

Referências bibliográficas

ABNT NBR 15603-2, **Televisão digital terrestre – Multiplexação e serviços de informação (SI) Parte 2: Estrutura de dados e definições da informação básica de SI**. Versão Corrigida. São Paulo. Brasil. 2007. ABNT, 2007

ABNT NBR 15604, **Televisão digital terrestre – Receptores**. Versão Corrigida. São Paulo. Brasil. 2008. ABNT, 2008a

ABNT NBR 15606-2. **Digital Terrestrial Television Standard 06: Data Codification and Transmission Specifications for Digital Broadcasting, Part 2 – GINGA-NCL: XML Application Language for Application Coding**. Corrected Version 2. São Paulo, Brazil. 2009. ABNT, 2009

ABNT NBR 15606-3. **Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 3: Especificação de transmissão de dados**. Versão Corrigida 2. São Paulo, Brasil. 2008. ABNT, 2008b

ARIB Standard. ARIB STD-B23, **Application Execution Engine Platform for Digital Broadcasting**, 2004. ARIB, 2004a

ARIB Standard. ARIB STD-B24, Version 4.0: **Data Coding and Transmission Specification for Digital Broadcasting**, 2004. ARIB, 2004b

ARIB Standard. 3GPP TS 26.346, Version 8.4.0: **Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs**, 2009. ARIB, 2009

ATSC Standard. **Advanced Application Platform (ACAP)**, 2009, Document A/101. ATSC, 2009

ATSC Standard. **ATSC Data Broadcasting Standard**, 2000, Document A/90. ATSC, 2000

BARBOSA, N. PERKUSICH, A. **Estudo Empírico Comparativo de Modelos de Componentes para o Desenvolvimento de Software com Suporte à Evolução Dinâmica e Não Antecipada**. Anais do Workshop de Teses em Engenharia de Software (WTES 2006). 2006. Barbosa, 2006

BRUNETON, E. et al. **The FRACTAL Component Model and Its Support in Java**. Software, Practice and Experience. 36(11-12): 1257-1284. 2006. Bruneton, 2006

CABLELABS Standard. OpenCable Application Platform (OCAP), **OpenCable Application Platform Specification, 2005**. CableLabs, 2005

CASTRO, M.; LISKOV, B. **Practical Byzantine fault-tolerance and proactive recovery**. ACM TOCS, 20(4):398–461, 2002. Castro, 2002

COSTA, Romualdo; MORENO, Marcelo; SOARES, Luiz F. **Intermedia Synchronization Management in DTV Systems**. Proceedings of the ACM Symposium on Document Engineering. São Paulo, Brasil. 2006. ISBN: 1-59593-515-0. Costa, 2006a

COSTA, Romualdo; MORENO, Marcio; RODRIGUES, Rogério; SOARES, Luiz F. **Live Editing of Hypermedia Documents**. In Proceedings of the 2006 ACM Symposium on Document Engineering (Amsterdam, The Netherlands, October 10 - 13, 2006). DocEng '06. ACM, New York, NY, 165-172. DOI=<http://doi.acm.org/10.1145/1166160.1166202> - DocEng'06, 2006, Amsterdam. Proceedings of the ACM Symposium on Document Engineering - DocEng'06, 2006. p. 165-172. Grande área: Ciências Exatas e da Terra / Área: Ciência da Computação / Subárea: Sistemas de Computação / Especialidade: Sistemas Hiperímídia. ISSN/ISBN: 1595935150 Costa, 2006b

COULSON G. et al. **A Generic Component Model for Building Systems Software**. ACM Transaction on Computer Systems (TOCS), Volume 26, Issue 1, 2008. Coulson, 2008

COX, A. **Video4Linux Programming**, 2000. Disponível em <http://kernelbook.sourceforge.net/videobook.html/>. Acesso em: 07 jun. 2010. Cox, 2000

ETSI Standard. **Digital Video Broadcasting (DVB), Implementation guidelines for Data Broadcasting 1.2.1**, 2003, ETSI TR 101 202. ETSI, 2003

ETSI Standard. **Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2**, 2010, ETSI TS 102 727. ETSI, 2010

ETSI Standard. **IP Datacast over DVB-H: Content Delivery Protocols (CDP)**, A101r1, dTS 102 472 v1.3.1, 2009. ETSI, 2009

FILHO S. M. et al. **FLEXCM – A Component Model for Adaptive Embedded Systems**. Proceedings of the 31st Annual International Computer Software and Applications Conference – Volume 01. Pages: 119-126. 2007. Filho, 2007

FUGINI, M.; MUSSI, E. **Recovery of Faulty Web Applications through Service Discovery**. 1st International Workshop on Semantic Matchmaking and Resource Retrieval: Issues and Perspectives (SMR 2006). Seoul, Korea. September 11, 2006. Fugini, 2006

GARG, S. et al. **A methodology for detection and estimation of software aging**. In International Symposium on Software Reliability Engineering, pages 283–292, November 1998. Garg, 1998

GOMES A. T. A. **LindaX: Uma Linguagem de Descrição de Sistemas de Comunicação Adaptáveis**. Tese de Doutorado. Departamento de Informática. PUC-Rio. Março de 2005. Gomes, 2005

HUANG, Y. et al. **Software rejuvenation: Analysis, module and applications**. In International Symposium on Fault-Tolerant Computing, pages 381–390, June 1995. Huang, 1995

IERUSALIMSCHY, Roberto. **Programming in Lua**, second edition, Lua.org, 2006. Ierusalimschy, 2006

ISO/IEC 13818-1. **Information technology – Generic coding of moving pictures and associated audio information – Part 1: Systems**. ISO Standard, 2007. ISO, 2007

ISO/IEC 13818-6. **Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC**. ISO Standard, 1998. ISO, 1998

ITU-R – International Telecommunications Union – Radiocommunication. Recommendation BT 1699-1, 2009. **Harmonization of declarative application formats for interactive TV**. Geneva, January, 2009. ITU-R, 2009

ITU-T – International Telecommunications Union – Telecommunications. ITU-T Recommendation H.761, 2009. **Nested Context Language (NCL) and GINGA-NCL for IPTV**. Geneva, April, 2009. Disponível em: < <http://www.itu.int/itu-t/aap/AAPRecDetails.aspx?AAPSeqNo=1894>>. Acesso em: 07 jun. 2010. ITU-T, 2009

KOREN, I.; KRISHNA, C. **Fault-tolerant Systems**. Morgan Kaufmann, 2007. Koren, 2007

LAYAÏDA, O.; HAGIMONT, D. **Designing Self-Adaptive Multimedia Applications through Hierarchical Reconfiguration**, 5th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), Athens, Greece, June 2005. Layaïda, 2005

MARANHAO, Suzana; RODRIGUES, Rogério; SOARES, Luiz F. **On-the-Fly Time Scaling for Compressed Audio Streams**. International Conference on Signal Processing and Multimedia Applications. SIGMAP. 2007. Maranhao, 2006

MARSH, N. **Introduction to the Command Line: A Beginner's Guide to Unix and Linux Commands**. CreateSpace. 204 pages. 2009. Marsh, 2009

MEDINA, Vitor; MORENO, Marcio; SOARES, Luiz F. **GINGA-NCL: Implementação de Referência para Dispositivos Portáteis**. In: XIV Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2008, Vila Velha. Anais do XIV Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2008. p. 67-74. Medina, 2008

MORENO, Marcelo et al. **GINGA Live CD: Presentation and Testing Environment for Interactive Multimedia Content**, Relatório Técnico, Laboratório TeleMídia, Departamento de Informática, PUC-Rio, 2009. Moreno, 2009a

MORENO, Marcio. **GINGA-NCL: Relating Imperative, Declarative and Media Objects**. EuroITV 2009. Doctoral Consortium of European Conference on Interactive Television. Leuven, Belgium June 03-05, 2009. **EuroITV Best Phd Award**. Moreno, 2009b

MORENO, Marcio. **Um Middleware Declarativo para Sistemas de TV Digital Interativa**, Dissertação de Mestrado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Abril de 2006. Moreno, 2006

MORENO, Marcio; COSTA, Romualdo; SOARES, Luiz F. **Sincronismo entre Fluxos de Mídia Contínua e Aplicações Multimídia em Redes por Difusão**. In: XIV Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2008, Vila Velha.

