

面向天基实时系统的 QoS 监控算法研究^{*})

曾斌^{1,2} 李之棠¹ 王威² 刘海庆²

(华中科技大学计算机学院 武汉 430074)¹ (海军工程大学管理工程系 武汉 430030)²

摘要: 天基网研究正日益得到广泛的重视,但如果天基网缺乏相应的 QoS 机制,系统过载时将会造成巨大的损失。一个提供多类服务的基于路径的服务质量确保策略被提出来,它建立了一个连续 QoS 监测模型,以路径响应延迟为核心度量标准,将同一类应用抽象为一条虚拟路径,并以路径黏合起系统的各个组件。同时据此提出了一套动态自适应的 QoS 监控算法。测试表明,即使经历严重的过载与业务突发,也能保证用户感知的服务质量。

关键词: 天基实时系统, QoS 监控, 路径响应延迟, 网络监测

Research of QoS Monitor Algorithm for Space Based Real-time System

ZENG Bin^{1,2} LI Zi-Tang¹ WANG Wei² LIU Hai-Qing²

(School of Computer, Huazhong University of Science and Technology, Wuhan 43007)¹

(Department of Management, Navel University of Engineering, Wuhan 430030)²

Abstract Space Based Network (SPN) is going to be an important part of the future military research. But if the current SPN architecture is short of QoS ensuring mechanism, it will lead to significant losses while system is overloaded. A QoS ensuring method for SPN which provides multi-class service is presented in this paper. It uses path response delay as the core measurement criterion of the architecture based on a continuous model of resource requirements. One class application is abstracted to a virtual path, and every component of SPN are glued tighter. And a set of dynamic adaptive QoS diagnosis algorithm is also presented. Experiments show that, although suffering severe overloading and burst, the user-perceived QoS can be ensured.

Keywords Space based real-time system, QoS monitor, Path response delay, Network monitoring

1 引言

天基网系统在运行中即要保证安全运行,又要保证任务能够实时完成,它所有的子系统互相关联,在统一的时限要求下协同工作。它的输入数据来源于不断监视地球环境的传感卫星,所以工作负载根据环境的不同而动态变化^[1]。如果要保证这样一个系统的 QoS 需求,需要对其延迟事件和错误率进行有效监测并能灵活地识别出 QoS 故障的问题所在。

天基网作为一个复杂的分布式实时系统,它由一系列时间延迟受限和优先级受限的子系统集合而成,这些子系统相互配合、相互协作,以流水线或路径的方式共同完成工作。路径又可细分为子路径,它们作为路径的组成部分,即包括完成某种功能的程序模块,又包括各计算模块之间的通信链路。QoS 监测系统不断抽样测量这些子路径,分析统计测量结果,并对 QoS 故障原因作出诊断,再分配系统资源。

如图 1,以天基监测路径为例,当低轨传感卫星监测到威胁目标后,通过两条通信链路传到地面关口站进行信息过滤,随后传送到评估管理系统进行威胁度判别,当目标得到确认后,交由决策中心分析并决定行动方案。这时可能激发作战路径,一个典型的作战方案即为向目标发射导弹。当导弹升空后就进入引导路径,该路径可能包括调整卫星位置、信息过滤、目标评估、方案决策以及向导弹发送引导命令等子路径。

