

Apéndice A

Detalles de implementación de la herramienta de simulación VoDSim

Resumen

Este apéndice mostramos los detalles más relevantes de la implementación del entorno de simulación VoDSim. Explicamos las clases más importantes que componen la aplicación mediante sus diagramas de clases y se comentan los paquetes software estándar utilizados en su implementación. Por último se adjuntan los ficheros de configuración del simulador y varios ejemplos de ficheros para la especificación de los parámetros de la herramienta.

A.1 Detalles de implementación

El simulador ha sido desarrollado de forma modular, basándose en el paradigma de orientación de objetos. El lenguaje escogido ha sido el C++ debido a la gran cantidad de librerías soportadas existentes para este lenguaje. El entorno de desarrollo utilizado ha sido el KDevelop 2.2.3 para el sistema operativo Linux (en concreto la versión Mandrake 9.0 basada en el kernel 2.4.19).

La aplicación resultante está compuesta por alrededor de unas 30.000 líneas de código distribuidas entre más de 40 módulos que componen la presente versión del simulador. A pesar del tamaño del código propietario requerido, siempre que ha sido posible se ha intentado aprovechar los paquetes de software existentes, cuya funcionalidad fuese necesaria para la aplicación. Este es el caso de los paquetes de software STL, libxml2 y Sim++.

La utilización de componentes estándar nos ha facilitado considerablemente la tarea del desarrollo del simulador, aportando una mayor compatibilidad al producto final y facilitando su modificación para los futuros investigadores interesados en su utilización.

STL (libstdc++)

La gestión de las distintas estructuras de datos (listas, pilas, mapas, etc...) requeridas para gestionar los distintos componentes de la simulación se han basado en la utilización de la librería STL (Standard Template Library) [Ste95] [Wei96] [Stl03]. STL es una biblioteca genérica de plantillas que nos suministra un conjunto de algoritmos básicos y estructuras de datos. Al ser STL una biblioteca genérica sus componentes se encuentran completamente parametrizados, de hecho, casi todos los componentes de STL son plantillas.

Brevemente diremos que esta biblioteca provee un conjunto de estructuras de datos y algoritmos sumamente flexibles, eficientes y fiables, que facilitan enormemente la implementación de la mayoría de los algoritmos, al mismo tiempo que mejoran la claridad del código y evitan lidiar con diversos problemas molestos, entre otros el manejo y fragmentación de la memoria. Muchas ventajas que sin duda compensan el tiempo que se debe invertir en aprender a utilizarla antes de poder hacer algo productivo con ella.

Sim++

La librería de software Sim++ soporta la construcción de aplicaciones de simulación basadas en los lenguajes C y C++. Este software soporta los modelos de simulación de colas y dirigido por eventos, siendo compatible con el diseño propuesto para nuestra herramienta de simulación.

Esta librería forma parte de una colección de herramientas para la simulación por ordenador, denominado SimPack [Sim1].

La funcionalidad aportada por Sim++ a nuestra aplicación es la siguiente:

- Gestión de la cola de eventos del simulador
La librería soporta varias técnicas para gestionar los contenidos de la lista de eventos: mediante una lista enlazada, en forma de árbol ó mediante una tabla hash.
- Planificación de eventos basada en prioridades.
- Un conjunto de rutinas básicas para la gestión de números aleatorios y algunas distribuciones (uniforme, exponencial y normal entre otras)

Parser de Xml (libxml2)

La gestión de todos los parámetros de configuración de la simulación se han basado en el lenguaje de etiquetas XML (eXtensible Markup Lenguaje). Para poder procesar este lenguaje dentro del simulador se necesita de un parser de XML que procese los ficheros de configuración y genere un árbol jerárquico con todos los parámetros de configuración. Esta funcionalidad se ha logrado mediante la utilización de la librería libxml2, la cual también ofrece soporte en C++ para el recorrido y búsqueda de información dentro del árbol generado a partir del fichero XML.

A.1.1 Diagrama de clases

En la figura A-1 se muestra la jerarquía de las principales clases que componen el simulador.

La clase central de esta jerarquía es la clase *VoDSimulator*. Esta clase tiene como principal función hacer de intermediario entre el resto de clases que componen el simulador. Además esta clase es el punto de arranque de la simulación y, por lo tanto, la encargada de procesar los ficheros de configuración y de gestionar la generación los ficheros de traza y estadísticas. Esta clase también soporta la gestión de los eventos (creación, ejecución y liberación) a través de un interfase entre el simulador y la librería *Sim++*.

El resto de clases coinciden con cada uno de los módulos que componen el sistema de VoD. La clase *VoDSystem* es el contenedor que agrupa por un lado la arquitectura del sistema y, por otro, los usuarios activos del sistema. La mayoría de la funcionalidad relacionada con la gestión de los usuarios se realiza en esta clase. A continuación, y dependiendo de la arquitectura, tenemos las clases que implementan cada uno de los componentes de la arquitectura: servidores, redes y canales de transmisión (ó flujos de información).