Anais do XIV Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2008. p. 202-209. Moreno, 2008

MORENO, Marcio; MORENO, Marcelo; SOARES, Luiz F. **Transmissão de Aplicações e Comandos de Edição ao Vivo em IPTV e DTV Terrestre**. In: XVI Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2010, Belo Horizonte. Anais do XVI Simpósio Brasileiro de Sistemas Multimídia e Hiperemídia, 2010. Moreno, 2010a

MORENO, Marcio; RODRIGUES, Rogério; SOARES, Luiz F. **A Resource Identification Mechanism for Interactive DTV Systems**. In Proceedings of the Ninth IEEE international Symposium on Multimedia Workshops (December 10 - 12, 2007). IEEE Computer Society, Washington, DC, 215-220. DOI=<http://dx.doi.org/10.1109/ISMW.2007.6> Workshop on New Techniques for Consuming, Managing, and Manipulating Interactive Digital Media at Home (ISM 2007) - CMMIDMH, p. 215-220 Meio de divulgação: Impresso; ISSN/ISBN: 0-7695-3084-2 and 9780-7695-3084-0 Digital Object Identifier: 10.1109/ISM.Workshops.2007.44 Moreno, 2007

MORENO, Marcio; SOARES, Luiz F.; CERQUEIRA, Renato. **Introduzindo Resiliência à Apresentação de Aplicações Interativas Ginga-NCL**, Relatório Técnico, Laboratório TeleMídia, Departamento de Informática, PUC-Rio, 2010. Moreno, 2010b

MORENO, Marcio; SOARES, Luiz F.; CERQUEIRA, Renato. **Uma Arquitetura Orientada a Componentes para o Middleware Ginga-NCL**, Relatório Técnico, Laboratório TeleMídia, Departamento de Informática, PUC-Rio, 2010. Moreno, 2010c

MORENO, Marcio; SOARES NETO, Carlos; SOARES, Luiz F. **Adaptable Software Components in an Electronic Program/Service Guide Application Architecture for Context Aware Guide Presentation**. In: International Journal of Advanced Media and Communications. Vol. 3 No. 4. 2009. p. 351-364. Moreno, 2009c

MORRIS, S., SMITH-CHAIGNEAU, A. **Interactive TV Standards**, 2005, Elsevier Inc. Morris, 2005

Presidência da República, Casa Civil. **Decreto Nº 5.820, de 29 de Junho de 2006**. Disponível em http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Decreto/D5820.htm. Acesso em Junho de 2010. PRCC, 2010

OMG Specification (2004). **Common Object Request Broker Architecture (CORBA/IIOP)**, 2004. OMG, 2004

RAMDHANY, R. et al. **MANETKit: supporting the dynamic deployment and reconfiguration of ad-hoc routing protocols**. Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware. Urbana, Illinois. 2009. Ramdhany, 2009

RFC 3550, Request for Comments: 3550. Standard 64, **RTP: A Transport Protocol for Real-Time Applications**, 2003. RFC, 2003

RFC 3926, Request for Comments: 3926. **FLUTE: File Delivery over Unidirectional Transport**, 2004. RFC, 2004

RFC 3986. Request for Comments: 3986. **Uniform Resource Identifier (URI): Generic Syntax**, 2005. RFC, 2005

RODRIGUES, Rogério. **Formatação e Controle de Apresentações Hiperímídia com Mecanismos de Adaptação Temporal**, Tese de Doutorado, Departamento de Informática, PUC-Rio, Rio de Janeiro, Março de 2003. Rodrigues, 2003

SOARES, Luiz F.; BARBOSA, Simone. **Programando em NCL**, 1ª Edição, Editora Elsevier Campos, 483p. 2009. Soares, 2009a

SOARES, Luiz F.; COSTA, Romualdo; MORENO, Marcio; MORENO, Marcelo. **Multiple Exhibition Devices in DTV Systems**. In: ACM International Conference on Multimedia, 2009, Beijing. Proceedings of the ACM International Conference on Multimedia, 2009. p. 281-289. Soares, 2009b

SOARES, Luiz F.; MORENO, Marcio; SOARES NETO, Carlos; MORENO, Marcelo. **Ginga-NCL: Declarative Middleware for Multimedia IPTV Services**. In: IEEE Communications Magazine. Vol.48, No.6. June 2010. Soares, 2010

SOARES, Luiz F.; RODRIGUES, Rogério. **Nested Context Language 3.0: Part 8 – NCL (Nested Context Language) Digital TV Profiles**. Monografias em Ciência da Computação do Departamento de Informática, PUC-Rio, No. 35/06. Rio de Janeiro. Outubro de 2006. ISSN 0103-9741. Disponível em: <ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/mccs/2006_10_dtvprofiles.pdf>. Acesso em: 07 jun. 2010. Soares, 2006

SOARES, Luiz F.; RODRIGUES, Rogério. **Nested Context Model 3.0: Part 1 – NCM Core**. Relatório Técnico, Laboratório TeleMídia, PUC-Rio, Brasil, 2005. Soares, 2005

SOARES, Luiz F.; RODRIGUES, Rogério; MORENO, Marcio. **Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System**. Journal of the Brazilian Computer Society, v. 12, p. 37-46, 2007. Soares, 2007

SOUSA, P. **Resilient Intrusion Tolerance Through Proactive and Reactive Recovery**. The 13th Pacific Rim International Symposium on Dependable Computing. Proceedings of PRDC'07. IEEE Computer Press. Melbourne, Australia. Dezembro de 2007. Souza, 2007

SIMPSON, Wes. **Video Over IP, Second Edition: IPTV, Internet Video, H.264, P2P, Web TV, and Streaming: A Complete Guide to Understanding the Technology**, Kindle Edition, 2008. Simpson, 2008

SZYPERSKI, C., GRUNTZ, D., MURER, S., **Component Software – Beyond Object-Oriented Programming**. Second edition. ACM Press, 2002. Szyperski, 2002

YUJUAN, B.; XIAOBAL, S.; TRIVEDI, K.S. **Adaptive software rejuvenation: Degradation model and rejuvenation scheme**. In International Conference on Dependable Systems and Networks, pages 241–248, June 2003. Yujuan, 2003

Anexo I – Conceitos MPEG-2

Este anexo apresenta os principais conceitos, utilizados nos sistemas de TVD, especificados no padrão MPEG-2 Sistemas (ISO, 2007).

Fluxo de Transporte MPEG-2

As especificações do fluxo de transporte fazem parte do padrão MPEG-2 Systems (ISO, 2007), ou MPEG-2 Sistemas, e estabelece como um ou mais sinais de áudio e vídeo, assim como outros dados (imagens estáticas, textos etc.), devem ser combinados de forma a serem transmitidos. Um fluxo de transporte é formado por um ou mais fluxos elementares (ES – *Elementary Stream*). Um fluxo elementar é definido como um fluxo de dados gerado pela codificação do conteúdo de vídeo, áudio ou outros dados.

As especificações MPEG-2 Sistemas definem ainda o termo programa, chamado de serviço no contexto da TV digital, como um grupo composto de um ou mais fluxos elementares com uma mesma base temporal (ISO, 2007). O fluxo de transporte pode conter vários serviços (programas) simultaneamente, cada um podendo ter uma base de tempo diferente.