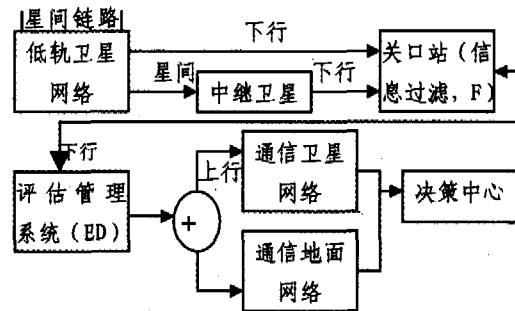


图 1 天基实时系统应用实例

其中监测路径属于数据驱动、软实时路径,如果在一个周期内无法按规定期限内处理数据,该路径还必须处理随后的跟踪数据。当发现目标后,监测频率也会随之提高。由于无法事先知道目标数目和威胁度,也就很难预先获得监测路径工作负载的最大值和平均值。作战路径属于硬实时、事件驱动、一次性路径,由于它必须在规定期限内完成,因此其优先级要高于监测路径和引导路径。引导路径被导弹发射事件驱动,激活后则为周期性路径,由于针对不同数量的目标发射的导弹数目不同,它同监测路径一样无法事先知道工作负载的最大值和平均值。

关于分布实时系统的 QoS 研究,近年来已经取得了一些

^{*})国家“八六三”高技术研究发展计划基金项目(2002AA7170)。

研究成果,例如 RTPOOL^[2]、DQM^[3~5]以及 EPIQ^[6]等,它们主要着眼于分级 QoS 的实现,利用复杂的中间件技术来解决一般网络的拥塞问题。它们划分的 QoS 粒度一般低于路径,主要为任务级,而且假设系统行为遵循一些静态已知的模式,不能有效满足复杂的类似 C3I 系统的天基网络连续性 QoS 需求。

2 监控模型

我们首先利用 PC 机和局域网模拟了天基卫星网络的基本特性^[7],并设计了一套软件包仿真天基信息网的任务级应用,例如传感数据仿真器(SM)、过滤算法仿真器(FM)、评估决策仿真软件(DM)等,它们之间通过实时中间件进行连接。而本文描述的 QoS 监控器作为实时中间件的一部分,用于对天基应用软件执行实时 QoS 保障。主要功能包括:1)实时监测路径的性能变化和资源使用情况,确定瓶颈位置和解决方案。2)确定满足路径 QoS 需求的副本进程数目以及最合适的宿主机。3)利用实时中间件发送调控命令。

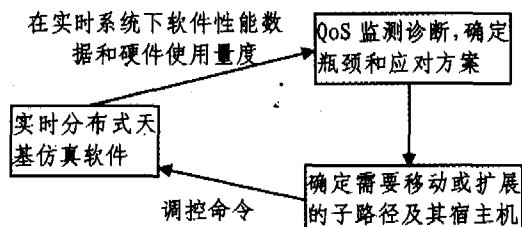


图 2 QoS 监控过程

表 1 系统模型的符号定义

| 符号 | 含义 |
|------------------------------------|--|
| c | 当前采样周期 |
| P_i | 路径 i |
| $SP_{i,j}$ | 子路径,路径 i 中某一应用 j |
| $SP_{i,j,k}$ | 应用 j 在路径 i 中运行的进程 k |
| $DL(P_i)$ | 路径 i 的处理期限 |
| $QD(P_i)$ | 路径 i 的排队延迟 |
| $\lambda_{OBS}(P_i, c)$ | 在周期 c 内测得的路径 i 实际延迟 |
| $\Delta_{PROF}(P_i, DS(P_i, c))$ | 在当前负载下的路径 i 先验延迟 |
| $\lambda_{EST}(SP_{i,j}, d, u)$ | $SP_{i,j}$ 在宿主机使用情况为 u , 数据量为 d 时的预估执行时间 |
| $\lambda_{EST}(DS, d, c)$ | 消息 $DS(d$ 字节) c 内的预估传输延迟 |
| $SL_{MIN}(P_i)$ | 路径 i 要求的最小松弛度 |
| $SL_{MAX}(P_i)$ | 路径 i 要求的最大松弛度 |
| $HOSTS(SP_{i,j})$ | 子路径 $SP_{i,j}$ 占用的计算资源集合 |
| $RES(P_i)$ | 路径 i 可使用的资源集合 |
| $DS(P_i, c)$ | 在周期 c 内路径 i 处理的数据流 |
| $DS(SP_{i,j}, c)$ | 在周期 c 内 $SP_{i,j}$ 处理的数据流 |
| $DS(SP_{i,j}, SP_{i,j+1}, c)$ | 在周期 c 内子路径间 ($SP_{i,j}, SP_{i,j+1}$) 传输的数据流 |
| $DS(SP_{i,j,k}, SP_{i,j,k+1}, c)$ | 在周期 c 内副本进程间 ($SP_{i,j,k}, SP_{i,j,k+1}$) 传输的数据流 |

