

7

Referências Bibliográficas

- 1 OMG - OBJECT MANAGEMENT GROUP. **CORBA: Common Object Request Broker Architecture**. Version 3.0. An Adopted Specification of the Object Management Group, Inc., November, 2002. Editorial edit to cover formal/02-11-03.
- 2 CERQUEIRA, R. F. G.; RODRIGUEZ, N. L. R.; IERUSALIMSCHY, R. **Binding an Interpreted Language to CORBA**. In II Simpósio Brasileiro de Linguagens de Programação. Campinas, Brazil, September 1997. P. 23-36.
- 3 HENNING, M.; VINOSKI, S. **Advanced CORBA Programming with C++**. Addison-Wesley Publisher, 1999.
- 4 SIEGEL, J. **CORBA 3 - Fundamentals and Programming**. Second Edition. OMG Press. Wiley Computer Publishing, 2000.
- 5 CERQUEIRA, R. F. G. **Um Modelo de Composição Dinâmica entre Sistemas de Componentes de Software**. Tese de Doutorado. Departamento de Informática. PUC-Rio. Rio de Janeiro, Brasil, Agosto, 2000.
- 6 CERQUEIRA, R. F. G.; NOGUEIRA, L.; MOURA, A. L. **The LuaORB Manual**. Version 1.5. Departamento de Informática. PUC-Rio. Rio de Janeiro, Brasil, 2000.
- 7 IERUSALIMSCHY, R.; FIGUEIREDO, L. H.; CELES, W. **Reference Manual of the Programming Language Lua 5.0**. TecGraf. PUC-Rio. Rio de Janeiro, Brasil, 2003.
- 8 IERUSALIMSCHY, R.; FIGUEIREDO, L. H.; CELES, W. **Lua - An Extensible Extension Language. Software: Practice and Experience**. Departamento de Informática. PUC-Rio. Rio de Janeiro, Brasil, 1996.
- 9 CHU, H. H. **CPU Service Classes: A Soft Real Time Framework for Multimedia Applications**. Ph.D. Thesis. Department of Computer Science. University of Illinois at Urbana-Champaign. August, 1999.
- 10 YUAN, W.; NAHRSTEDT, K.; KIM, K. **R-EDF: A Reservation-Based EDF Scheduling Algorithm for Multiple Multimedia Task Classes**. In Proceedings of the 7th IEEE Real-Time Technology and Applications Symposium. Taipei, Taiwan, May, 2001.
- 11 YAMANE, T. **The Design and Implementation of the 2K Resource Management Service**. Master's Thesis. Department of Computer Science. University of Illinois at Urbana-Champaign. February, 2000.
- 12 IONA. **ORBacus for C++ and Java**. Version 4.1.0. An Adopted Specification of the IONA Technologies, Inc. 2000.
- 13 MOURA, A. L.; URURAHY, C.; CERQUEIRA, R. F. G.; RODRIGUEZ, N. L. R. **Dynamic Support for Distributed Auto-Adaptive Applications**. In Proceedings of AOPDCS'02 - Workshop on Aspect Oriented Programming for Distributed Computing Systems (held in conjunction with IEEE ICDCS 2002). Vienna, Austria, July, 2002. P. 451-456.
- 14 MOURA, A. L. **Um Ambiente de Suporte à Adaptação Dinâmica de Aplicações Distribuídas**. Dissertação de Mestrado. Departamento de Informática. PUC-Rio. Rio de Janeiro, Brasil, Setembro, 2000.
- 15 Kon, F.; Campbell, R.; Mickunas, D.; Nahrstedt, K. **2K: Distributed Component-Based System**. In Proceedings of the 9th IEEE International Symposium on High Performance Distributed Computing (HPDC-9). Pittsburgh, PA. August 2000.

