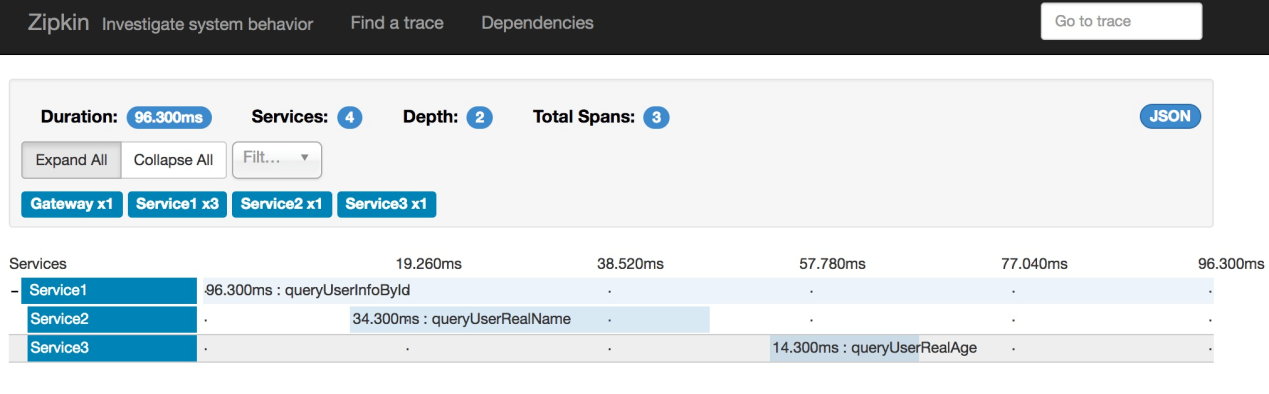


上图的①和⑥是一次完整的RPC调用，它发生在服务器0和服务器1之间，显而易见的是，用于描述该RPC调用的Span的spanId是1000，所以，这是同一个Span的，只是它的数据来源于两台不同的服务器（应用）：服务器0和服务器1

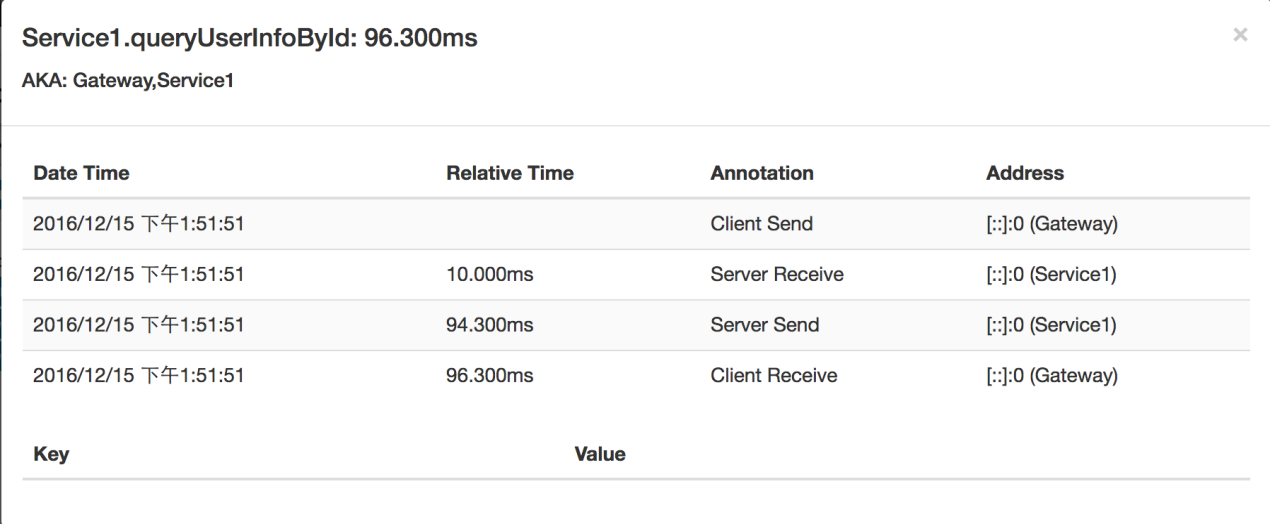
对于步骤①来说，服务器1上的sr减去服务器0上的cs的时间就是约等于网络耗时（这里忽略不同服务器时钟的差异）

我们接着看请求步骤②和④，从跟踪树的层次来说他们属于①下的子调用，所以它们的parentSpanId就是①的1000。步骤②和④都会分别产生一个spanId（上面的1001和1002），所以如上图，看似一次简单的RPC过程，其实共产生了6条Span日志，它们将在Zipkin服务端组装成3个Span。



为啥经历过四个服务器的RPC调用在图中的跟踪树中只有三个节点？因为在跟踪树中， 一个Span（准确的说是一个spanId）只会展现成一个树节点，比如树节点Service1表示了Gateway（服务器0）调用Service1（服务器1）的过程，树节点Service2表示Service1（服务器1）调用Service2（服务器2）的过程。

单击Serice1节点，于是打开了Span的详细信息



Relative Time 是相对时间，表示此事件（cs、sr、ss、cr）已经发生了多久（相对起始时间点），因为Service1是顶级节点，所以第一行的Relative Time是空的，于是乎，该请求的网络耗时（Gateway请求Service1）为10ms，应答的网络耗时（Service1应答Gateway）为96.3-94.3=2ms，所以，从Zipkin目前的页面设计来看，网络耗时只能通过点树节点的详细信息页面来看，而且还需要做简单的计算，并不直观。