我们用 $\{P_1, P_2, P_3, \dots\}$ 表示实时应用的路径集合,每条路径由一系列串行工作的子路径(执行程序或通信链路)组成,用 $P_i = \{SP_{i,1}, SP_{i,2}, SP_{i,3}, \dots\}$ 表示,周期性路径 P_i 的当前周期用 $c(P_i)$ 表示, P_i 在周期 c 内处理的数据项数目用 $|DS(P_i, c)|$ 表示。本文假设系统中的计算子路径都可以产生

多个副本进程,所有的副本共享源子路径的数据流, $SP_{i,j} = \{SP_{i,j,1}, SP_{i,j,2}, SP_{i,j,3}, \dots\}$ 。其他定义见表 1 及后文。

3 算法描述

3.1 QoS 诊断

如果 $\lambda_{OBS}(P_i, c) > DL(P_i)$, 则证明路径 P_i 有故障。我们把故障原因分为两类:一类是由于工作负载过大引起的延迟;另一类是由于竞争冲突引起的路径延迟。这两类延迟的解决方案是不同的^[8]。当多个进程或数据流争用相同资源时,会造成冲突。这时仍有足够资源,最好的解决方法是绕开冲突资源。对于计算子路径,可以把其迁移到负载较轻的主机上运行,对于通信子路径(计算子路径之间的链路),可以把其迁移到负载较轻的网络上。当资源不够时则造成过载,如果计算子路径过载,可以通扩展其计算资源,如果通信子路径过载,可以启用备用链路或优化路由参数。

为了区分过载和冲突这两种故障,我们引入两个定义。排队延迟 $QD(P_i)$:它是无冲突时先验延迟(记录在说明文件中,预先通过实验或计算得出的延迟)和实际测量延迟的差值,计算公式为

$$QD(P_i) = \lambda_{OBS}(P_i, c) - \lambda_{PROF}(P_i, DS(P_i, c)) \quad (1)$$

最大松弛度 $SL_{MAX}(P_i)$:处理时限和先验延迟之差,它表示排队延迟的最大可允许时间,计算公式为

$$SL_{MAX}(P_i) = DL(P_i) - \lambda_{PROF}(P_i, DS(P_i, c)) \quad (2)$$

因此我们的算法为:当发生故障时,如果 $SL_{MAX}(P_i) < 0$, 认为路径 P_i 过载,选择 $SL_{max}(SP_{i,j}), j=1, \dots, k$ 最大的子路径为瓶颈路径,对其创建副本直至路径 P_i 达到需求延迟;如果 $QD(P_i) > SL_{MAX}(P_i)$, 则路径 P_i 冲突,选择 $QD_{max}(SP_{i,j}), j=1, \dots, k$ 最大的子路径为瓶颈路径,把它迁移到负载较轻的主机或网络上。

3.2 基于预测的副本创建算法

当路径过载时,需要对瓶颈子路径创建合适数量的副本进程。传统方式是根据当前可用资源情况来确定副本数量,这种方法不能有效适应天基应用 QoS 的动态变化,为此我们建立了一个基于预测的自适应副本创建算法。该算法的基本思想是:首先利用统计回归方法预测增加副本后该子路径的处理延迟,并推导出瓶颈子路径的处理时限,随后不断增加副本直至预测延迟小于时限。