- 16 KON, F.; YAMANE, T.; HESS, C. K.; CAMPBELL, R. H.; MICKUNAS, M. D. **Dynamic Resource Management and Automatic Configuration of Distributed Component Systems**. In Proceedings of the 6th USENIX COOTS'2001 (Conference on Object-Oriented Technologies and Systems). San Antonio, Texas, January, 2001.
- 17 KON, F.; CAMPBELL, R. H.; NAHRSTEDT, K. **Using Dynamic Configuration to Manage a Scalable Multimedia Distribution System**. Communication Journal (Special Issue on QoS-Sensitive Distributed Systems and Applications), Vol. 24. Elsevier Science Publisher. P. 105-123.
- 18 MARQUES, J. R.; KON, F. **Gerenciamento de Recursos Distribuídos em Sistemas de Grande Escala**. Anais do 20º SBRC2002 (Simpósio Brasileiro de Redes de Computadores). Búzios, Rio de Janeiro, Brasil, Maio, 2002. P. 800-813.
- 19 CERQUEIRA, R. F. G.; CASSINO, C. R. S. P.; IERUSALIMSKY, R. **Dynamic Component Gluing Across Different Componentware Systems**. In International Symposium on Distributed Objects (DOA'99). Edinburgh, Scotland, 1999. IEEE Press.
- 20 FIGUEIREDO, L. H.; IERUSALIMSKY, R.; CELES, W. **Lua - An Extensible Embedded Language**. Dr. Dobbs's Journal. December, 1996. P. 21(12):26-33.
- 21 RODRIGUEZ, N. L. R.; IERUSALIMSKY, R. **Dynamic Reconfiguration of CORBA-Based Applications**. In Proceedings of AOPDCS'02 - Workshop on Aspect Oriented Programming for Distributed Computing Systems (held in conjunction with IEEE ICDCS 2002). Springer-Verlag. (LNCS 1725). Milovy, Czech Republic, 1999. P. 451-456.
- 22 RODRIGUEZ, N. L. R.; IERUSALIMSKY, R.; CERQUEIRA, R. F. G. **Dynamic Configuration with CORBA Components**. In 4th International Conference on Configurable Distributed Systems. Anápolis, Goiás, Brasil, MD, 1998. P. 27-34.
- 23 OMG - OBJECT MANAGEMENT GROUP. **Reference Model of Open Distributed Processing**. Part 1. An Adopted Specification of the Object Management Group, Inc. Overview. May, 1995.
- 24 MOWBRAY, T. J.; RUTH, W. A. **Inside CORBA: Distributed Object Standards and Applications**. Addison-Wesley Publisher. 1998.
- 25 ORFALI, R.; HARKEY, D. **Client/Server Programming with JAVA and CORBA, 2**. Wiley Computer Publisher. 1998.
- 26 OLIVEIRA, R. S. **Mecanismos de Adaptação para Aplicações Tempo Real na Internet**. Congresso da Sociedade Brasileira de Computação (SEMISH'98). Belo Horizonte, Minas Gerais, Brasil, 1998.
- 27 GOMES, A. T. A.; COLCHER, S.; SOARES, L. F. G. **Um Framework para Provisão de QoS em Ambientes Genéricos de Processamento e Comunicação**. In Anais do XVII Simpósio Brasileiro de Redes de Computadores (SBRC'99). Salvador, Bahia, Brasil, Maio, 1999.
- 28 FOSTER, I.; KESSELMAN, C.; LEE, C.; LINDELL, R.; NAHRSTEDT, K.; ROY, A. **A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation**. International Workshop on Quality of Service (QoS). 1999.
- 29 FOSTER, I.; KESSELMAN, C. **The Globus Project: A Status Report**. In Proceedings of the IPPS/SPDP '98 Heterogeneous Computing Workshop. 1998. P. 4-18.
- 30 Framework CSBase. <http://www.tecgraf.puc-rio.br/csbase/>.
- 31 CZAJKOWSKI, K.; FOSTER, I.; KARONIS, N.; KESSELMAN, C.; MARTIN, S.; SMITH, W.; TUECKE, S. **A Resource Management Architecture for Metacomputing Systems**. Proc. IPPS/SPDP'98 Workshop on Job Scheduling Strategies for Parallel Processing. 1998. P. 62-82.

