

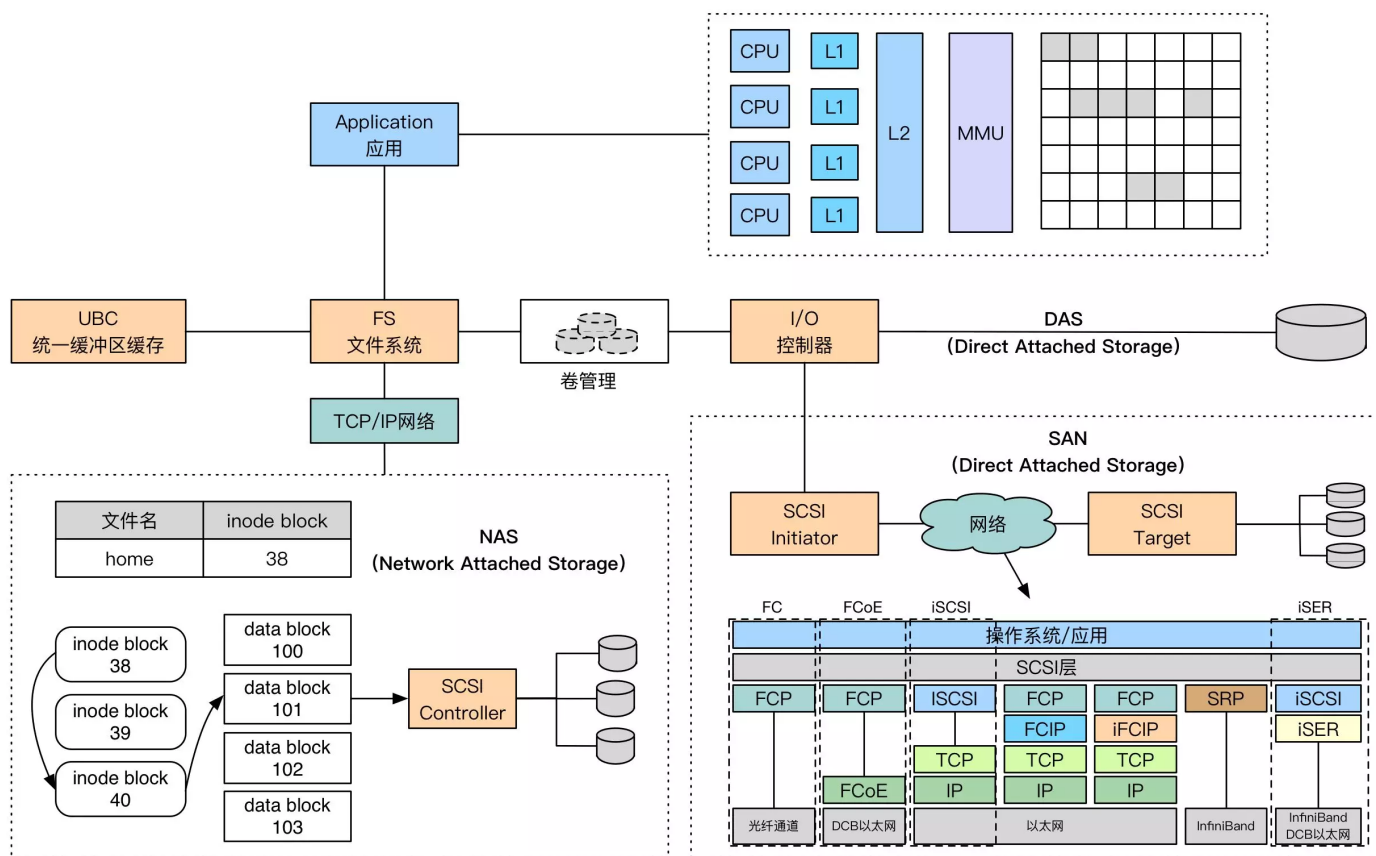
# 【底层原理】存储数据包的一生（上）

2018-01-17 Yikun 码农有道

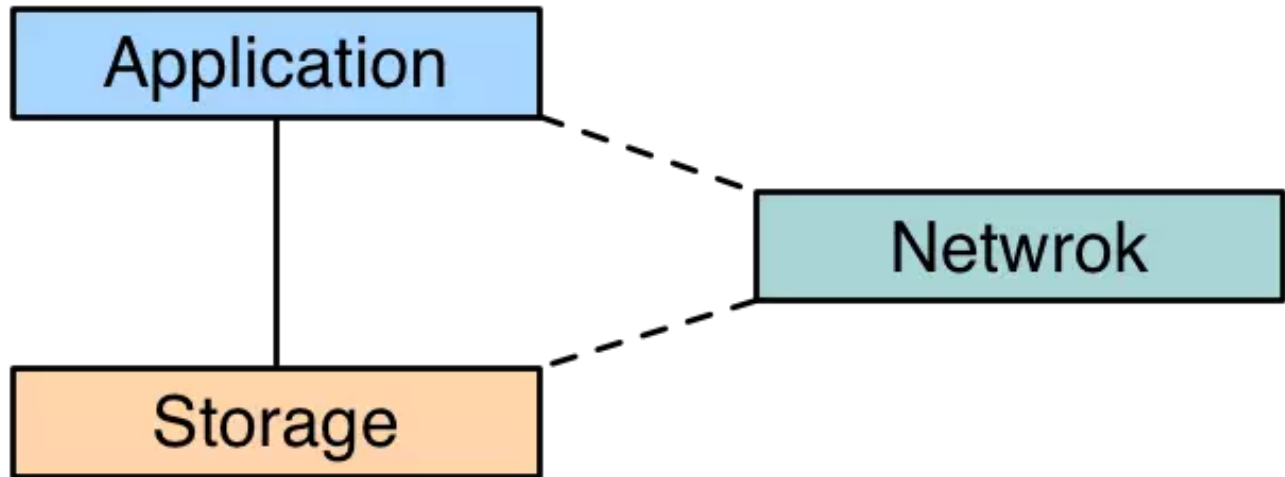
本文来自热心读者投稿，原文链接：<http://yikun.github.io/>

最近认认真真学习了一个叫《Life of a Storage Packet》讲座，借助这个讲座将整个存储的过程理解了下，不放过任何一个有疑问的点。这篇文章算是对讲座的理解和自己收获的总结，同时也为那些对存储系统不够了解又想要了解的初学者，展现一个存储数据包的“生命”。这个演讲主要聚焦在“整体的存储”，强调存储系统中各个基本元素的关系，并且尽可能简单、清楚地用一种不同的方式可视化一些存储的概念。

先上一张大图，可以说这篇文章目的就是解释这个图：



## ▸ 整体视角



存储系统从**整体的分层**一览，包括了主机/应用，存储介质，存储网络。对于存储来说就做了一件事：Here is a bit of data. Hold onto it. Give that same bit back to me when I ask for it.