Simplificadamente, multiplexar serviços em um fluxo de transporte significa organizar os pacotes dos vários fluxos elementares, pertencentes aos serviços contemplados, em um único fluxo. Para isso, é necessário inserir no fluxo de transporte informações que permitam ao decodificador MPEG-2 identificar a qual serviço um dado fluxo elementar pertence. Essas informações são dispostas como um conjunto de tabelas de informação específica de programa (PSI – *Program Specific Information*). Uma PSI particular, denominada PMT (*Program Map Table*), contém a lista de identificadores dos fluxos elementares que compõem um serviço. Cada PMT encontrada representa um serviço disponível. As PMTs são localizadas através de outra PSI denominada PAT (*Program Association Table*), que contém identificadores dos fluxos elementares contendo as PMTs. O fluxo

elementar que possui a PAT possui identificador fixo com o valor hexadecimal 0x00.

Eventos de Sincronismo DSM-CC

Sincronizar o comportamento de uma aplicação com o conteúdo de uma programação de TV específica (áudio e vídeo principal) é extremamente desejável, principalmente quando a aplicação em questão possui relação semântica com essa programação. O sincronismo pode ser realizado através de selos de tempos (paradigma *timeline*), quando os dados das aplicações são enviados de forma síncrona, ou sincronizada (ISO, 2007) com o áudio e vídeo principal, ou alternativamente através de eventos de sincronismo DSM-CC (ou simplesmente eventos DSM-CC), quando os dados são transmitidos de forma assíncrona (sem o uso de selos de tempo, utilizando, por exemplo, o carrossel de objetos, a ser discutido na próxima seção).

Para criar um evento DSM-CC, é inserida uma estrutura no fluxo de transporte, denominada descritor de eventos. Cada descritor de eventos possui um identificador numérico único, que o identifica no fluxo de transporte. Uma vez que não existe a possibilidade do provedor de conteúdo saber exatamente a posição que o descritor será inserido no fluxo, cada descritor possui uma referência temporal que indica em qual instante o evento deverá ocorrer, usualmente baseado em um fluxo DSM-CC denominado NPT (*Normal Play Time*) (ISO, 1998). Como caso particular, um descritor de eventos pode informar ao sistema receptor que o evento deve ocorrer imediatamente – esse tipo de evento é chamado de evento “*do it now*”. Além do identificador e da referência temporal, o descritor de eventos possui também um campo para dados específicos das aplicações, que pode ser utilizado de acordo com sintaxes e semânticas a serem tratadas pelas próprias aplicações. Para evitar que um descritor de evento DSM-CC seja perdido devido a falhas no meio de difusão, um mesmo descritor normalmente é enviado diversas vezes pelo provedor de conteúdo, cabendo ao receptor interpretar esses eventos uma única vez.

Os descritores de eventos DSM-CC, bem como os outros dados DSM-CC (por exemplo, dados referenciados pelos eventos e que devem ser sincronizados com o áudio e vídeo principal), precisam de estruturas especiais para serem

transportados. Para tanto, as especificações do protocolo DSM-CC determinam estruturas de dados denominadas seções DSM-CC. Essas estruturas possuem um cabeçalho, especificado com o objetivo de informar ao decodificador como as seções estão sendo utilizadas para transportar os dados, como elas devem ser remontadas, quais os tipos de dados transportados, além de outros parâmetros para o tratamento apropriado da informação (ISO, 1998).

Carrossel de Objetos DSM-CC

O carrossel de objetos é um protocolo de transmissão cíclica de dados. Os dados são representados por objetos (objeto de diretório, objeto de arquivo etc. (ISO, 1998)), que contêm atributos (nome, tipo e, possivelmente, conteúdo). Segundo as especificações DSM-CC, que são compatíveis com o framework ORB (*Object Request Broker*) definido pelas especificações CORBA (*Common Object Request Broker Architecture*) (OMG, 2004), cada objeto deve ser encapsulado em uma mensagem BIOP (*Broadcast Inter ORB Protocol*), que são transmitidas em módulos. Uma mensagem BIOP deve ser transmitida em um único módulo, mas um módulo pode conter mais de uma mensagem. Os módulos, por sua vez, são divididos em blocos de dados, que são encapsulados em mensagens denominadas DDB ou *DownloadDataBlock* (ISO, 1998). Os blocos de dados, por sua vez, são encapsulados em seções DSM-CC. As regras de encapsulamento de blocos de dados em seções foram especificadas para que blocos possam ser adquiridos diretamente do fluxo de transporte através de filtros (ETSI, 2003).

Cada seção DSM-CC de um carrossel é transmitida no fluxo de transporte, como um fluxo elementar de dados, uma após a outra. Depois de transmitir a última seção, a transmissão é reiniciada. O resultado disso é um fluxo elementar que contém o sistema de arquivos transmitido de forma cíclica. Assim, se um determinado dispositivo de recepção não recebeu um bloco de dados em particular (devido a uma falha na transmissão ou por ter sintonizado o canal após a transmissão desse bloco), basta esperar pela retransmissão correta da seção contendo esse bloco de dados.

Cada instância de carrossel de objetos é representada por um *Service Domain* (ISO, 1998), que consiste em uma identificação única do carrossel de objetos. Todo *Service Domain* possui um *Service Gateway* (ISO, 1998), que

contém referências para todos os objetos dispostos na raiz do carrossel de objetos. O padrão DSM-CC utiliza a estrutura de referências IOR (*Interoperable Object Reference*), definidas nas especificações CORBA. No contexto do protocolo carrossel de objetos, uma IOR é normalmente composta pelo identificador do carrossel, seguido do identificador do módulo e pelo identificador do objeto. O *Service Gateway* é um objeto do carrossel de objetos cuja localização é transmitida em uma mensagem *DownloadServerInitiate* (DSI) (ISO, 1998), que pode ser encontrada através das informações PSI.

Como exemplo, considere o sistema de arquivos apresentado na Figura 14. Considere ainda um carrossel de objetos gerado a partir desse sistema de arquivos, com o *Service Domain* de valor hexadecimal “1”, conforme apresentado também na Figura 14.

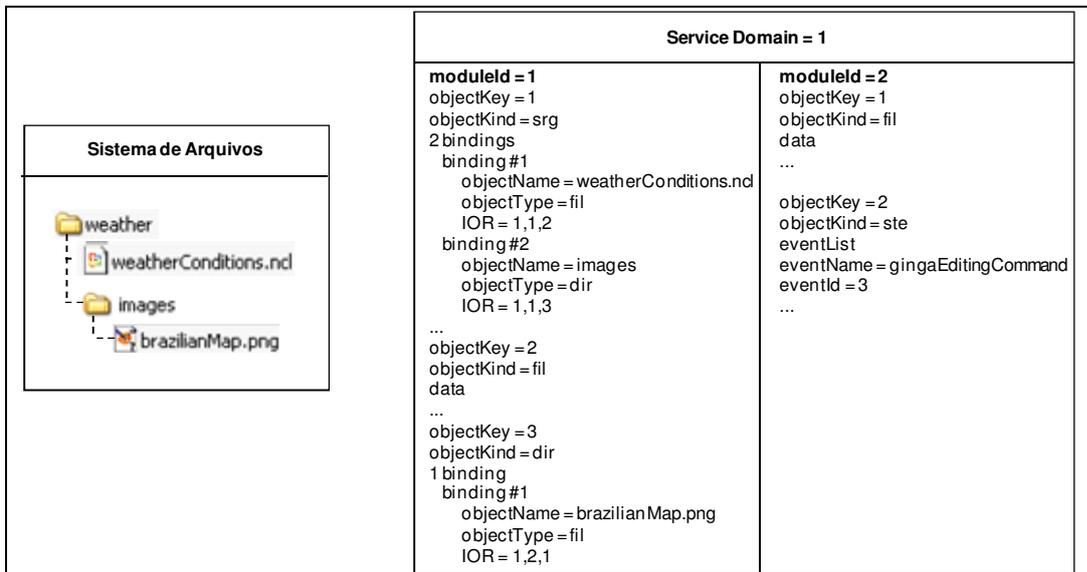


Figura 14 – Exemplo de Carrossel de Objetos gerado a partir de um sistema de arquivos.