- 32 CHAPIN, S.; KATRAMATOS, D.; KARPOVICH, J.; GRIMSHAW, A. **Resource Management in Legion**. Proceedings of the 56th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP'99) in conjunction with the International Parallel and Distributed Processing Symposium (IPDPS'99). San Juan, Puerto Rico, April, 1999.
- 33 BASNEY, J.; LIVNY, M. **Managing Network Resources in Condor**. Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9). Pittsburgh, Pennsylvania, August, 2000. P. 298-299.
- 34 RAMAN, R.; LIVNY, M. **High Throughput Resource Management**. Chapter 13 in The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco, California, 1999.
- 35 TANENBAUN, A. S.; RENESSE, R.; STAVEREN, H.; SHARP, G. J.; MULLENDAR, S. J.; JANSEN, J.; ROSSUM, G. **Experiences with the Amoeba Distributed Operating System**. Communications of the ACM. December, 1990. P. 33(12):47-63.
- 36 OMG - OBJECT MANAGEMENT GROUP. The Common Object Request Broker: Architecture and Specification. Revision 2.0. July, 1995.
- 37 OMG - OBJECT MANAGEMENT GROUP. <http://www.omg.org/news/about/index.htm> .
- 38 CATUNDA, M. **Extensão Dinâmica de Agentes CORBA**. Dissertação de Mestrado. Departamento de Informática. PUC-Rio. Rio de Janeiro, Brasil, 1998.
- 39 Nahrstedt K.; Chu, H.; Narayan, S. **QoS-Aware Resource Management for Distributed Multimedia Applications**. Journal on High-Speed Networking. Special Issue on Multimedia Networking. IOS Press. Vol. 8. Num. 3-4. 1998. P. 227-255.

8 Anexos

1. IDL de gerência do *framework* CSBase (arquivo sga-daemon.idl):

```

... ..
/**
 * Informações para monitoração de um processo.
 * As informações sobre um processo em execução na máquina hospedeira
 * são tipicamente obtidas através da execução do comando 'ps'.
 */
struct ProcessInfo {
    long pid;
    long ppid;
    string command;
    ProcessState state;
    long processorId;
    long memoryRamSizeMb;
    long memorySwapSizeMb;
    double CPUPerc;
    long CPUTimeSec;
    long wallTimeSec;
};

/ * Informações para monitoração de um comando.
 * A execução de um comando pode gerar um ou mais processos, portanto são
 * fornecidas as informações de todos os processos que compõem o comando.
 */
typedef sequence<ProcessInfo> CommandInfo;

/**
 * Carga na máquina hospedeira do SGA.
 * Essa informação corresponde ao percentual de ocupação do(s)
 * processador(es) do nó. Deve-se observar que este
 * percentual pode ser maior que 100% (valor superior a 1.0) caso o número
 * de processos ativos exceda o número de processadores disponíveis.
 */
struct HostLoadAvgPerc {
    double loadAvg1min; /**< carga no último minuto */
    double loadAvg5min; /**< carga nos últimos 5 minutos */
    double loadAvg15min; /**< carga nos últimos 15 minutos */
};
... ..

```

```

... ..
/**
 * Requisição de execução de um comando.
 * @param command comando a ser executado
 * @param cmdid identificador do comando a ser executado
 * @return referência para o comando
 */
SGACommand executeCommand( in string command, in string cmdid, in
string hostName, in long lPeriod, in float dPercentage);
... ..

```

2. Implementação da interface do serviço SGA_DAEMON (arquivo sga-command.lua):

```

... ..
function Command( sga_command )
if not sga_command.command then
    SGA_DAEMON:writeLog( "Comando nulo abortado!" )
    return nil
end
sga_command.command = tostring( sga_command.command )

if not sga_command.id then
    SGA_DAEMON:writeLog( "Comando sem identificador abortado!" )
    return nil
end
sga_command.id = tostring( sga_command.id )

-- Verifica se a maquina escolhida existe ou escolhe uma maquina.
local host = sga_command.host
local output
local err_output
if host then
    if host ~= SGAD_CONF.name then
        local nodes = SGAD_CONF.nodes
        local i = 1
        local found
        while nodes[i] and not found do
            if nodes[i].name == host then found = 1 end
            i = i + 1
        end
        if not found or not SGAD_CONF.network[ host ] then
            local errmsg = "No configuration found for host: "..host
            SGA_DAEMON:writeLog( "\n\n"..errmsg.."!!!\n\n")

```

```

        return nil, errmsg
    end
    else host = nil
    end
elseif SGA_DAEMON:isCluster() then
    local nodes = SGA_DAEMON:getNodesDynamicInfo()
    local minAvg = nodes[1].load_avg_perc.loadAvg1min
    host = nodes[1].name
    for i,v in nodes do
        if type(v) == "table" and v.load_avg_perc.loadAvg1min < minAvg then
            minAvg = v.load_avg_perc.loadAvg1min
            host = v.name
        end
    end
    output = "/dev/null"
    err_output = "/dev/null"
end
sga_command.host = host
-- escolheu uma maquina para a execucao do comando...

print("\nsga_command.command = "..sga_command.command)
print("sga_command.Period = "..sga_command.Period)
print("sga_command.Percentage = "..sga_command.Percentage)

local handle, err = SGAD_CONF.serverManager.executecommand(
    sga_command.command, nil, output, err_output, sga_command.host,
    sga_command.Period, sga_command.Percentage)
... ..