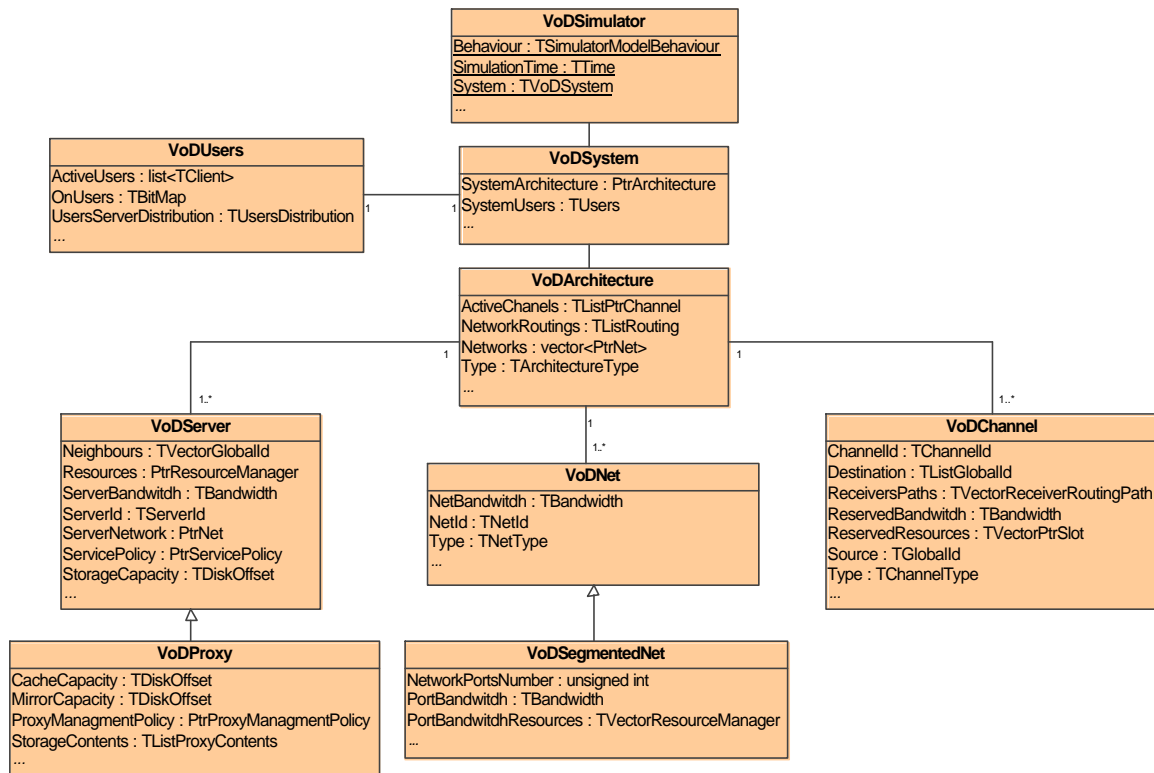


Figura A-1. Clases principales del simulador VodSim

Desde el punto de vista de los servidores, la funcionalidad de los servidores estándar y de los servidores-proxy se soporta en dos clases diferentes. La primera de ellas (*VoDServer*) implementa toda la funcionalidad relacionada con la gestión de los recursos del servidor (ancho de banda de servicio, etc.), y las políticas relativas a la gestión de los distintos tipos de petición que pueden requerir los usuarios (play, stop, cancelación de servicio, etc...). Esta clase soporta la topología de la arquitectura, mediante la definición de la conectividad con los servidores vecinos.

La clase *VoDProxy* implementan toda aquella funcionalidad que varía respecto al servidor estándar. En este caso, se tiene en cuenta la gestión de los contenidos almacenados en el proxy y el soporte necesario para todas las políticas de gestión de los contenidos ubicados en el almacenamiento local (políticas de gestión de la cache y del mirror).

Las clases *VoDNet* y *VoDSegmentedNet* implementan los dos tipos de redes soportadas por el simulador: las redes no segmentadas, en las cuales el ancho de banda es compartido por todos los puertos; y las redes segmentadas, en las cuales el gestor de recursos gestiona un ancho de banda independiente para cada uno de los puertos de la red.

Por último, tenemos la clase *VoDChannel* que implementa un canal de transmisión ó flujo de información entre dos ó más componentes del sistema. Esta clase soporta comunicaciones unicast (1 a 1) y multicast (1 a N).

Arquitecturas

Las clases que definen las distintas arquitecturas modeladas en el simulador (Figura A-2) están basadas en un clase principal (*VoDArchitecture*). Esta clase provee de toda aquella funcionalidad básica para la arquitectura y define el interfase que debe cumplir toda clase derivada. Este interfase define una serie de métodos virtuales cuya implementación final depende del tipo de arquitectura. A través de estos métodos cada una de las arquitecturas soportadas definen la funcionalidad específica que las distingue del resto de arquitecturas (topología, routing, jerarquía de servicio, etc..).

La clase *VoDArchitecture* es la responsable de gestionar los canales de transmisión activos en el sistema y los métodos para localizar el camino entre dos componentes cualesquiera de la arquitectura. Por último sirven de contenedor para los servidores y las redes de comunicación del sistema.

Políticas de gestión

Las políticas de gestión del sistema VoD soportadas por el simulador se pueden dividir en 3 categorías: políticas de servicio, políticas de gestión de la caches de los servidores-proxy y las

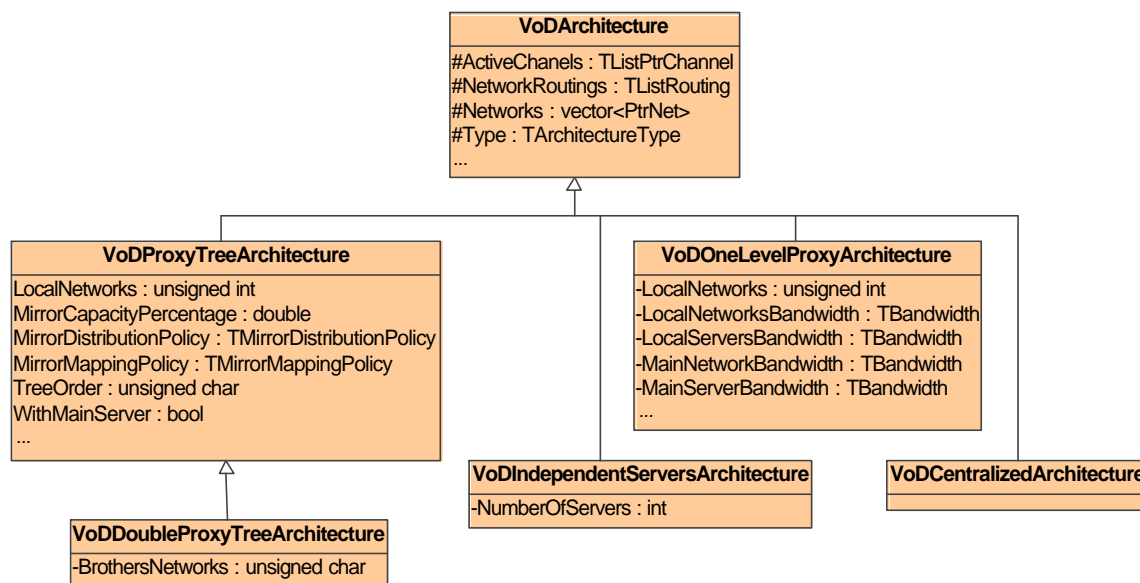


Figura A-2. Diagrama de clases que soportan las arquitecturas LVoD en el simulador VodSim

políticas de gestión de recursos.