Nesse *Service Domain*, dois módulos são gerados e identificados com valores em hexadecimal “1” e “2”. O identificador de cada objeto é apresentado através do campo “objectKey”. O objeto que representa o *Service Gateway* (na Figura 14, objeto do tipo “srg”) é identificado com o valor “1” e é encapsulado no módulo “1”. Como consequência, a IOR do objeto *Service Gateway* transmitida na mensagem *DownloadServerInitiate* é definida por “1,1,1” (*Service Domain* = 1, id do módulo = 1 e id do objeto = 1). Os objetos que representam o arquivo “weatherConditions.ncl” e o diretório “images” são identificados pelos valores hexadecimais “2” e “3”, respectivamente, e também são encapsulados no módulo “1”. Já o objeto que representa o arquivo contendo o conteúdo da imagem é

identificado com o valor “1”, mas encapsulado no módulo “2”. O objeto *Service Gateway* possui, por sua vez, duas IORs. Para relacionar uma IOR ao nome do objeto que a mesma referencia, bem como ao tipo desse objeto (arquivo, diretório etc.), as especificações DSM-CC utilizam o conceito de *binding*. Assim, no exemplo, o objeto *Service Gateway* possui dois *bindings* para os dois objetos do módulo “1” que são filhos diretos da raiz do sistema de arquivos representado pelo carrossel.

Os objetos do tipo diretório (“dir”) possuem a mesma sintaxe e semântica dos objetos do tipo *Service Gateway*. A diferença é que um objeto *Service Gateway* apenas representa o diretório raiz do *Service Domain*. Os objetos do tipo arquivo (“fil”) possuem como atributo, além da identificação do seu tipo, os dados relativos ao seu conteúdo.

A Figura 14 apresenta também um objeto do tipo evento de fluxo (“ste”). Esse objeto é utilizado para definir tipos de eventos DSM-CC possíveis de serem descritos no fluxo de transporte. Para isso, o objeto relaciona identificadores de descritores de eventos DSM-CC a uma string (na figura, conteúdo do campo “eventName”).

Finalmente, é importante notar, no exemplo da Figura 14, que a identificação da raiz do sistema de arquivos (diretório “weather”) é perdida, quando da geração do carrossel.

Anexo II – Comandos de Edição para Entidades NCL

Comando	tag	Descrição
addRegion (baseld, documentId, regionBaseld, regionId, xmlRegion)	0x0B	Adiciona um elemento <region> como filho de outro <region> no <regionBase>, ou como filho do <regionBase> (regionId="null") de um documento NCL em uma base privada aberta.
removeRegion (baseld, documentId, regionId)	0x0C	Remove um elemento <region> de um <regionBase> de um documento NCL em uma base privada aberta.
addRegionBase (baseld, documentId, xmlRegionBase)	0x0D	Adiciona um elemento <regionBase> ao elemento <head> de um documento NCL em uma base privada aberta. Se a especificação XML do regionBase for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlRegionBase é apenas uma referência para esse conteúdo.
removeRegionBase (baseld, documentId, regionBaseld)	0x0E	Remove um elemento <regionBase> do elemento <head> de um documento NCL em uma base privada aberta.
addRule (baseld, documentId, xmlRule)	0x0F	Adiciona um elemento <rule> ao <ruleBase> de um documento NCL em uma base privada aberta.
removeRule (baseld, documentId, ruleId)	0x10	Remove um elemento <rule> do <ruleBase> de um documento NCL em uma base privada aberta.
addRuleBase (baseld, documentId, xmlRuleBase)	0x11	Adiciona um elemento <ruleBase> ao elemento <head> de um documento NCL em uma base privada aberta. Se a especificação XML do ruleBase for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlRuleBase é apenas uma referência para esse conteúdo.
removeRuleBase (baseld, documentId, ruleBaseld)	0x12	Remove um elemento <ruleBase> do elemento <head> de um documento NCL em uma base privada aberta.
addConnector (baseld, documentId, xmlConnector)	0x13	Adiciona um elemento <connector> ao <connectorBase> de um documento NCL em uma base privada aberta.
removeConnector (baseld, documentId, connectorId)	0x14	Remove um elemento <connector> do <connectorBase> de um documento NCL em uma base privada aberta.
addConnectorBase (baseld, documentId, xmlConnectorBase)	0x15	Adiciona um elemento <connectorBase> ao elemento <head> de um documento NCL em uma base privada aberta. Se a especificação XML do connectorBase for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlConnectorBase é apenas uma referência para esse conteúdo.
removeConnectorBase (baseld, documentId, connectorBaseld)	0x16	Remove um elemento <connectorBase> do elemento <head> de um documento NCL em uma base privada aberta.
addDescriptor (baseld, documentId, xmlDescriptor)	0x17	Adiciona um elemento <descriptor> ao <descriptorBase> de um documento NCL em uma base privada aberta.

documentId, xmlDescriptor)		documento NCL em uma base privada aberta.
removeDescriptor (baseId, documentId, descriptorId)	0x18	Remove um elemento <descriptor> do <descriptorBase> de um documento NCL em uma base privada aberta.
addDescriptorSwitch (baseId, documentId, xmlDescriptorSwitch)	0x19	Adiciona um elemento <descriptorSwitch> ao <descriptorBase> de um documento NCL em uma base privada aberta. Se a especificação XML do descriptorSwitch for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlDescriptorSwitch é apenas uma referência para esse conteúdo.
removeDescriptorSwitch (baseId, documentId, descriptorSwitchId)	0x1A	Remove um elemento <descriptorSwitch> do <descriptorBase> de um documento NCL em uma base privada aberta.
addDescriptorBase (baseId, documentId, xmlDescriptorBase)	0x1B	Adiciona um elemento <descriptorBase> ao elemento <head> de um documento NCL em uma base privada aberta. Se a especificação XML do descriptorBase for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlDescriptorBase é apenas uma referência para esse conteúdo.
removeDescriptorBase (baseId, documentId, descriptorBaseId)	0x1C	Remove um elemento <descriptorBase> do elemento <head> de um documento NCL em uma base privada aberta.
addTransition (baseId, documentId, xmlTransition)	0x1D	Adiciona um elemento <transition> ao <transitionBase> de um documento NCL em uma base privada aberta.
removeTransition (baseId, documentId, transitionId)	0x1E	Remove um elemento <transition> do <transitionBase> de um documento NCL em uma base privada aberta.
addTransitionBase (baseId, documentId, xmlTransitionBase)	0x1F	Adiciona um elemento <transitionBase> ao elemento <head> de um documento NCL em uma base privada aberta. Se a especificação XML do transitionBase for enviada em um sistema de transporte como um sistema de arquivo, o parâmetro xmlTransitionBase é apenas uma referência para esse conteúdo.
removeTransitionBase (baseId, documentId, transitionBaseId)	0x20	Remove um elemento <transitionBase> do elemento <head> de um documento NCL em uma base privada aberta.
addImportBase (baseId, documentId, docBaseId, xmlImportBase)	0x21	Adiciona um elemento <importBase> à base (elemento <regionBase>, <descriptorBase>, <ruleBase>, <transitionBase>, or <connectorBase>) de um documento NCL em uma base privada aberta.
removeImportBase (baseId, documentId, docBaseId, documentURI)	0x22	Remove um elemento <importBase>, cujo atributo documentURI é identificado pelo parâmetro documentURI, a partir da base (elemento <regionBase>, <descriptorBase>, <ruleBase>, <transitionBase>, or <connectorBase>) de um documento NCL em uma base privada aberta.
addImportedDocumentBa se (baseId, documentId, xmlImportedDocumentBa se)	0x23	Adiciona um elemento <importedDocumentBase> ao elemento <head> de um documento NCL em uma base privada aberta.
removeImportedDocumen tBase (baseId, documentId, importedDocumentBaseId)	0x24	Remove um elemento <importedDocumentBase> do elemento <head> de um documento NCL em uma base privada aberta.