## ▸ Application视角

每个系统都会有会有很多应用程序运行在CPU上，对于这些Application来说，他们觉得自己有很多很多足够的可用内存。

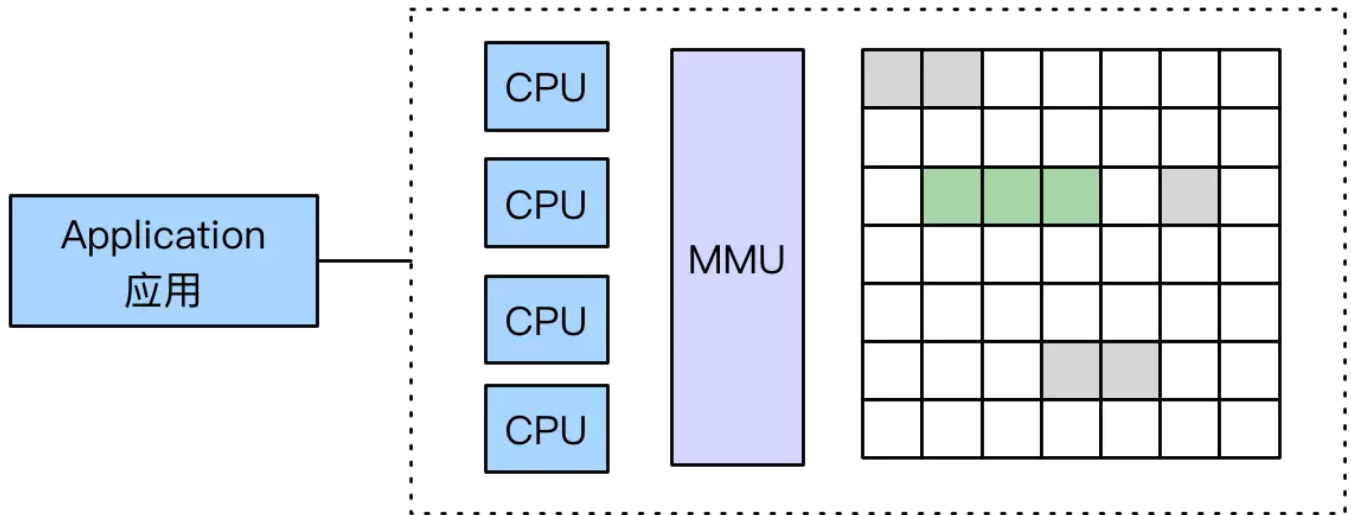
### CPU和MMU

计算机系统有一个“内存管理单元”（**MMU**，Memory Management Unit）的概念，MMU负责与DRAM内存直接通信，并且获得一些可用的“页”

**多租户**：内存被某个进程（比如一个应用）独享，这些内存不能被其他进程重写。

**地址**：将CPU的物理地址翻译成独有的DRAM地址或者是很多行DRAMs，有了MMU以后，对于每个进程来说，他们就像是一个人独占了所有的内存一样。

### 访问内存



Application在他需要访问的时候，获取这些内存，在实际进行访问时，会发生如下事情：

- 1: Application会问操作系统 “Hey，我需要一些可用内存”
- 2: 操作系统向MMU说，“能给我一些可以使用内存页不？”
- 3: MMU把一些可用的内存页给操作系统
- 4: 操作系统把这些可用的内存页给Application。

## 分配内存

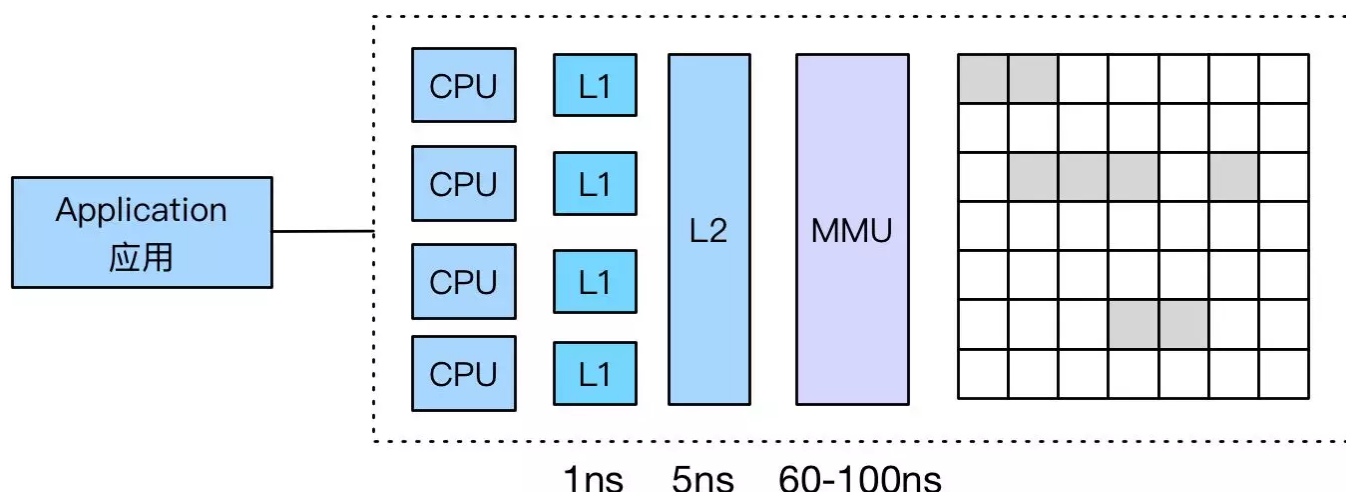
在上图中，获取的那些内存，实际是可以分布在内存中的任何地方的（**非连续**），MMU在把这些内存给应用的时候会进行初始化操作，当然，当应用不再访问这些内存页的时候，也会MMU也会负责回收这些内存。

## 加速访问

如果Application对内存的访问，每次都要都要从MMU获取的话，那太慢了，因此有个经验法则是：Always put storage/memory as close to the CPU as possible