- o *Políticas de servicio.*

Todas las políticas de servicio derivan de una clase común *VoDServicePolicy* (figura A-3). A partir de este primer nivel, tenemos la clase *VoDUnicastServicePolicy* que se encarga de soportar la funcionalidad requerida para servir a cada una de las peticiones mediante su propio canal de transmisión (stream).

La siguiente clase de la jerarquía (*VoDMulticastServicePolicy*) soporta un multicast sin retardo. Permite agrupar el servicio de dos ó más peticiones siempre que todas ellas sean recibidas por el servidor correspondiente antes que se haya iniciado el servicio de la primera petición y sin suponer ningún tipo de retardo (salvo el indispensable derivado de la gestión del servidor). Además esta clase permite agrupar toda aquella funcionalidad compartida por las distintas políticas de multicast. De esta clase se derivan las políticas de multicast más específicas: *Batching* y *Patching*.

Por último, se soportan las políticas de servicio combinadas mediante la clase *VoDCombinedServicePolicy*. Las políticas combinadas permitan utilizar de forma conjunta distintas políticas de servicio (unicast y multicast) en función de cual de ellas se adapta mejor a las condiciones específicas del sistema.

- o *Políticas de gestión de la cache*

Las políticas de gestión de la cache en los servidores-proxy también disponen de una clase común que permite concentrar la funcionalidad básica de estas políticas. El simulador soporta actualmente dos políticas diferentes de caching: el caching autónomo y el caching cooperativo, cada una de ellas soportadas por su propia clase (figura A-4).

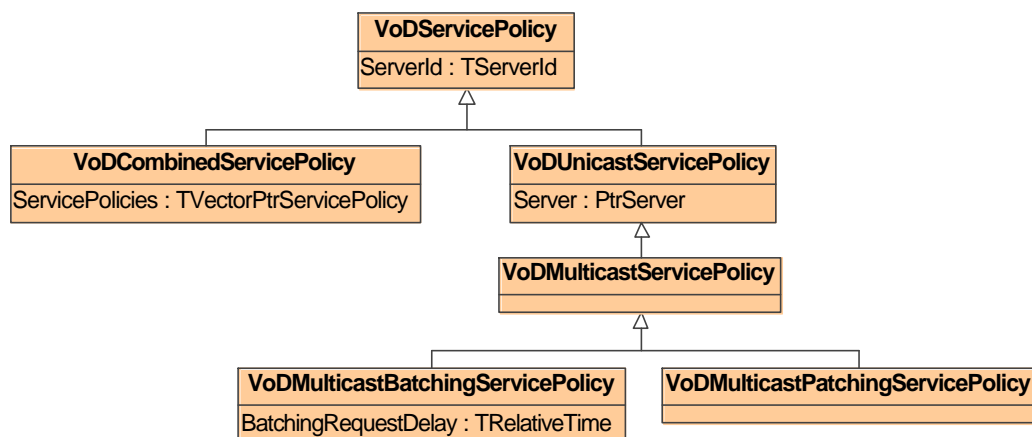


Figura A-3. Diagrama de clases que soportan las políticas de servicio

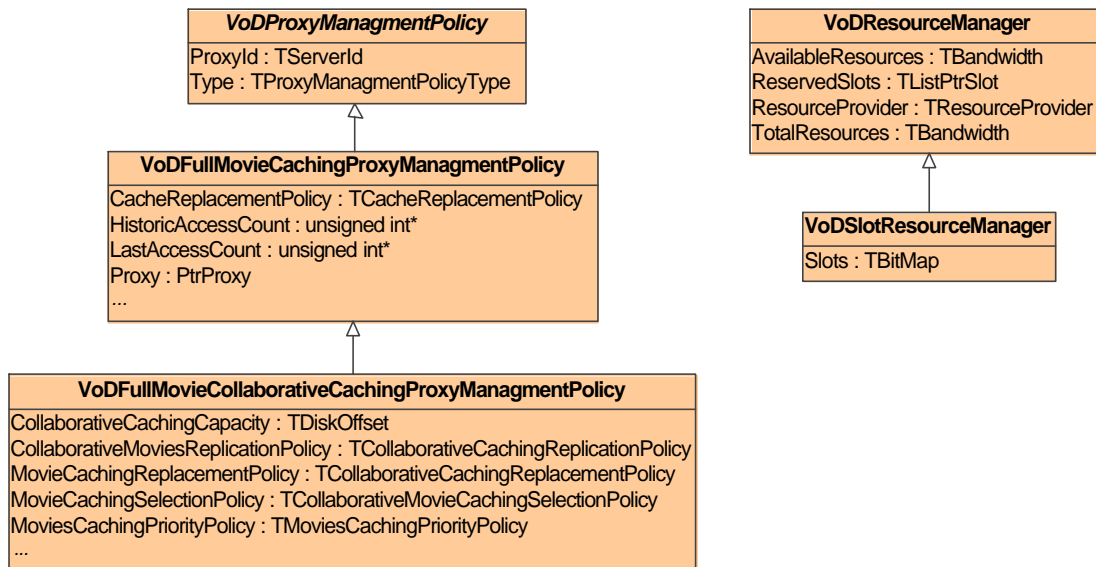


Figura A-4. Diagrama de clases que soportan las políticas de gestión de la cache y gestión de los recursos.

- o Políticas de gestión de recursos

La gestión de los recursos del sistema se basa en dos clases (figura A-4): *VoDResourceManager* que gestiona los recursos mediante un modelo unidimensional y *VoDSlotResourceManager* que gestiona los recursos utilizando el modelo bidimensional.

A.2 Ficheros de configuración de los parámetros de simulación

A continuación se adjunta el fichero de especificación de los parámetros permitidos por la herramienta de simulación (DTD) y dos ficheros de configuración para la simulación de una arquitectura Proxy-Tree y una arquitectura Double P-Tree respectivamente.