)		
addImportNCL (baselId, documentId, xmlImportNCL)	0x25	Adiciona um elemento <importNCL> ao elemento <importedDocumentBase> de um documento NCL em uma base privada aberta.
removeImportNCL (baselId, documentId, documentURI)	0x26	Remove um elemento <importNCL>, cujo atributo documentURI é identificado pelo parâmetro documentURI, a partir do <importedDocumentBase> de um documento NCL em uma base privada aberta.
addNode (baselId, documentId, compositelId, {uri, id}+)	0x27	Adiciona um nó (elemento <media>, <context> ou <switch>) a um nó de composição (elemento <body>, <context> ou <switch> identificado por compositelId) de um documento NCL em uma base privada aberta. A especificação XML do nó e seu conteúdo de mídia podem: a) ser enviados sem solicitação pela rede de difusão de dados; nesse caso, o par {uri, id} é usado para relacionar um conjunto de caminhos de arquivos, definidos no documento XML da especificação do nó, com suas respectivas localizações no sistema de transporte; b) recebidos pelo canal de interatividade sob demanda, ou já ser residentes no receptor; para esses arquivos, nenhum par {uri, id} necessita ser enviado, exceto o par {uri, "null"} associado à especificação XML do nó NCL que deverá ser adicionado em compositelId, caso o documento XML não seja recebido sem solicitação (pushed file).
removeNode(baselId, documentId, compositelId, nodelId)	0x28	Remove um nó (elemento <media>, <context> ou <switch>) de um nó de composição (elemento <body>, <context> ou <switch>) de um documento NCL em uma base privada aberta.
addInterface (baselId, documentId, nodelId, xmlInterface)	0x29	Adiciona uma interface (<port>, <area>, <property> ou <switchPort>) a um nó (elemento <media>, <body>, <context> ou <switch>) de um documento NCL em uma base privada aberta.
removeInterface (baselId, documentId, nodelId, interfacelId)	0x2A	Remove uma interface (<port>, <area>, <property> ou <switchPort>) de um nó (elemento <media>, <body>, <context> ou <switch>) de um documento NCL em uma base privada aberta. A interfacelId deve obrigatoriamente identificar um atributo name de um elemento <property> ou um atributo id de um elemento <port>, <area> ou <switchPort>.
addLink (baselId, documentId, compositelId, xmlLink)	0x2B	Adiciona um elemento <link> a um nó de composição (elemento <body>, <context> ou <switch>) de um documento NCL em uma base privada aberta.
removeLink (baselId, documentId, compositelId, linkId)	0x2C	Remove um elemento <link> de um nó de composição (elemento <body>, <context> ou <switch>) de um documento NCL em uma base privada aberta.
setProperty(baselId, documentId, nodelId, propertyId, value)	0x2D	Atribui o valor a uma propriedade. A propertyId deve obrigatoriamente identificar um atributo name de um elemento <property> ou um atributo id de elemento <switchPort>. O <property> ou <switchPort> deve obrigatoriamente pertencer a um nó (elemento <body>, <context>, <switch> ou <media>) de um documento NCL em uma base privada aberta identificada pelos parâmetros.

Tabela 15 – Comandos de edição de Entidades NCL.

Anexo III – Avaliação da Arquitetura por Medições

Este anexo primeiro discute sobre as aplicações NCL utilizadas na avaliação das duas configurações discutidas no Capítulo 3 (MONO e COMP). Uma vez discutidas as aplicações, são apresentados os cálculos para determinar o tamanho da amostra necessário para a obtenção de um grau de confiança de 95% na realização das medições. Finalmente, as medições realizadas são apresentadas.

Como discutido no Capítulo 3, uma aplicação NCL pode conter objetos de mídia de diferentes tipos, como texto, imagem, vídeo, áudio e aplicações (objeto XHTML ou NCLua). Neste anexo, por hipótese, o uso de determinados tipos de objetos de mídia, especificados nas aplicações, varia de acordo com o gênero do programa de TV que a aplicação é semanticamente relacionada.

A Tabela 16 apresenta, na coluna “Objetos”, os objetos de mídia comumente especificados, por hipótese, nas aplicações que possuem relacionamento semântico com o programa de TV de gênero apresentado na coluna “Classificação”. Os gêneros são listados de acordo com a classificação definida no padrão brasileiro (ABNT, 2007).

Por hipótese, nas aplicações, os objetos de imagem e texto são sincronizados com o áudio principal ou o vídeo principal para enriquecer o programa de TV. Objetos NCLua são utilizados para a realização de enquetes e pesquisas a serem respondidas pelos telespectadores. As respostas são enviadas às emissoras por meio de um canal de interatividade. Objetos XHTML oferecem ao telespectador a opção de informações adicionais sobre o programa e compra de produtos, bem como a navegação na Web utilizando um canal de interatividade. Objetos de áudio são utilizados para oferecer opções ao programa como, por exemplo, narrativas em programas educativos e músicas em programas musicais.

Assim como nas medições apresentadas na Seção 3.3, objetos de vídeo não são considerados neste anexo, uma vez que, em plataformas de receptores reais, tais objetos são decodificados por hardware. Além disso, a decodificação desses objetos por software demanda tanta memória e CPU que prejudicaria a escala dos

gráficos de comparação e tornaria difícil observar as mudanças na utilização dos recursos nos outros casos.

Classificação	Objetos				
	Imagem	Texto	NCLua	XHTML	Áudio
Jornalismo	✓	✓	✓	✓	
Esporte	✓	✓	✓	✓	
Educativo	✓	✓	✓	✓	✓
Novela, Minissérie e Série/Seriado	✓	✓	✓	✓	
Auditório	✓	✓	✓	✓	
Show e Musical	✓	✓	✓	✓	✓
<i>Making of</i>	✓	✓			
Feminino	✓	✓	✓	✓	
Game Show	✓	✓	✓	✓	
Reality show	✓	✓	✓	✓	
Culinária	✓	✓			
Moda	✓	✓			
Rural	✓	✓			
Saúde	✓	✓	✓	✓	
Turismo	✓	✓	✓	✓	
Humorístico	✓	✓	✓	✓	
Infantil	✓	✓	✓	✓	✓
Erótico	✓	✓	✓	✓	
Filme	✓	✓	✓	✓	✓
Sorteio, televentas, premiação	✓	✓	✓	✓	
Debate / Entrevista	✓	✓	✓	✓	
Desenho adulto	✓	✓			
Político	✓	✓	✓	✓	
Religioso	✓	✓	✓	✓	

Tabela 16 – Hipótese dos Tipos de Objetos de Mídias Especificados em Aplicações que possuem Relação Semântica com programas de TV.

Como pode ser observado na Tabela 16, de acordo com a hipótese deste anexo, o uso dos objetos de mídia pode ser agrupado em três conjuntos:

- 1) Uso dos objetos de imagem e texto: making of, culinária, moda, rural e desenho adulto;

- 2) Uso dos objetos de imagem, texto, NCLua e XHTML: jornalismo, esporte, novela, minissérie, série/seriado, auditório, feminino, game show, reality show, saúde, turismo, humorístico, erótico, sorteio, televidas, premiação, debate/entrevista, político e religioso;
- 3) Uso dos objetos de imagem, texto, NCLua, XHTML e áudio: educativo, show, musical, infantil e filme.