首先通过实时 benchmark 软件获得天基仿真系统的先验数据,并以宿主机利用率和工作负载为自变量,利用回归方程拟合出各子路径的延迟预测函数。本算法采用的回归方程如下:

$$\lambda_{EST}(SP_{ij}, d, u) = f(d, u) = (a_1 u^2 + a_2 u + a_3) d^2 + (b_1 u^2 + b_2 u + b_3) d \quad (3)$$

其中 d 为子路径处理的数据项数目, u 为宿主机 CPU 利用率(%),延迟以毫秒为单位。

为了预测两个相邻子路径的通信延迟,算法把延迟分作两个部分:数据项传输前的缓冲时间 D_{buf} 和在链路间的传输时间 D_{tran} ,其公式如下:

$$\lambda_{EST}(DS(SP_{i,j}, SP_{i,j+1}, c)) = D_{buf}(d, c) + D_{tran}(d) \quad (4)$$

其中 d 为 DS 数据项的数目, c 为路径 P_i 的当前周期。

我们在不同工作负载条件下对天基实时仿真系统进行了 benchmark 测试,发现 D_{buf} 是负载的递增函数,可以用一个简单的线性函数来拟合 D_{buf} 。

$$D_{buf}(d, c) = k \times \sum_{i=1}^d DS(P_i, c) \quad (5)$$

其中 k 为回归线的梯度。

传输时间 D_{tran} 与其他子路径的消息无关, 仅与消息本身大小有关。设 bw 为链路带宽, 消息传输延迟为:

$$D_{tran}(d) = \frac{d}{bw} \quad (4)$$

设 d_{init} 为子路径 $SP_{i,j}$ 最初处理的数据项数目, u_{init} 为 $SP_{i,j}$ 所在宿主机最初利用率, c_{init} 为路径 P_i , 处理 d_{init} 个数据项时所处的周期, $DS(SP_{i,j}, SP_{i,j+1}, c)$ 简称为 $DS_{i,j}$, 则计算子路径 $SP_{i,j}$ 的处理时限为:

$$DL(SP_{i,j}) = \lambda_{EST}(SP_{i,j}, d_{init}, u_{init}) + (DL(P_i) - \sum_{k=1}^m \lambda_{EST}(SP_{i,k}, d_{init}, u_{init}) - \sum_{k=i+1}^m \lambda_{EST}(DT_{i,k}, d_{init}, c_{init})) \times \left[\frac{\lambda_{EST}(SP_{i,j}, d_{init}, u_{init})}{\sum_{k=i}^m \lambda_{EST}(SP_{i,k}, d_{init}, u_{init}) + \sum_{k=i+1}^m \lambda_{EST}(DS_{i,k}, d_{init}, c_{init})} \right] \quad (6)$$

消息 $DS_{i,j}$ 的时限公式为:

$$DL(DS_{i,j}) = \lambda_{EST}(DS_{i,j}, d_{init}, c_{init}) + (DL(P_i) - \sum_{k=1}^m \lambda_{EST}(DS_{i,k}, d_{init}, c_{init}) - \sum_{k=i+1}^m \lambda_{EST}(SP_{i,k}, d_{init}, u_{init})) \times \left[\frac{\lambda_{EST}(DS_{i,j}, d_{init}, c_{init})}{\sum_{k=i}^m \lambda_{EST}(DS_{i,k}, d_{init}, c_{init}) - \sum_{k=i+1}^m \lambda_{EST}(SP_{i,k}, d_{init}, u_{init})} \right] \quad (7)$$