A.2.1 Fichero de especificación de parámetros de simulación (DTD)

```

<!--?xml encoding="UTF-8" -->
<!--==== DTD for VoDSimulator Run Parameters =====>
<ELEMENT VoDSimulation (System? | BehaviourModel? | EndCondition? | SinTest? )+ >
  <ATTLIST VoDSimulation TestName CDATA "">
  <ATTLIST VoDSimulation StatisticsFile CDATA "">
  <ATTLIST VoDSimulation OutputFile CDATA "">
  <ATTLIST VoDSimulation PrintEvents ( All | MainEvents | None) *None* >
  <ATTLIST VoDSimulation OutputMode ( Long | Short ) *Long* >
  <ATTLIST VoDSimulation WritesStatistics (Yes | No) *Yes* >
  <ATTLIST VoDSimulation WriteOutput (Yes | No) *No* >
  <ATTLIST VoDSimulation RandomEvents (Yes | No) *No* >
<ELEMENT System (Architecture, Users, MoviesSize?, MoviesPlayRate?) >
  <ATTLIST System MoviesNumber CDATA #IMPLIED>
<ELEMENT MoviesSize (Distribution)>
<ELEMENT MoviesPlayRate (Distribution)>
<ELEMENT Architecture ( CentralizedArch | IndependentServersArch | OneLevelProxyArch | PTreeArch | DoublePtreeArch | ProxyInternetArch ) >
  <ATTLIST Architecture Type (Centralized | IndependentServers | OneLevelProxy | PTree | DoublePtree | ProxyInternet ) #REQUIRED>
  <ATTLIST Architecture UseSegmentedNetworks (Yes | No) *No* >
  <ATTLIST Architecture NetworkPorts CDATA #IMPLIED>
  <ATTLIST Architecture ServerPorts CDATA #IMPLIED>
  <ATTLIST Architecture TrafficBalancing (Yes | No) *No* >
  <ATTLIST Architecture RandomServerSelection (Yes | No) *No* >
<ELEMENT CentralizedArch ( Server | Network | SegmentedNetwork )+ >
<ELEMENT IndependentServersArch ( Server | Network | SegmentedNetwork )+ >
  <ATTLIST IndependentServersArch ServersNumber CDATA #REQUIRED>
<ELEMENT OneLevelProxyArch ( Server | Proxy | Network | SegmentedNetwork )+ >
  <ATTLIST OneLevelProxyArch LocalNetworks CDATA #REQUIRED>
  <ATTLIST OneLevelProxyArch InitCache (Yes | No) *No* >
  <ATTLIST OneLevelProxyArch CalculateBandwidthDistribution (Yes | No) *No* >
<ELEMENT PtreeArch ( Server | Proxy | Network | SegmentedNetwork )+ >
  <ATTLIST PtreeArch LocalNetworks CDATA #REQUIRED>
  <ATTLIST PtreeArch TreeOrder CDATA "2" >
  <ATTLIST PtreeArch WithMainServer (Yes | No) *No* >
  <ATTLIST PtreeArch InitCache (Yes | No) *No* >
  <ATTLIST PtreeArch InitCollaborativeCache (Yes | No) *No* >
  <ATTLIST PtreeArch StorageInitializationStepByStep (Yes | No) *No* >
  <ATTLIST PtreeArch BestCacheMirrorDistribution (Yes | No) *No* >
  <ATTLIST PtreeArch StorageDistributionPolicy (Default | Global | Adaptive) *Default* >
  <ATTLIST PtreeArch MirrorDistributionPolicy (ByIndividualMovies | ByIndividualMoviesPostOrder | ByBalancedMoviesGroups | ByBalancedMoviesGroupsPostOrder) *ByIndividualMovies* >
  <ATTLIST PtreeArch MirrorMappingPolicy (Secuencial | ByMoviesDistance) *ByMoviesDistance* >
  <ATTLIST PtreeArch MirrorDistributionByLevels (Yes | No) *No* >
  <ATTLIST PtreeArch BuildingMirrorStepByStep (Yes | No) *No* >
  <ATTLIST PtreeArch BuildMirrorRandomly (Yes | No) *No* >
  <ATTLIST PtreeArch MoviesToMapEveryStep CDATA "0" >
  <ATTLIST PtreeArch TakeInAccountCollateralEffects (Yes | No) *No* >
  <ATTLIST PtreeArch CollaborativeMulticast (Yes | No) *No* >
  <ATTLIST PtreeArch UnicastCollaborativeMulticast CDATA "50" >
  <ATTLIST PtreeArch AdaptiveMirroring (Yes | No) *No* >
  <ATTLIST PtreeArch MultipleTestsBestDistribution (Yes | No) *No* >
<ELEMENT DoublePtreeArch ( Server | Proxy | Network | SegmentedNetwork )+ >
  <ATTLIST DoublePtreeArch LocalNetworks CDATA #REQUIRED>
  <ATTLIST DoublePtreeArch TreeOrder CDATA "2" >
  <ATTLIST DoublePtreeArch WithMainServer (Yes | No) *No* >
  <ATTLIST DoublePtreeArch BrothersNetworks CDATA #REQUIRED>
  <ATTLIST DoublePtreeArch InitCache (Yes | No) *No* >
  <ATTLIST DoublePtreeArch InitCollaborativeCache (Yes | No) *No* >
  <ATTLIST DoublePtreeArch StorageInitializationStepByStep (Yes | No) *No* >
  <ATTLIST DoublePtreeArch BestCacheMirrorDistribution (Yes | No) *No* >
  <ATTLIST DoublePtreeArch StorageDistributionPolicy (Default | Global | Adaptive) *Default* >
  <ATTLIST DoublePtreeArch MirrorDistributionPolicy (ByIndividualMovies | ByIndividualMoviesPostOrder | ByBalancedMoviesGroups | ByBalancedMoviesGroupsPostOrder) *ByIndividualMovies* >
  <ATTLIST DoublePtreeArch MirrorMappingPolicy (Secuencial | ByMoviesDistance) *ByMoviesDistance* >
  <ATTLIST DoublePtreeArch MirrorDistributionByLevels (Yes | No) *No* >