Com o objetivo de representar os três conjuntos citados, três aplicações foram especificadas. Para as medições na implementação COMP, nessas aplicações, os objetos de mídia têm suas apresentações disparadas sequencialmente, como segue:

- Conjunto 1: apresentação em seqüência de um objeto de imagem, seguido de um objeto de texto. Essa seqüência é repetida duas vezes;
- Conjunto 2: apresentação em seqüência de um objeto de imagem, seguido de um objeto de texto, seguido de um objeto declarativo com código HTML, seguido de um objeto imperativo com código Lua. Essa seqüência é repetida duas vezes;
- Conjunto 3: apresentação em seqüência de um objeto de imagem, seguido de um objeto de texto, seguido de um objeto declarativo com código HTML, seguido de um objeto imperativo com código Lua, seguido de um objeto de áudio; seqüência que é repetida duas vezes.

Para a implementação MONO, os objetos de mídia de texto, imagem, áudio, HTML e Lua, são disparados simultaneamente no início da aplicação, para, em seguida, iniciar as seqüências definidas nas três aplicações acima. Como discutido na Seção 3.3, o objetivo é obrigar o carregamento de todos os componentes decodificadores pela biblioteca DirectFB desde o início, ou seja, sem utilizar sua facilidade de carga dinâmica, simulando assim o comportamento monolítico, não componentizado.

Para se obter um melhor resultado e não fazer medições desnecessárias é necessário calcular o número ideal de medições que devem ser feitas. Para esse cálculo, será considerado um fator de confiança, o erro máximo desejado e também o desvio padrão, obtido a partir de uma coleta prévia de medições (dez medições).

A Equação 1 apresenta a fórmula utilizada para determinar o tamanho amostral n , onde: E é o erro permissível, Z é o valor da variável normal padrão associado ao grau de confiança adotado e s é o desvio padrão da coleta prévia.

$$n = \left(\frac{Z \cdot s}{E} \right)^2$$

Equação 1 – Tamanho da amostra.

Para a avaliação do consumo de memória, a coleta prévia apresentou o desvio padrão máximo de 722,26 kB. Considerando o valor 1,96 (valor da variável normal padrão associado ao grau de confiança 95%) e que o erro não supere 350 kB; obtém-se o valor 16,36 (17 medições) como o tamanho da amostra n . A Figura 15, Figura 17 e Figura 19 apresentam gráficos comparativos da quantidade de código alocado na memória entre as duas implementações (MONO e COMP) enquanto realizam a apresentação das aplicações que representam, respectivamente, o conjunto 1, 2 e 3.

A coleta prévia para avaliação do consumo de CPU (porcentagem) apresentou o desvio padrão máximo de 0,13. Considerando o valor 1,96 (valor da variável normal padrão associado ao grau de confiança 95%) e que o erro não supere 0,07 %; obtém-se o valor 16,27 (17 medições) como o tamanho da amostra n . A Figura 16, Figura 18 e Figura 20 apresentam gráficos comparativos do uso de CPU entre as duas implementações (MONO e COMP) enquanto realizam a apresentação das aplicações que representam, respectivamente, o conjunto 1, 2 e 3.

Nas figuras citadas, para as medições da implementação COMP, o valor médio é representado na cor verde. O valor máximo do intervalo (média COMP + intervalo de confiança) é representado na cor vermelha e o valor mínimo do intervalo (média COMP – intervalo de confiança) é representado na cor lilás. Para as medições MONO, o valor médio é representado na cor laranja. O valor máximo do intervalo (média MONO + intervalo de confiança) é representado na cor ciano e o valor mínimo do intervalo (média MONO – intervalo de confiança) é representado na cor cinza.

Além dos resultados discutidos na Seção 3.3, deve ser notado nos gráficos que para a apresentação das aplicações representativas dos três conjuntos em discussão, o valor mínimo do intervalo medido para a implementação MONO está

sempre acima do valor máximo do intervalo medido para a implementação COMP.

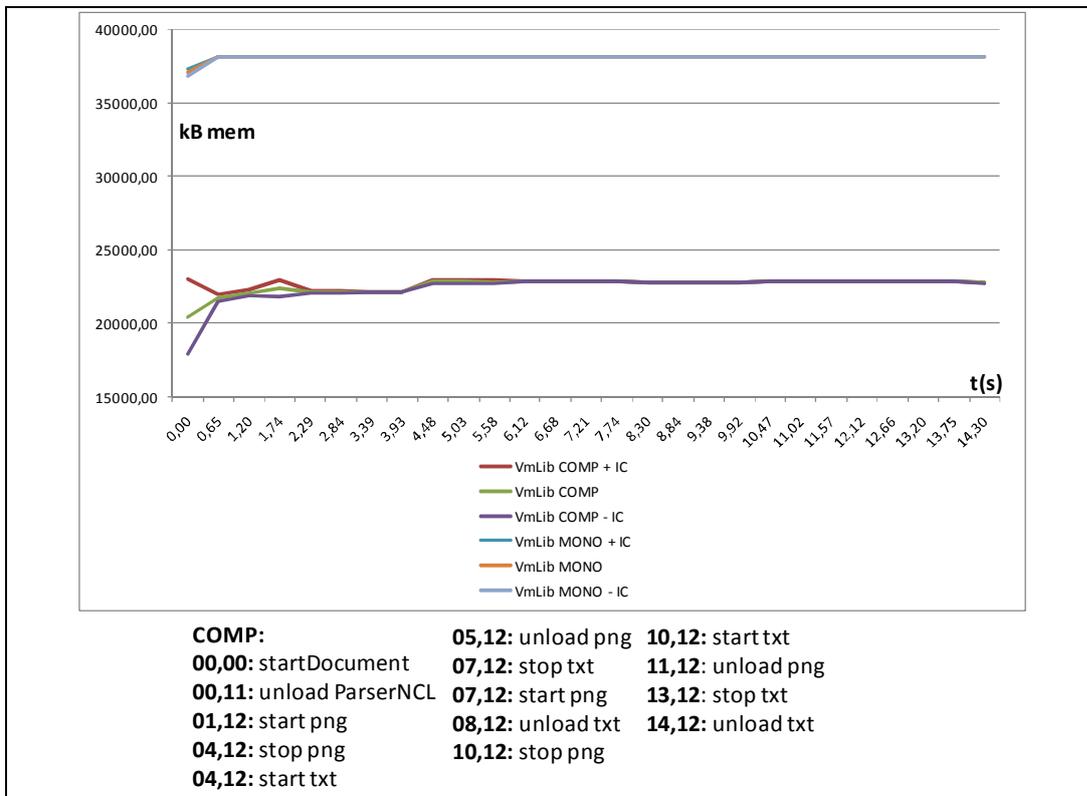


Figura 15 – Quantidade de Código Alocado na Memória para a Apresentação da Aplicação Representativa do Conjunto 1.