于是副本创建算法描述如下:

```

ReplicateSubPath(SPi,j, c)
/* SPi,j 为子路径, c 为当前周期 */
(1) HOSTBest = FindBestHost() / * 见 3.3 节 * /
(2) RES(Pi) = RES(Pi) - HOSTBest;
(3) HOSTS(SPi,j) = HOSTS(SPi,j) ∪ {HOSTBest};
(4) For every h ∈ Hosts(SPi,j)
    4.1 ui = u(h, t); / * u(h, t) 为 c 周期时 h 主机的利用率 * /
    4.2 di = DS(Pi, c) / | Hosts(SPi,j) |
        / * d 为每个副本需处理的数据量 * /
    4.3 λEST(SPi,j,k, d, u) = (a1u2 + a2u + a3)d2 + (b1u2 + b2u + b3)di
    4.4 λEST(DSi,j,k, d, c) = Dbut(d, c) + Dtran(d);
        / * 其中 Dbut(d, c) = k * ∑ DS(Pi, c), Dtran(d) = d/bw; * /
    4.5 TotalDelayi = λEST(SPi,j,k, d, u) + λEST(DSi,j,k, d, c);
    4.6 If TotalDelayi > DL(SPi,j)
        Goto 1;
End for
    
```

3.3 资源分配算法

从上可知, 无论创建副本或迁移路径, 都需要找到一个最为适合的宿主。为此我们引入了一个适合度函数, 它通过综合考虑主机 QoS 度量和网络 QoS 度量来选择最适合的宿主。对每个备选主机 H_i , 其主机度量包括 CPU 利用率 $u(H_i)$ 和使用趋势 $ut(H_i)$ 。而网络度量主要针对(重心主机, 备选主机)对, 其中重心主机定义为与瓶颈子路径通信密集的主机集。网络度量包括网络吞吐量 $netu(H_i)$ 和吞吐量趋势 $netut(H_i)$ 。资源分配算法描述如下:

```

FindBestHost()
(1) for every eligible host Hi
    1.1 收集并设置 u(Hi), ut(Hi);
    1.2 收集并设置 netu(Hi,j), netut(Hi,j)
(2) for every eligible host Hi
    2.1 Set netut(Hi) = ∞;
    2.2 For each heavy host Hj do
        2.2.1 If netu(Hi,j) < netu(Hj) then
            netu(Hi) = netu(Hj);
            netut(Hi) = netut(Hj);
(3) umax = max{u(Hi)};
    netumax = max{netu(Hi,j)};
(4) for every eligible host Hi
    4.1 uw(Hi) = 1.0 + atan(uttr(Hi) * w);
        / * w 为常数 * /
    4.2 netuw(Hi) = 1.0 + atan(netut(Hi,j) * (1-w));
        / * 利用 atan 函数规格化 ut 和 netut 至 [0, 1.57] * /
    4.3 BEST(Hi) = ([u(Hi)/umax] × uw(Hi) × whost + [netu(Hi,j)/netumax] × netuw(Hi) × wnet);
        / * whost 为计算资源权重, wnet 为网络资源权重 * /
(5) BEST(Hi) 最大的主机作为 BestHost 返回。
    
```

4 性能评测与分析

测试环境由 4 台 PC 服务器构成 (S0, S1, S2, S3), 操作系统为 LINUX9.2, 网络带宽为 100Mb/s。传感路径的时限设置为 4s。起始时传感模拟程序 (SM)、过滤模拟程序 (FM) 和决策模拟程序 (DM) 都运行在 S0 上, FM 的一个副本安装在 S1 上, DM 的副本安装在 S2 上。诊断程序 QDM 和资源管理程序 RM 运行在 S3 上。每台服务器上都运行了 QoS 监测程序 QM。图 3 为其中一个典型的测试场景。

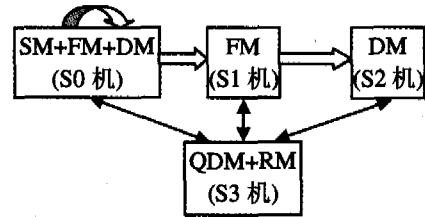


图 3 测试环境配置图

通过测量实验获得的公式系数如表 2。

表 2 预测函数公式(3)的系数

| 子路径 | a1 | a2 | a3 |
|-----|-----------|------------|------------|
| SM | -0.00155 | 1.535E-05 | 0.11816174 |
| FM | 0.002123 | -1.596E-05 | 0.022324 |
| DM | 0.001362 | 1.273E-05 | 0.047392 |
| 子路径 | b1 | b2 | b3 |
| SM | 0.0298276 | 0.000285 | 0.983699 |
| FM | 0.023927 | 0.000108 | 1.443762 |
| DM | 0.021843 | 0.000145 | 1.280479 |

