应用的功能独立成动态库或静态库,有助于在保持实现方案可伸缩性和灵活性的同时,获得更好的性能优势。
3. 后端消息处理逻辑和 HTTP 服务器逻辑可分解为不同的工作线程,这样做的好处是消息处理线程在处理前面的消息时,HTTP 服务器逻辑可以继续保持工作状态,消息处理线程与 HTTP 服务器端线程之间保持一个“请求/响应”队列。从 NGINX 的角度来看,这实现了异步消息处理方式,可以获得更高的处理速度。
4. 根据消息类型的不同,将请求消息分发给不同的工作线程,并行地将消息处理负载分配给不同的处理器内核有助于提高整体处理速度。
5. 将 HTTP 服务器线程和消息处理应用线程作为不同的应用,能够带来更高的灵活性。对于高效的“请求/响应”消息交换而言,应用程序(HTTP 服务器和消息处理器)之间的共享内存技术可用于避免进程间的数据/消息复制,也可以使用大页面内存技术在进程之间实现数据包内存共享。

## 总结

一般而言,构建 5G 核心网中的 SBA 架构有集成式或分布式多种方法,并可在基于 NFV 架构的虚拟机或容器中进行部署。本文所述的测试结果表明,集成式 SBA 架构以其高事务吞吐率和低延迟的特性,相比其他方案有着更优秀的性能表现;同时,该方案还可带来更为全面的互操作性。但是,集成式限制了灵活性,所以从长远演进来看,我们可以考虑采用分布式方法,并通过一些优化措施来提高性能,同时保持灵活性。

### 联合编写单位及作者(作者按姓氏笔画排序):

中国移动研究院网络与 IT 技术研究所

孙滔 刘超 陆璐

英特尔

Khaled Qubaiah Balendu Burla 严峰 陈艳庆 贺楠

### 参考文献

- 1) 3GPP TS 23.501 – 5G 系统架构
- 2) 3GPP TS 23.502 – 5G 系统流程
- 3) 3GPP TS 29.500 – 基于服务的架构的技术实现
- 4) NGINX HTTP 服务器 -- <https://nginx.org/en/docs/>
- 5) RapidJSON -- <http://rapidjson.org/>
- 6) OpenAPI – <https://www.openapis.org/>



英特尔技术特性和优势取决于系统配置,并可能需要支持的硬件、软件或服务才能激活。没有计算机系统是绝对安全的。更多信息,请见 Intel.com,或从原始设备制造商或零售商处获得更多信息。描述的成本降低情景均旨在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。

英特尔、Intel,至强是英特尔公司在美国和其他国家的商标。英特尔商标或商标及品牌名称资料库的全部名单请见 intel.com 上的商标。

\*其他的名称和品牌可能是其他所有者的资产。0618/SHA/OGI/XX/PDF