

一种结合 AADL 和 TPN 的系统能耗建模方法

马 铮, 黄连丽, 陶冰冰, 周海鹰

(湖北汽车工业学院 电气与信息工程学院, 湖北 十堰 442002)

摘 要:结构分析设计语言(AADL)可以描述嵌入式系统的实时性、可靠性等关键性能特性,是嵌入式领域的体系结构建模、分析和验证的重要手段。针对嵌入式系统能耗无法预知的问题,提出了一种基于 AADL 和 TPN(时间 Petri 网)的系统能耗建模与分析方法。该方法利用 AADL 基本语言建立了系统级架构模型,通过系统能耗相关性能分析,利用 AADL 属性集定义了能耗相关属性。由于 AADL 只是一种半形式化的语言,对系统的执行不够精确,而 Petri 网能较好地描述和模拟离散事件系统的动态行为及过程,为此,定义了 AADL 到 TPN 的模型转换规则,实现了 AADL 到时间 Petri 网的模型变换。通过对 TPN 模型进行严格的数学推导分析,得到系统整体能耗。为验证所提出的建模与分析方法,以某传感器节点为实例建立了系统能耗模型。实例验证表明,所提出的嵌入式系统能耗建模与分析方法可行有效。

关键词:结构分析设计语言;嵌入式系统;时间 Petri 网;能耗分析

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2017)06-0151-05

doi:10.3969/j.issn.1673-629X.2017.06.031

An Embedded System Energy Consumption Modeling Method Using AADL and TPN

MA Zheng, HUANG Lian-li, TAO Bing-bing, ZHOU Hai-ying

(School of Electrical & Information Engineering, Hubei University of Automotive Technology,
Shiyan 442002, China)

Abstract:Architecture Analysis and Design Language (AADL) can describe key performance features such as the real time and reliability of embedded system, which is an important means of architecture modeling, analysis and verification in embedded field. In view of the unpredictable energy consumption problem of the embedded system, a system modeling and energy consumption analysis method has been put forward based on AADL and TPN (Time Petri Net). It sets up system level structure model with AADL basic language, through analysis of system energy consumption related performance and definition of energy consumption related properties with AADL property set. Since the AADL is just a semi-formal language, the execution of the system is not accurate, but the Petri net can well describe and simulate the dynamic behavior processes of the discrete event system. Therefore, AADL-TPN model transformation rules has been defined, and the transformation between AADL model and time Petri nets model has been realized. The system energy consumption can be obtained through strict mathematical deduction and analysis. For the validation of the proposed modeling and analysis method, a sensor node has been chosen as an example and system energy model has been established. The operation of the example shows that the proposed system modeling and energy consumption analysis method are both feasible and effective.

Key words:AADL; embedded systems; TPN; energy consumption analysis

0 引言

目前,嵌入式系统在航空航天、智能交通、工业控制、医疗卫生等领域应用广泛,嵌入式设备每年电力消耗达到 1 000 亿 kw·h 以上^[1]。在全球提倡“节能减排”的背景下,如何降低嵌入式系统的软、硬件能耗已成为嵌入式系统设计的热点问题^[2]。

传统的能量评估技术主要从软件^[3]和硬件两方面入手,然而仅从硬件或软件着手分析系统的能耗是不完善的。嵌入式实时系统体系结构分析与设计语言(AADL)^[4]可以描述复杂嵌入式系统软硬件的功能属性及可靠性、安全性、可调度性等非功能属性,在航空航天、汽车电子等复杂嵌入式系统中应用广泛。因此,

收稿日期:2016-07-06

修回日期:2016-10-19

网络出版时间:2017-04-28

基金项目:湖北省科技支撑计划项目(对外合作)(2014BHE024)

作者简介:马 铮(1992-),女,硕士研究生,研究方向为汽车电子、嵌入式系统设计;周海鹰,教授,研究方向为嵌入式系统、云计算。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170428.1703.052.html>

利用 AADL 建立系统的软硬件能耗模型并进行预测分析将会是一重大突破。

在 AADL 能耗相关的研究中, E. Senn 等^[5-6] 基于 AADL 以及 CAT 工具, 给出一种进程间通信 (IPC) 电量和能源消耗的建模方法, 并通过建立以太网进程间通信模型进行验证。罗增等^[7] 针对 AADL 行为附件定义了能耗元素, 通过元素与处理器虚拟指令之间的映射关系并借助处理器 Strong ARM 的虚拟指令能耗模型, 构建出一种面向 AADL 语言的移动软件能耗评估模型。上述方法仅考虑了软件部分的能耗, 而在实际应用中硬件部分能耗是不可忽略的。综上所述, AADL 不仅可以作为模型描述语言还可借助其他工具对其能耗进行分析。