对于 3 个子路径, 式(5)的系数 k 都为 0.7。

如图 4, 当试验开始时, 循环周期 $c \in [1, 23]$, 负载(传感目标数目)设为 2000, 从图 4 看出这时实测延迟小于期限。当 $c=23$ 时, 我们增加负载 500, 路径延迟也开始增加; 当 $c=55$ 时, 我们再次提高负载 500, 此时路径延迟逐步增加到超过 QoS 期限。从图 5, 6 可以看出 FM 和 DM 的延迟也开始随着负载的增加而增加。

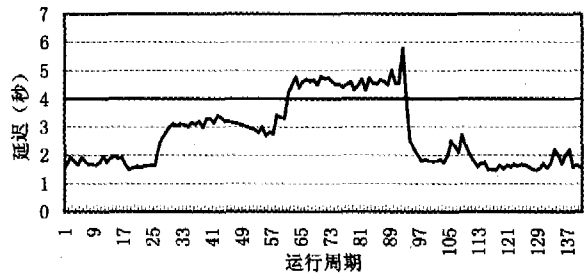


图 4 传感路径延迟

在周期段 [59, 79] 期间监测程序 QM 发现 QoS 超标, 在 $c=79$ 时交由 QDM 诊断可能的故障原因, 并推荐运行 FM 和 DM 的副本进程。随后由 RM 进行资源分配和启动副本进程。由于建立 SOCKET 通道需要花费几个周期的时间, 因此启动副本进程也会有相应的延迟。

从图 5 可以看到, 当 $c=92$ 时, 随着 FM 副本进程的启动, FM 两个进程的延迟都得到降低。同样, 图 6 中 DM 的两个进程在 $c=95$ 时也开始得到改善。

随着 FM 和 DM 副本进程的启动,传感路径的延迟也开始在 $c=95$ 时下降。

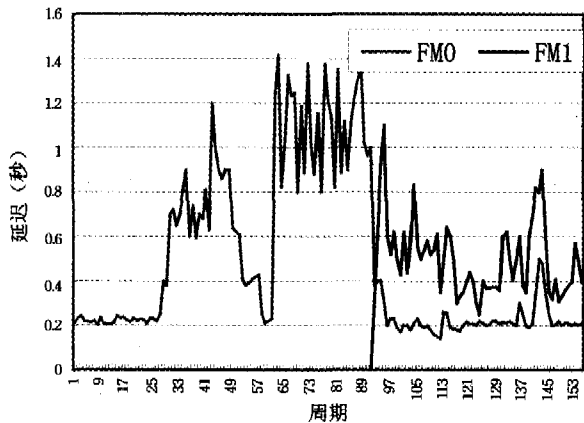


图 5 FM 的延迟变化

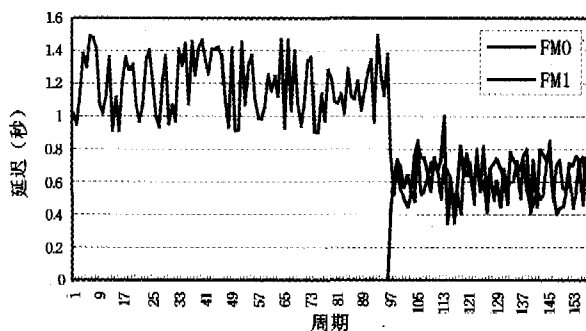


图 6 DM 的延迟变化

总结 天基网正日益得到许多国家的重视,因此天基网

的 QoS 研究有着重大意义。本文提出一个提供多类服务的基于路径的服务质量监控策略及对应算法,它以路径响应延迟为核心度量,将运行在天基网上的不同应用系统抽象为多个虚拟路径。本文分析了基于路径的实时系统 QoS 特性,研究了动态实时的 QoS 诊断、基于预测的副本创建和资源分配算法。实验表明,即使经历严重的过载与业务突发,也能保证用户感知的服务质量。