```

APÉNDICE A

```
<!ATTLIST DoublePtreeArch BuildingMirrorStepByStep (Yes | No) "No">
<!ATTLIST DoublePtreeArch BuildMirrorRandomly (Yes | No) "No">
<!ATTLIST DoublePtreeArch MoviesToMapEveryStep CDATA "0" >
<!ATTLIST DoublePtreeArch TakeInAccountCollateralEffects (Yes | No) "No">
<!ATTLIST DoublePtreeArch CollaborativeMulticast (Yes | No) "No">
<!ATTLIST DoublePtreeArch UmbraCollaborativeMulticast CDATA "50" >
<!ATTLIST DoublePtreeArch AdaptiveMirroring (Yes | No) "No">
<!ATTLIST DoublePtreeArch MultipleTestsBestDistribution (Yes | No) "No">
<ELEMENT ProxyInternetArch ( Server | Proxy | Network | SegmentedNetwork | Topology)* >
  <!ATTLIST InternetProxyArch LocalNetworks CDATA #REQUIRED>
  <!ATTLIST ProxyInternetArch WithMainServer (Yes | No) "No">
  <!ATTLIST ProxyInternetArch InitCache (Yes | No) "No">
  <!ATTLIST ProxyInternetArch InitCollaborativeCache (Yes | No) "No">
  <!ATTLIST ProxyInternetArch StorageInitializationStepByStep (Yes | No) "No">
  <!ATTLIST ProxyInternetArch BestCacheMirrorDistribution (Yes | No) "No">
  <!ATTLIST ProxyInternetArch StorageDistributionPolicy (Default | Global | Adaptive) "Default" >
  <!ATTLIST ProxyInternetArch MirrorDistributionPolicy (ByIndividualMovies | ByIndividualMoviesPostOrder | ByBalanceMoviesGroups | ByBalanceMoviesGroupsPostOrder) "ByIndividualMovies" >
  <!ATTLIST ProxyInternetArch MirrorMappingPolicy (Sequential | ByMoviesDistance) "ByMoviesDistance" >
  <!ATTLIST ProxyInternetArch MirrorDistributionByLevels (Yes | No) "No">
  <!ATTLIST ProxyInternetArch BuildingMirrorStepByStep (Yes | No) "No">
  <!ATTLIST ProxyInternetArch BuildMirrorRandomly (Yes | No) "No">
  <!ATTLIST ProxyInternetArch MoviesToMapEveryStep CDATA "0" >
  <!ATTLIST ProxyInternetArch TakeInAccountCollateralEffects (Yes | No) "No">
  <!ATTLIST ProxyInternetArch CollaborativeMulticast (Yes | No) "No">
  <!ATTLIST ProxyInternetArch UmbraCollaborativeMulticast CDATA "50" >
  <!ATTLIST ProxyInternetArch AdaptiveMirroring (Yes | No) "No">
<ELEMENT Server (#PCDATA)>
  <!ATTLIST Server Id CDATA #REQUIRED>
  <!ATTLIST Server Bandwidth CDATA #REQUIRED>
  <!ATTLIST Server ServicePolicy ( Unicast | Batching | Patching) #REQUIRED>
  <!ATTLIST Server ResourceManager (Normal | Slot | Complex) "Slot">
  <!ATTLIST Server StorageCapacity CDATA #IMPLIED>
  <!ATTLIST Server FirstUser CDATA #IMPLIED>
  <!ATTLIST Server LastUser CDATA #IMPLIED>
<ELEMENT Proxy (ProxyManagementPolicy)* >
  <!ATTLIST Proxy Id CDATA #REQUIRED>
  <!ATTLIST Proxy Bandwidth CDATA #REQUIRED>
  <!ATTLIST Proxy ServicePolicy ( Unicast | Batching | Patching) #REQUIRED>
  <!ATTLIST Proxy CachingPolicy ( PullMovieCaching | PullMovieCollaborativeCachingPolicy | PrefixCaching) #REQUIRED>
  <!ATTLIST Proxy ResourceManager (Normal | Slot | Complex) "Slot">
  <!ATTLIST Proxy StorageCapacity CDATA #REQUIRED>
  <!ATTLIST Proxy MirrorCapacity CDATA #IMPLIED>
  <!ATTLIST Proxy MirrorCapacityPercentage CDATA #IMPLIED>
  <!ATTLIST Proxy FirstUser CDATA #IMPLIED>
  <!ATTLIST Proxy LastUser CDATA #IMPLIED>
<ELEMENT ProxyManagementPolicy (PullMovieCachingPolicy | PullMovieCollaborativeCachingPolicy)* >
  <!-- PullMovieCachingPolicy (#PCDATA) -->
  <!-- PullMovieCachingPolicy ReplacementPolicy ( NoReplacement | LessFrequentAccessed | LessFrequentAccessed ) "LessFrequentAccessed" -->
  <!-- PullMovieCachingPolicy ReplacementUmbra CDATA "0" -->
  <!-- PullMovieCachingPolicy OnlyReplaceNotUsedMovies (Yes | No) "No" -->
  <!-- PullMovieCachingPolicy DoAgeing (Yes | No) "No" -->
  <!-- PullMovieCachingPolicy AgeingFrequency CDATA "0" -->
  <!-- PullMovieCollaborativeCachingPolicy (PullMovieCachingPolicy, CollaborativeCachingPolicy) -->
  <!-- CollaborativeCachingPolicy (#PCDATA) -->
  <!-- CollaborativeCachingPolicy CollaborativeCachingCapacity CDATA #IMPLIED -->
  <!-- CollaborativeCachingPolicy CollaborativeCachingCapacityPercentage CDATA #IMPLIED -->
  <!-- CollaborativeCachingPolicy CachingMovieSelectionPolicy ( MostAccessedLimited | BestMovieReplication ) "MostAccessedLimited" -->
  <!-- CollaborativeCachingPolicy FillDistributedMirrorCaps (Yes | No) "No" -->
  <!-- CollaborativeCachingPolicy ReplacementPolicy ( NoReplacement | LessFrequentAccessed | LessPopular ) "LessFrequentAccessed" -->
```

```

<!--ATTLIST CollaborativeCachingPolicy ReplacementUsbral CDATA "0">
<!--ATTLIST CollaborativeCachingPolicy MovieReplicationPolicy ( ConstantReplication | PonderMovieReplication) "ConstantReplication" >
<!--ATTLIST CollaborativeCachingPolicy MoviesReplication CDATA "2">
<!--ATTLIST CollaborativeCachingPolicy StatisticsRefreshPeriod CDATA "60">
<!--ATTLIST CollaborativeCachingPolicy MoviesCachingPriorityPolicy ( Default | None | ByAccessReplicationPercentage | ByMovieDistance | ByMovieDistanceAndReplicationPercentage ) "Default">
<!--ELEMENT Network (#PCDATA)>
<!--ATTLIST Network Id CDATA #REQUIRED>
<!--ATTLIST Network Bandwidth CDATA #REQUIRED>
<!--ATTLIST Network ResourceManager (Normal | Slot | Complex) "Slot">
<!--ATTLIST Network FirstUser CDATA #IMPLIED>
<!--ATTLIST Network LastUser CDATA #IMPLIED>
<!--ELEMENT SegmentedNetwork (#PCDATA)>
<!--ATTLIST SegmentedNetwork Id CDATA #REQUIRED>
<!--ATTLIST SegmentedNetwork Bandwidth CDATA #REQUIRED>
<!--ATTLIST SegmentedNetwork PortsNumber CDATA #IMPLIED>
<!--ATTLIST SegmentedNetwork ResourceManager (Normal | Slot | Complex) "Slot">
<!--ATTLIST SegmentedNetwork FirstUser CDATA #IMPLIED>
<!--ATTLIST SegmentedNetwork LastUser CDATA #IMPLIED>
<!--ELEMENT Topology (Link) * >
<!--ELEMENT Link (#PCDATA) >
<!--ATTLIST Link SourceServerId CDATA #REQUIRED>
<!--ATTLIST Link DestinationServerId CDATA #REQUIRED>
<!--ELEMENT Users (UserRequestDistribution)>
<!--ATTLIST Users UsersNumber CDATA #REQUIRED>
<!--ATTLIST Users UsersServerDistribution (Uniform | NonUniform | OneLevelProxies) "Uniform" * >
<!--ATTLIST Users BufferSize CDATA "5">
<!--ELEMENT UserRequestDistribution (Distribution)>
<!--ELEMENT Distribution (#PCDATA)>
<!--ATTLIST Distribution Type (Default | Constant | Exponential | Erlang | Uniform | Hyperexponential | Normal | Zipf | Poisson ) "Default">
<!--ATTLIST Distribution Mean CDATA #IMPLIED>
<!--ATTLIST Distribution Deviation CDATA #IMPLIED>
<!--ATTLIST Distribution Begin CDATA #IMPLIED>
<!--ATTLIST Distribution End CDATA #IMPLIED>
<!--ELEMENT BehaviourModel (RequestInterval?, RequestMovie?, RenegingInterval?)>
<!--ELEMENT RequestInterval (Distribution)>
<!--ELEMENT RequestMovie (Distribution)>
<!--ELEMENT RenegingInterval (Distribution)>
<!--ELEMENT EndCondition ( ByTime | ByRequestNumber | ByRefusedRequestNumber | BySystemSaturation | ByMaxSystemBandwidth ) * >
<!--ATTLIST EndCondition ForcedEnd (Yes | No) "No" >
<!--ELEMENT ByTime (#PCDATA) >
<!--ATTLIST ByTime Time CDATA #REQUIRED >
<!--ELEMENT ByRequestNumber (#PCDATA) >
<!--ATTLIST ByRequestNumber MaxRequests CDATA #REQUIRED >
<!--ELEMENT ByRefusedRequestNumber (#PCDATA) >
<!--ATTLIST ByRefusedRequestNumber MaxRefusedRequest CDATA #REQUIRED >
<!--ELEMENT BySystemSaturation (#PCDATA) >
<!--ATTLIST BySystemSaturation MaxRefusedRequest CDATA #REQUIRED >
<!--ATTLIST BySystemSaturation MinEndsRequest CDATA "0" >
<!--ELEMENT ByMaxSystemBandwidth (#PCDATA) >
<!--ATTLIST ByMaxSystemBandwidth MaxSystemBandwidth CDATA #REQUIRED >
<!--ATTLIST ByMaxSystemBandwidth MultiplyBySystemNetworks (Yes | No) "No">
<!--ELEMENT SymTest ( ( Simple | Scalability | Mean | Storage | StorageDistribution | OptimalMeanMovieDistance | RequestRate | MeanDistanceCorrelation ), StatisticsFields?, Plot? ) >
<!--ATTLIST SymTest StatisticsFile CDATA "">
<!--ATTLIST SymTest AddStatistics (Yes | No) "No" >
<!--ATTLIST SymTest LaunchSeveralExecutions (Yes | No) "No" >
<!--ATTLIST SymTest NumberOfExecutions CDATA "1">
<!--ATTLIST SymTest StorePartialResults (Yes | No) "No" >
<!--ELEMENT Simple (#PCDATA) >
<!--ELEMENT Mean (#PCDATA) >
<!--ELEMENT Scalability (#PCDATA) >
<!--ATTLIST Scalability MinLocalNetworks CDATA #IMPLIED >
<!--ATTLIST Scalability MaxLocalNetworks CDATA #IMPLIED >
<!--ATTLIST Scalability InclLocalNetworks CDATA #IMPLIED >