为此, 基于 AADL 嵌入式系统软硬件的综合考虑进行建模, 由于 AADL 属于高层半形式化语言, 需要将 AADL 模型转换为更底层的形式化模型, 而时间 Petri 网 (TPN)^[8-9] 可以描述和模拟离散事件系统的动态行为及过程。综合 AADL 和时间 Petri 网, 根据系统需求分析建立了 AADL 的系统能耗模型, 在利用 AADL 建模时主要用到了 OSATE2 建模工具; 在定义了转换规则后, 利用 Ocarina^[10] 工具实现了 AADL 模型到时间 Petri 网的转换。基于 TPN 模型, 以某传感器节点为实例进行了节点能耗仿真验证。结果表明, 所提出的方法可行有效。

1 AADL 和 TPN 概述

1.1 AADL

2004 年, 美国汽车工程师协会 (Society of Automotive Engineers, SAE) 在 MetaH 和 UML 的基础上, 提出了 AADL^[11]。AADL 以组件为核心, 将系统软件及硬件都看作组件的集合, 将整个系统描述为一组交互组件的集合。AADL 定义了三种组件: 软件组件、执行平台组件以及系统组件。软件组件包括数据、线程、线程组、子程序、进程、进程组; 硬件组件包括处理器、设备、存储器、总线。各组件定义为: 类型和实现并且可以使用关键词 extends 进行继承; 组件类型描述输入输出接口, 组件实现可详细描述内部结构 (子构件、连接、属性值等), 一个组件类型可以有一个或多个实现。各个组件最终集成于系统实现中, 可在系统实现中将软件绑定硬件中, 使软件正确执行。

1.2 时间 Petri 网简介

Petri^[12] 网具有强大的数学定义、严格规范的推导方法和直观的图形描述能力, 非常适合描述系统的控制流、并发特性和异步行为。TPN 通过在基本形式 Petri 网的变迁节点或位置节点中引入表征时间的参量, 不仅能描述系统在逻辑层次上的关系, 而且能够表征

系统在时间层次上的关系。结合 AADL 线程的时间属性, 给出时间 Petri 网的定义。

定义 1: 时间 Petri 网 $TPN = (P, T, F, K, M_0, W, D)$ 是一个一元组, 其中: $P = \{p_1, p_2, \dots, p_n\} (n \geq 0)$ 是一个有限库所集; $T = \{t_1, t_2, \dots, t_m\}$ 是一个有限变迁集, $T = T_{\text{immediation}} \cup T_{\text{delay}}$ 且 $T_{\text{immediation}} \cap T_{\text{delay}} = \emptyset$, $T_{\text{immediation}}$ 是所有立即事件或变迁集, T_{delay} 是延时变迁; $F \subseteq (P \times T) \cup (T \times P)$ 是所有状态间转移弧的集合; K 是 P 上的容量函数; M_0 是初始状态集或满足条件的状态集; W 是弧的容量函数; $D = \{D_1, D_2, \dots, D_M\}$ 是一组 T_{delay} 上相联系的时延参量。

定义 2: TPN 在 t 中使能的定义:

$$\{M(p_i) \geq W(p_i, t)\} \cap \{M(p_j) + W(t, p_j) \leq K(p_j)\}, \forall p_i \in {}^*t, \forall p_j \in t^*$$

定义 3: 立即变迁 ($T_{\text{immediation}}$) 即当某一变迁的发生条件满足时, 该变迁立即从相应的输入库所中移走相应的托肯, 且得到发生后果。定义如下:

当满足定义 2^[13], 则:

$$M(p) = \begin{cases} M(p) + W(t, p), & p \in t^* - {}^*t \\ M(p) - W(p, t), & p \in {}^*t - t^* \\ M(p) - W(p, t) + W(t, p), & p \in {}^*t \cap t^* \\ w(p), & \text{other} \end{cases}$$

根据 AADL 中相关的时间属性, 将延时分为区域值延时与固定延时, 其中固定延时是区域值延时的一种特殊情况。对于任意的时延变迁 $t \in T_{\text{delay}}$, 都有一个对偶区间 $[T_{\min}, T_{\max}]$ 与之相对应。当变迁 t 在时刻 b 的发生条件成立时, 则 t 可在时间区间 $[b + T_{\min}, b + T_{\max}]$ 内执行, 其相应的输入库所中的托肯将至少保留 T_{\min} , 最大保留 T_{\max} ; 若超过该时间, 则认为变迁不能发生了; 若 $T_{\min} = T_{\max}$, 则为固定值延时。

定义 4: T_{delay} (延时变迁), 即当某一个变迁的发生条件满足时, 该变迁要延迟一段时间才从相应的输入库所中移走相应的 token 并得到发生后果。在此, 不考虑延时变迁, 即认为一旦可执行性条件满足时立即发生。

