

The Markov Programming Language

Brendan Burns, Emery Berger, Mark Corner

May 16, 2005

1 Introduction

The task of programming high performance network applications is a challenging one. In addition to the complexities of network programming and the need to program the core logic correctly, the task of making the server *high performance*

2 Markov Semantics

2.1 Functional Units

The core of the Markov programming language are its functional units. Each functional unit has a set of inputs that it transforms into a set of outputs. The logic which achieves this transformation is specified by the user.

Functional units are linked together by directed edges. The outputs of one functional unit must match the inputs of the functional unit it is connected to. In any Markov program there are three different types of functional units:

Source Source nodes have input degree of zero and output degree one. The

return control back into the Markov program because to do so would potentially create a cycle in the program's graph structure.

2.4 Runtime Systems

The final choice for a Markov programmer is the selection of the runtime ar-

5 Flow Statements


```
CacheXML: [* , * , * , hi t] = CacheGet;  
CacheXML: [* , * , * , mi ss] = ProcessXML -> CacheStore;
```

```
handle error ReadWrite => FourOhFor;
```