```



```

    <!ATTLIST Scalability MinTopologyLevel CDATA #IMPLIED >
    <!ATTLIST Scalability MaxTopologyLevel CDATA #IMPLIED >
<!ELEMENT Storage (#PCDATA) >
    <!ATTLIST Storage MinStorageCapacity CDATA #IMPLIED >
    <!ATTLIST Storage MaxStorageCapacity CDATA #IMPLIED >
    <!ATTLIST Storage IncStorageCapacity CDATA #IMPLIED >
    <!ATTLIST Storage MinMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST Storage MaxMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST Storage IncMirrorCapacityPercentage CDATA #IMPLIED >
<!ELEMENT StorageDistribution (#PCDATA) >
    <!ATTLIST StorageDistribution MinMirroringStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution MaxMirroringStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution IncMirroringStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution IncMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST StorageDistribution MinCollaborativeCachingStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution MaxCollaborativeCachingStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution IncCollaborativeCachingStorageCapacity CDATA #IMPLIED >
    <!ATTLIST StorageDistribution IncCollaborativeCachingCapacityPercentage CDATA #IMPLIED >
<!ELEMENT OptimalMeanMoviesDistance (#PCDATA) >
    <!ATTLIST OptimalMeanMoviesDistance MinStorageCapacity CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MaxStorageCapacity CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance IncStorageCapacity CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MinMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MaxMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance IncMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MinLocalNetworks CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MaxLocalNetworks CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance IncLocalNetworks CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MinTopologyLevel CDATA #IMPLIED >
    <!ATTLIST OptimalMeanMoviesDistance MaxTopologyLevel CDATA #IMPLIED >
<!ELEMENT RequestRate (#PCDATA) >
    <!ATTLIST RequestRate MinRequestRate CDATA #IMPLIED >
    <!ATTLIST RequestRate MaxRequestRate CDATA #IMPLIED >
    <!ATTLIST RequestRate IncRequestRate CDATA #IMPLIED >
<!ELEMENT MeanDistanceCorrelation (#PCDATA) >
    <!ATTLIST MeanDistanceCorrelation MinStorageCapacity CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MaxStorageCapacity CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation IncStorageCapacity CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MinMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MaxMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation IncMirrorCapacityPercentage CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MinFactor1 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MaxFactor1 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation IncFactor1 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MinFactor2 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MaxFactor2 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation IncFactor2 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MinFactor3 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation MaxFactor3 CDATA #IMPLIED >
    <!ATTLIST MeanDistanceCorrelation IncFactor3 CDATA #IMPLIED >
<!ELEMENT StatisticsFields (#PCDATA)>
    <!ATTLIST StatisticsFields All CDATA #IMPLIED >
    <!ATTLIST StatisticsFields LocalNetworks CDATA #IMPLIED >
    <!ATTLIST StatisticsFields ProxyStorageInMovies CDATA #IMPLIED >
    <!ATTLIST StatisticsFields ProxyCachingStorageInMovies CDATA #IMPLIED >
    <!ATTLIST StatisticsFields ProxyCollaborativeCachingStorageInMovies CDATA #IMPLIED >
    <!ATTLIST StatisticsFields ProxyMirroringStorageInMovies CDATA #IMPLIED >
    <!ATTLIST StatisticsFields EffectiveBandwidth CDATA #IMPLIED >
    <!ATTLIST StatisticsFields PetitionsServed CDATA #IMPLIED >
    <!ATTLIST StatisticsFields PetitionsReneged CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemServerActiveLoad CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemNetworkActiveLoad CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemOptimalMoviesMeanDistance CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemMoviesMeanDistance CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemMoviesMeanDistancePond CDATA #IMPLIED >
    <!ATTLIST StatisticsFields SystemMoviesMeanDistancePondWithCaching CDATA #IMPLIED >
<!ELEMENT Plot (#PCDATA) >