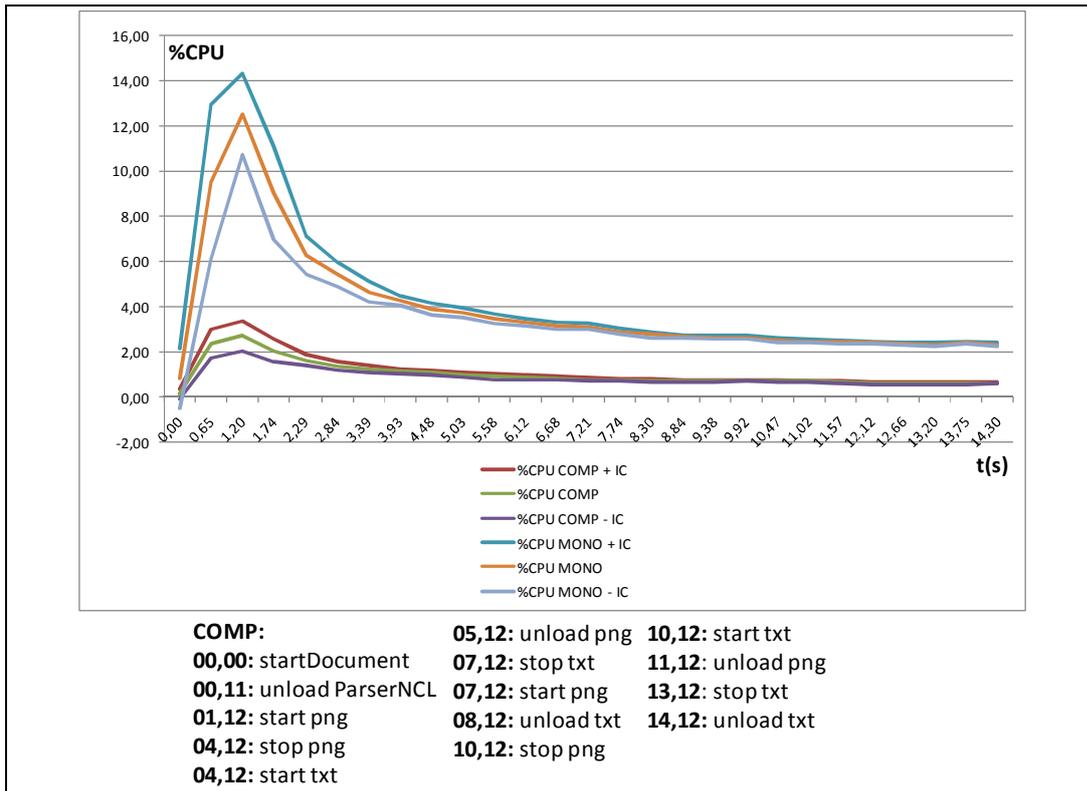


Figura 16 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 1.

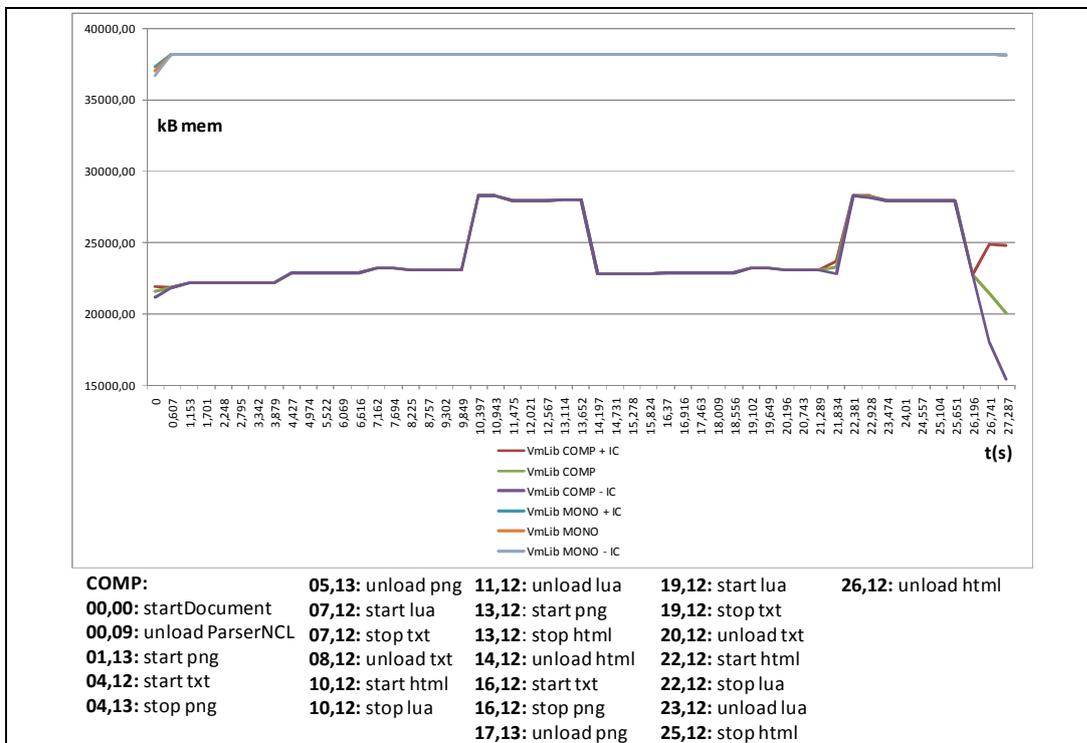


Figura 17 – Quantidade de Código Alocado na Memória para a Apresentação da Aplicação Representativa do Conjunto 2.

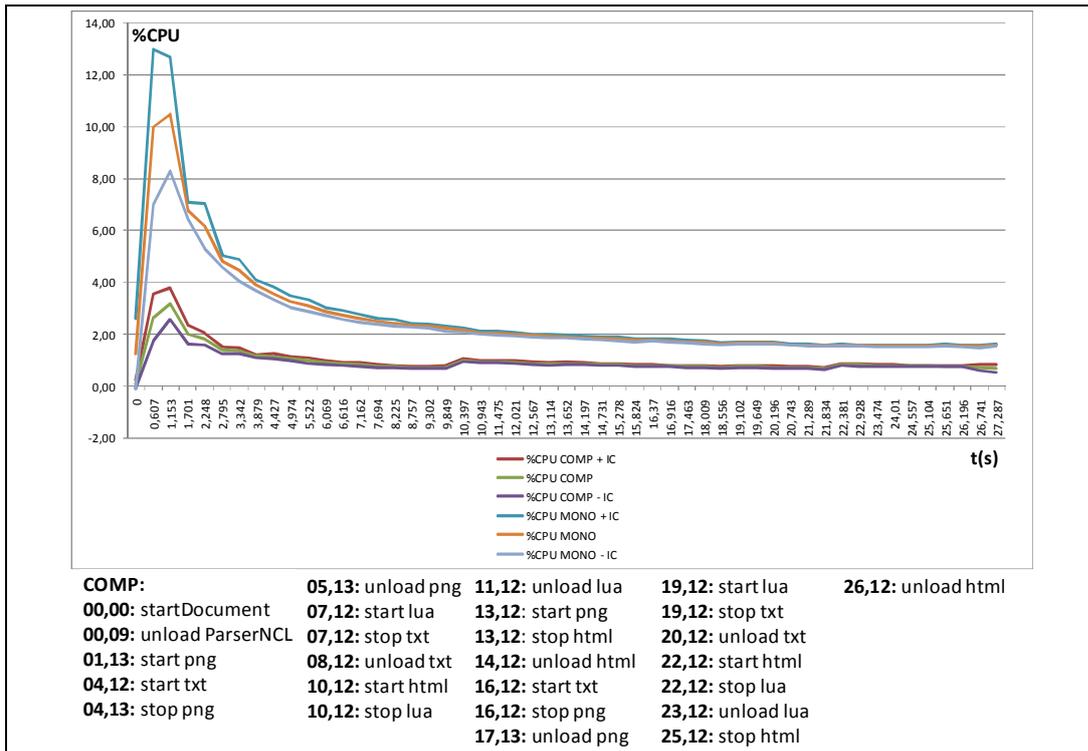


Figura 18 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 2.