参考文献

- 1 Henderson T R, Katz R H. Network simulation for LEO satellite networks [C]. In: Proceedings of the 18th AIAA International Communications Satellite Systems Conference, 2001. 101~110
- 2 Abdelzaher T F, Atkins E M, Shin K G. QoS negotiation in real-time systems and its application to automated flight control [C]. In: Proceedings of the Third IEEE Real-Time Technology and Applications Symposium, 1997. 228~238
- 3 Scot B, Nuut G, Berk T, et al. Soft real-time application execution with dynamic quality of service assurance [C]. In: Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service, 1998. 154~163
- 4 Scot B, Nuut G, Berk T, et al. A dynamic quality of service middleware agent for mediating application resource usage [C]. In: Proceedings of the Nineteenth IEEE Real-Time Systems Symposium, 1998. 307~317
- 5 Scot B, Nuut G, Klingenstein K. A discrete and dynamic approach to application/operating system QoS resource management [C]. In: Proceedings of the First Internet2 Joint Application/Engineering QoS Workshop, 1998. 1~5
- 6 Mallikarjun S, DeMiguel M, Liu W S. An end-to-end QoS management architecture [C]. In: Proceedings of the Fifth Real-Time Technology and Applications Symposium, 1999. 23~29
- 7 Zengbin, Lizitang, Wangwei. Design for the simulation of Space Based Information Network [J]. To appear in Journal of Systems Engineering and Electronics, 2006
- 8 Abdelzaher T F, Atkins E M, Shin K G. QoS negotiation in real-time systems and its application to automated flight control [C]. In: Proceedings of the Third IEEE Real-Time Technology and Applications Symposium, 1997. 228~238

中国计算机学会暨电子政务与办公自动化专委会 全国电子政务技术及应用学术研讨会(EGTA2007)

征文通知

(2007年9月14—16日 中国人民大学 北京)

为促进我国电子政务建设,推动国内电子政务相关技术和应用研究成果的交流,中国计算机学会暨电子政务与办公自动化专委会定于2007年9月14—16日在北京中国人民大学召开全国电子政务技术与应用学术研讨会。现征文如下:(更详细情况见《计算机科学》网址:www.jsjkk.com)

一、征文范围(包括但不限于)

电子政务组网关键技术;电子政务网络可信互联关键技术;电子政务门户技术;电子政务业务流程优化重组;数据库关键技术;信息检索与数据挖掘技术; workflow 模型;电子政务应用支撑平台;XML 与半结构化数据管理;组件与中间件技术;决策支持与分析技术;电子政务信息安全保障;电子政务应用系统设计;新技术在电子政务中的应用;电子政务优秀产品和技

二、来稿要求

1. 本次会议只接受 Email 中文投稿。
2. 论文字数一般不超过 6000 字,为了便于出版论文集,来稿必须附中英文摘要、关键词、资助基金与主要参考文献,注明作者及主要联系人姓名、工作单位、详细通信地址(包括 Email 地址)与作者简介。稿件要求采用 WORD 格式。

三、联系信息

1. 投稿地址:国家信息中心信息化研究部 单志广(shanzhiguang@263.net)
2. 会务情况:中国人民大学 杨楠 姜芳芳(wisa2007@gmail.com)
3. 大会网站:<http://www.ruc.edu.cn/wisa2007/>, <http://www.neu.edu.cn/wisa2007/>

四、重要日期

征文截至日期:2007年4月1日 录用通知发出日期:2007年4月20日 正式论文提交日期:2007年5月10日