可以说在存储中，改善时间的限制是永恒不变的主题，然后访问DRAM的话，需要60-100ns的时间。我们需要更快的访问，甚至达到“0”时间访问。

## 缓存的魔法



于是在CPU中增加了一些缓存，当然缓存也是分级别的，对于L1缓存大概花费1ns以内，L2缓存大概花费3-6ns，没错，这使得每次访问从60-100ns提升了几十倍！

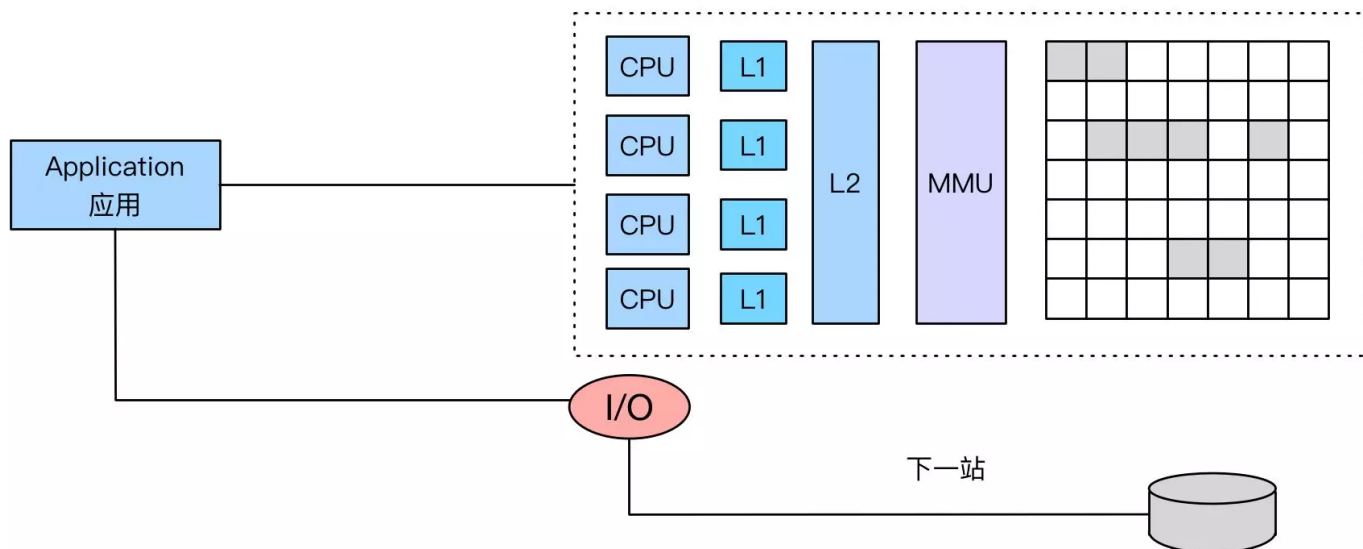
## 更多的空间

如果DRAM中也没有足够的空间时，这是你需要更多的持久化存储，比如说磁盘。当然直接去访问磁盘会非常的昂贵，当然这里指的是时间的花费非常昂贵。

在Jeff Dean大神的Software Engineering Advice from Building Large-Scale Distributed Systems给出了数据：L1缓存、L2缓存、主存、硬盘的访问分别是1ns、5ns、100ns、10,000,000ns级别；

举个比较形象的例子：你需要快递送个包裹，快递一个1公里（L1）、5公里（L2）外的包裹，oops，可能马云爸爸保证当日达；快递100公里（DRAM）的包裹，11点前下单，当日达，不能再快了！；快递1000万公里（Disk）的快递，啊亲，你真的需要快递吗？这可是从地球到月球（38.4万公里）十几个来回的距离啊亲。

## 小结



存储设备和RAM不太一样，不会和CPU直接进行“对话”，而是有一些其他的部分来帮助他们完成对话，我们在下文从[存储视角](#)进一步去讲述。

### 推荐阅读：

[人人都可以做深度学习应用：入门篇（下）](#)

[深入理解系统中log机制（下）](#)

[【C++札记】了解 typename 的双重意义](#)

[【C++札记】C++对象模型之内存布局（2）](#)

[【C++札记】C++对象模型之内存布局（1）](#)

**专注服务器后台技术栈知识总结分享**  
**欢迎关注交流共同进步**



码农有道

coding

码农有道，为您提供通俗易懂的技术文章，让技术变的更简单！