```

A.2.2 Fichero de configuración para una simulación basada en un sistema Proxy-Tree

```

<?xml version="1.0"?>
<!DOCTYPE VoDSimulation SYSTEM "VoDSimulator.dtd">
<VoDSimulation TestName="Proxy P-Tree With Patching Saturation 3SP LR63 Seg 1000s S254SR_BP
SD Poisson10 BalTra ByDistance PostOrder Step"
  WriteOutput="No" PrintEvents="MainEvents" OutputMode="Short" RandomEvents="Yes"
  *
  OutputFile="..//Pruebas/Estadisticas/ProxyPTree3SpPatchingMaxBandLR63Seg1000sS
254SR_BP3SDPoisson10_BalTraByDistancePostOrderStep.out"
  StatisticsFile="..//Pruebas/Estadisticas/ProxyPTree3SpPatchingMaxBandLR63Seg1
000sS254SR_BP3SDPoisson10_BalTraByDistancePostOrderStep.stl" >
<SymTest StatisticsFile="..//Pruebas/Estadisticas/ProxyPTree3SpPatchingMaxBandLR63Seg1000
S254SR_BP3SDPoisson10_BalTraByDistancePostOrderStep.dat"
  LaunchSeveralExecutions="Yes" NumberOfExecutions="10" StorePartialResults="Yes" >
  <Sample >
  </Sample >
  <StatisticsFields All="1" >
  </StatisticsFields >
</SymTest >
<EndCondition >
<SymSystemBandwidth MaxSystemBandwidth="1000s" MultiplyBySystemNetworks="Yes" >
<SymMaxSystemBandwidth >
</EndCondition >
<System MoviesNumber="100">
  <Architecture Type="PTree" UseSegmentedNetworks="Yes" NetworkPorts="24" ServerPorts="
3"
  <TreeArch
    TrafficBalancing="Yes" >
    TreeOrder="2" LocalNetworks="63" WithMainServer="No" InitCache="Yes"
    BestCacheMirrorDistribution="Yes"
    StorageDistributionPolicy="Default"
    MirrorDistributionPolicy = "ByIndividualMoviesPostOrder"
    MirrorDistributionByLevels = "No"
    BuildingMirrorStep@Step = "Yes"
    MoviesToMap@EveryStep = "1"
    MirrorMappingPolicy = "ByMoviesDistance"
    BuildMirrorRandomly="No" >
    <SegmentedNetwork Id="**" Bandwidth="1000s" ResourceManager="Slot" >
    </SegmentedNetwork >
    <Proxy Id="**" Bandwidth="3000s" ResourceManager="Slot" ServicePolicy="Patching"
    CachingPolicy="FullMovieCaching" StorageCapacity="2025000s" MirrorCapaci
tyPercentage="50%" >
    <ProxyManagementPolicy >
    <FullMovieCachingPolicy ReplacementPolicy="NoReplacement" Replacement@Memb
er="*" >
    </FullMovieCachingPolicy >
    </ProxyManagementPolicy >
    </Proxy >
    </TreeArch >
  </Architecture >
  <Users UsersNumber="1000000" UsersServerDistribution="Uniform" >
  <UserRequestDistribution >
  <Distribution Type="Uniform" Begin="1" End="1000000" >
  </Distribution >
  </UserRequestDistribution >
  </Users >
  <MoviesSize >
  <Distribution Type="Constant" Mean="90" >
  </Distribution >
  </MoviesSize >
  </System >
  <BehaviourModel >
  <RequestInterval >
  <Distribution Type="Poisson" Mean="10" >
  </Distribution >
  </RequestInterval >
  <RequestMovie >
  <Distribution Type="Zipf" Begin="1" End="100" Mean="0.729" >
  </Distribution >
  </RequestMovie >
  </BehaviourModel >
</VoDSimulation >

