**COStream语言执行模型—SDF**

编程语言是底层程序执行模型的体现。COStream采用同步数据流模型[1]（Synchronous Data Flow, SDF）作为语言的执行模型。

在同步数据流模型中，程序由一个带权重的有向图称为同步数据流图表示。图中，每个结点代表了一个计算任务，称为actor，每条边代表了生产者actor与消费者actor之间的数据流动，每条边上具有两个权值，分别代表生产者每次执行后生产数据的个数和消费者每次执行后消耗数据的个数。每个actor都是一个独立的计算单元，它有独立的指令流和地址空间，actor之间的数据流动通过FIFO队列来实现。actor的执行采用数据驱动的方式，只要actor的输入边有足够的数据消耗，它将不停地重复执行并产生数据到输出边。在静态同步数据流模型中，actor的每次执行消耗固定数目的数据，称为消耗率，同样地，actor每次执行产生固定数目的数据称为产生率。

图1例子表示数据流程序有2个actor，actor A和actor B都是独立的计算单元，actor A每次执行向队列缓存中产生3个数据，即生产速率为3，actor B每次执行从队列缓存中消耗2个数据，消耗速率为2。两个actor都采用数据驱动的方式执行，只要数据足够，自动开始执行。



图1：一个SDF图例子

COStream语言描述的数据流图是基于同步数据流图SDF的，两者的对应关系如下：

1. COStream中的operator对应于SDF中的actor；
2. COStream中的stream变量对应于SDF中的FIFO数据边；
3. COStream中的数据流上的window大小对应于SDF中actor对数据的生产和消耗速率。
4. COStream暂时不支持类似StreamIt中feedback loop的图[2]以及带有delay数据边的SDF图。

此外， COStream还增加了对sliding window的支持。