PUC-Rio - Certificação Digital Nº 0621320/CA

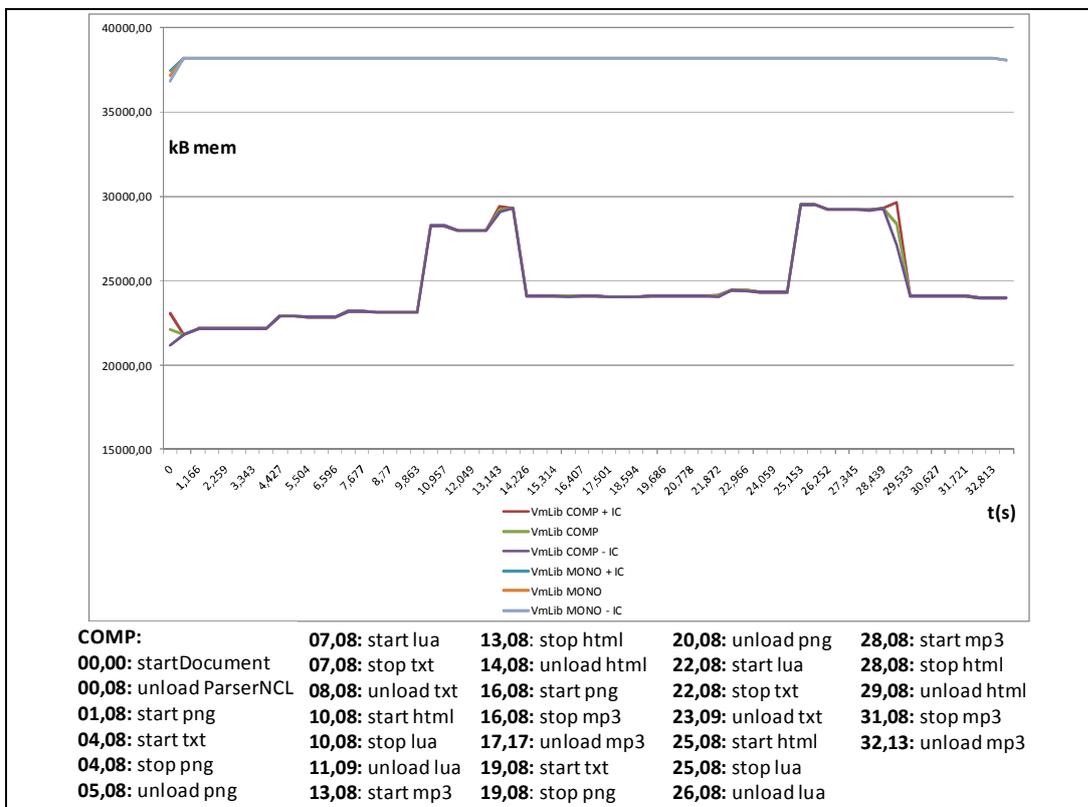


Figura 19 – Quantidade de Código Alocado na Memória para a Apresentação da Aplicação Representativa do Conjunto 3.

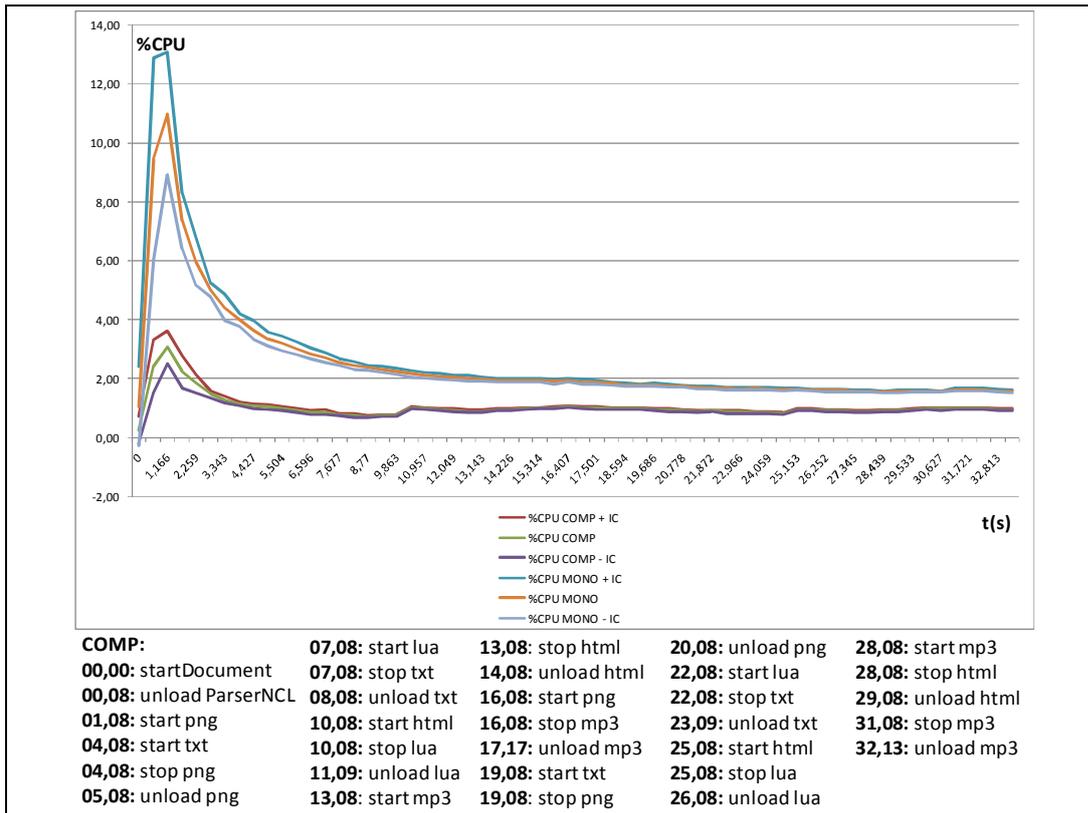


Figura 20 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 3.

Anexo IV – Avaliação do Plano de Recuperação por Medições

Este anexo apresenta medições realizadas para o plano de recuperação, considerando a injeção de falhas. O objetivo é medir o tempo necessário para as ações de recuperação.

Como discutido no Capítulo 4, um ponto que pode comprometer a confiabilidade de um middleware para TVD é o uso de bibliotecas de terceiros. Como apresentado no Capítulo 3, para a decodificação e renderização de conteúdo, referenciado pelos objetos de mídia, foram utilizadas diversas bibliotecas.

A Tabela 17 apresenta o tempo necessário para a recuperação dos objetos de mídia ao injetar uma falha crítica (SIGKILL) durante sua exibição. De forma similar às medições apresentadas no Anexo III, é considerado um grau de confiança de 95% para as medições. Além disso, a Equação 1 (veja Anexo III) foi utilizada para definir o tamanho da amostra.

A coleta prévia para avaliação do tempo necessário para a recuperação de objetos apresentou o desvio padrão máximo de 0,03. Considerando o valor 1,96 (valor da variável normal padrão associado ao grau de confiança 95%) e que o erro não supere 0,02s; obtém-se o valor 8,8 (9 medições) como o tamanho da amostra n .

Como pode ser notado na Tabela 17, o tempo necessário para a recuperação de um objeto de mídia é menor que um décimo de segundo.

	Objetos t(s)			
	Imagem	Texto	XHTML	Vídeo
Medição 1	0,016	0,013	0,042	0,033
Medição 2	0,014	0,016	0,063	0,030
Medição 3	0,037	0,080	0,067	0,042
Medição 4	0,035	0,039	0,080	0,018
Medição 5	0,017	0,018	0,013	0,024
Medição 6	0,038	0,079	0,038	0,036
Medição 7	0,030	0,084	0,044	0,036
Medição 8	0,044	0,037	0,080	0,036
Medição 9	0,014	0,017	0,029	0,039
Média	0,027	0,043	0,051	0,033
Desvio Padrão	0,012	0,030	0,023	0,008
Intervalo de Confiança (IC)	0,007	0,019	0,014	0,005
Valor mínimo (média - IC)	0,020	0,024	0,036	0,028
Valor máximo (média + IC)	0,035	0,061	0,065	0,037

Tabela 17 – Tempo de Recuperação de Objetos para Injeção de Falha SIGKILL.