```

A.2.3 Fichero de configuración para una simulación basada en un sistema Double P-Tree

```

<?xml version="1.0"?>
<!DOCTYPE VoDSimulation SYSTEM "VoDSimulator.dtd">
<VoDSimulation TestName="Double P-Tree With Collaborative Caching ConstRep2 Step Storage Dis
tribution Mirr12a20 Coll17a0 ConstantReplication=2 Unicast Saturation 3B 3SP LR63 Seg 100Mb S
204SR_BP5D Poisson10 BalTra ByDistance PostOrder Step"
  WriteOutput="No" PrintEvents="MainEvents" OutputMode="Short" RandomEvents="No"
  *
  OutputFile="..Pruebas/Estadisticas/DoublePTree3B3SPCollaborativeCachingConst
Rep2StorageDistributionMirr12_20Coll17_0UnicastMaxBandLR63Seg100mS204SR_BP5DPoisson10_BalTra
ByDistancePostOrderStep.out"
  StatisticsFile="..Pruebas/Estadisticas/DoublePTree3B3SPCollaborativeCachingC
onstRep2StorageDistributionMirr12_20Coll17_0UnicastMaxBandLR63Seg100mS204SR_BP5DPoisson10_Ba
lTraByDistancePostOrderStep.st1" >
  <SysTest StatisticsFile="..Pruebas/Estadisticas/DoublePTree3B3SPCollaborativeCachingCons
tRep2StorageDistributionMirr12_20Coll17_0UnicastMaxBandLR63Seg100mS204SR_BP5DPoisson10_BalTra
ByDistancePostOrderStep.dat"
    LaunchSeveralExecutions="Yes" NumberOfExecutions="10" StorePartialResults="No" >
  <StorageDistribution MirrMirrorringStorageCapacity="972000bs" MaxMirrorringStorageCapacit
y="162Gs"
    InMirrorringStorageCapacity="81000bs"
    MinCollaborativeCachingStorageCapacity="00bs"
    MaxCollaborativeCachingStorageCapacity="567000bs"
    InCollaborativeCachingStorageCapacity="81000bs" >
  </StorageDistribution>
  <StatisticsFields All="1" >
  </StatisticsFields>
  </SysTest>
  <EndCondition>
  <ByMaxSystemBandwidth MaxSystemBandwidth="1000bs" MultiplyBySystemNetworks="Yes" >
  </ByMaxSystemBandwidth>
  </EndCondition>
  <System MoviesNumber="100">
  <Architecture Type="DoublePTree" UseSegmentedNetworks="Yes" NetworkPorts="24" ServerP
orts="3"
    TrafficBalancing="Yes" >
  <DoublePTreeArch TreeOrder="2" LocalNetworks="63" WithMainServer="No" InitCache="Ye
s"
    BrothersNetworks="3" BestCacheMirrorDistribution="No"
    MirrorDistributionPolicy = "ByIndividualMoviesPostOrder"
    MirrorDistributionLevels = "No"
    BuildingMirrorStepsStep = "Yes"
    MoviesToRepEveryStep = "1"
    InitCollaborativeCache="Yes"
    MirrorMappingPolicy = "ByMoviesDistance"
    BuildMirrorRandomly="Yes"
    StorageInitializationStepByStep="Yes" >
  <SegmentedNetwork Id="**" Bandwidth="2000bs" ResourceManager="Slot" >
  </SegmentedNetwork>
  <Proxy Id="**" Bandwidth="4000bs" ResourceManager="Slot" ServicePolicy="Unicast"
    CachingPolicy="FullMovieCollaborativeCachingPolicy" StorageCapacity="162G
bs" MirrorCapacity="124000bs" >
  <ProxyManagementPolicy>
  <FullMovieCollaborativeCachingPolicy>
  <FullMovieCachingPolicy ReplacementPolicy="NoReplacement" ReplacementUm
bral="2" >
  </FullMovieCachingPolicy>
  <CollaborativeCachingPolicy CollaborativeCachingCapacity="0"
    CachingMovieSelectionPolicy="ByMovieReplicatio
n"
    FillDistributedMirrorCaps="No"
    ReplacementPolicy="NoReplacement"
    MovieReplicationPolicy="ConstantReplication"
    ReplacementUbral="5" MoviesReplication="2"
    StatisticsRefreshPeriod="60" >
  </CollaborativeCachingPolicy>
  </FullMovieCollaborativeCachingPolicy>
  </ProxyManagementPolicy>
  </Proxy>
  </DoublePTreeArch>
  </Architecture>

```

```
<Users UsersNumber="1000000" UsersServerDistribution="Uniform" BufferSize="10">
  <UserRequestDistribution>
    <Distribution Type="Uniform" Begin="1" End="1000000" >
      </Distribution>
    </UserRequestDistribution>
  </Users>
  <MoviesSize>
    <Distribution Type="Constant" Mean="90" >
      </Distribution>
    </MoviesSize>
  </System>
  <BehaviourModel>
    <RequestInterval>
      <Distribution Type="Poisson" Mean="10" >
        </Distribution>
      </RequestInterval>
    <RequestMovie>
      <Distribution Type="Zipf" Begin="1" End="100" Mean="0.729" >
        </Distribution>
      </RequestMovie>
    </BehaviourModel>
  </VocSimulation>
```