```

3. Programa C++ que recebe valores de Lua (arquivo sgaWindows.cpp):

```

... ..
if ( lua_isstring( L , 6 ) )
{
    CPU_IPeriod = lua_tonumber( L , 6 ) ;

    printf("\nCPU_IPeriod = %u, ", CPU_IPeriod); // %d

}
else
{
    /* lua_pushnil( L ) ;
    lua_pushstring( L , "Periodo de CPU não fornecido.\0" ) ;
    return 2 ;*/
}

```

```

if ( lua_isstring( L , 7 ) )
{
    CPU_dPercentage = lua_tonumber( L , 7 ) ;

    printf("\nCPU_dPercentage = %g, ", CPU_dPercentage);

}
else
{
    /* lua_pushnil( L ) ;
    lua_pushstring( L , "Percentual de reserva de CPU não fornecido.\0" ) ;
    return 2 ;*/
}

STARTUPINFO si;
PROCESS_INFORMATION pi;
ZeroMemory(&si, sizeof(STARTUPINFO));
si.cb = sizeof(STARTUPINFO);

if (defFiles == TRUE)
    si.dwFlags = STARTF_USESTDHANDLES ;
si.hStdOutput = hOutFile ;
si.hStdInput = hInFile ;
si.hStdError = hErrorFile ;

// Necessário para passar para a função CreateProcess
LPTSTR _CommandLine = ( LPTSTR ) malloc( sizeof( char ) * strlen(
CommandLine ) ) ;
strcpy( _CommandLine , CommandLine ) ;

if ( CreateProcess( NULL , // Application name, não condiz com linha de
comando
                    _CommandLine , // Linha de comando a ser executada
                    &processHandlesSecurity , // Com herança de handles de processo
                    &threadHandlesSecurity , // Com herança de handles de Threads
                    defFiles , // Com herança de handles "herdáveis"
                    CREATE_NEW_CONSOLE , // Flags de criação
                    NULL , // Flags de Ambiente
                    NULL , // Diretório da aplicação
                    &si , // Dados para inicialização de processos
                    &pi // Estrutura a ser preenchida com handles e ids do processo e
main thread
                    ) == TRUE )
{

    HANDLE hThrd;

    if(!DuplicateHandle(GetCurrentProcess(), pi.hThread, pi.hProcess, &hThrd,
        THREAD_ALL_ACCESS, FALSE, DUPLICATE_SAME_ACCESS))

```

```

    ErrorMessage("DuplicateHandle");

    CCPUApi* pCpuApi = new CCPUApi(pi.dwProcessId, pi.dwThreadId,
hThrd);
    CCPUReserve reserve(CPU_IPeriod, CPU_dPercentage);
    pCpuApi->Reserve(reserve);
    pCpuApi->StartRTRun();
    delete pCpuApi;
... ..

```

4. Implementação da exportação do sga.ior:

```

... ..
function SGA_DAEMON:initServer()
-- verifica a configuração dos nós
if not SGAD_CONF.nodes or type(SGAD_CONF.nodes) ~= type({}) then
    return nil, "Falha na configuração dos nós do SGA."
end

if not LuaOrbCfg.getIFR() or LuaOrbCfg.getIFR():_non_existent() then
    return nil, "Repositório de interfaces não encontrado!"
end

if SGAD_CONF.luaorb_verbose then
    LuaOrbCfg.verbose_mode = 1
end

-- ativa o servidor CORBA para o SGAD
self.servant = lo_createservant(
    self,
    "csbase::server::services::sgaservice::idl::SGAServer"
)
if not self.servant then
    return nil, "Impossível ativar o servidor SGAD"
end
writeto("sga.ior")
write(self.servant:_get_ior())
writeto()

-- obtém a localização do SSI e cria o seu proxy
if not SGAD_CONF.ssi_ior then
    return nil, "Falha na configuração do IOR do SSI."
end
self.ssi = lo_createproxy(
    SGAD_CONF.ssi_ior,

```



```
        "csbase::server::services::sgaservice::idl::SGAManager"
    )
    if not self.ssi then return nil, "Impossível localizar o SSI." end

    -- inicializacao do campo enabled do SGA
    if (not SGAD_CONF.nodes[1].disabled) then
        self.is_enabled = 1
    end

    return 1
end
... ..
```