2 基于 AADL 的系统能耗建模

基于 AADL 对系统能耗进行建模和分析, 图 1 是一个完整的基于 AADL 的系统能耗与评估框架, 其流程如下: 首先对系统软硬件进行需求分析, 建立 AADL 系统模型; 然后定义 AADL 模型到时间 Petri 网模型的转换规则, 将 AADL 模型转换为 TPN 模型; 接着利用已有的 TPN 模型的分析方法和工具 Tina, 对转换生成的 TPN 模型进行仿真计算和分析。

以某传感器节点为例, 图 2 给出了 AADL 的传感器节点模型架构方框图。

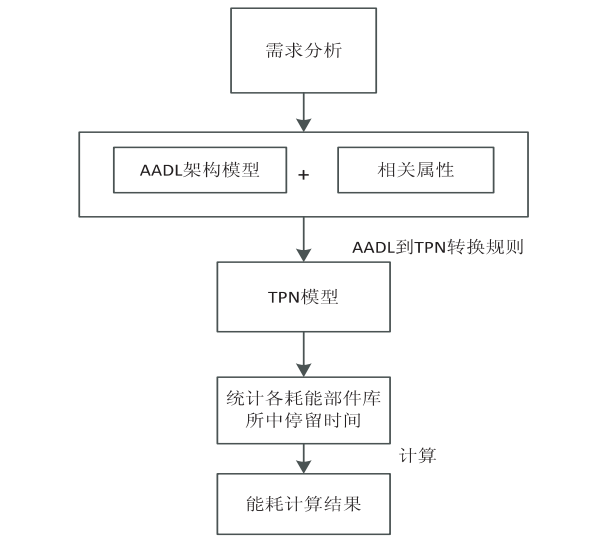


图1 系统能耗评估框架

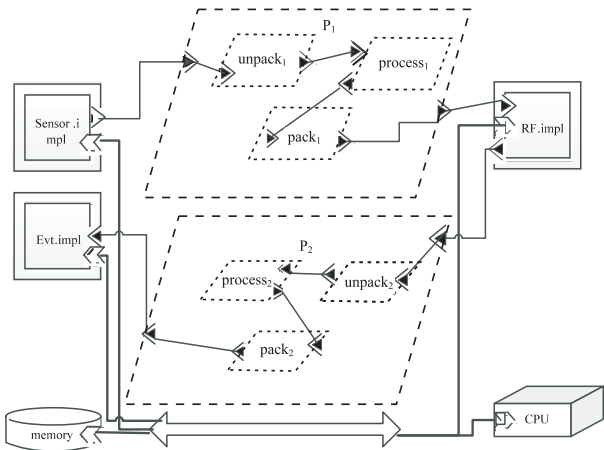


图2 传感器节点的 AADL 模型方框图

该系统拥有两个进程 P_1 和 P_2 , 进程 P_1 处理传感器采集数据, 进程 P_2 处理外部接收数据。进程 P_1 包含三个线程: 线程 $unpack_1$ 拆数据包; 线程 $process_1$ 分析处理数据包; 线程 $pack_1$ 压缩封装数据包。三个线程具有相同的地址空间, 分别绑定在 CPU 上, 进程 P_2 同样拥有解包、处理、封装三个线程。传感器节点架构所对应的 AADL 系统代码如下:

```
system implementation s_ss.impl
subcomponents
s_sensor:device sensor.impl;
p1:process p_sample.impl;
s_rf :device rf.impl;
cpu:processor cpu.impl;
s_rom:memory memory_1.impl;
bus_1:bus bus_1.impl;
p2:process p_receve.impl;
s1:device evt.impl;
connections
cn1:port p2.resout->s1.evt_data;
.....
bn5:bus access bus_1->s_rf.bal;
```

```
properties
Actual_Processor_Binding=>(reference(cpu)) applies to p1.th_u,p1.th_p,p1.th_pro,p2.th_u,p2.th_p,p2.th_pro;
Actual_Memory_Binding=>(reference(s_rom)) applies to p1,p2;
end s_ss.impl;
```

线程 $unpack$ 、 $process$ 、 $pack$ 之间通过数据、事件端口连接。数据、事件存储于 $memory$ 中, 各设备 (Sensor、RF) 以及处理器 (CPU)、存储器 (Mem) 通过总线 (Bus) 互联。线程部分代码实现如下:

```
thread implementation processing.impl
calls main;{
opg:subprogram handle;};
connections
cn1:parameter RF_data->opg.d_handle;
properties
Dispatch_Protocol=>periodic;
Period=>50ms;
Initialize_Deadline=>50ms;
compute_deadline=>50ms;
Reference_Processor=>classifier(s_0::cpu);
end processing.impl;
```

定义线程 $processing$ 为周期性线程, 周期 50 ms ; $compute_deadline$ 为 50 ms ; $initialize_deadline$ 为 50 ms 。该线程描述了当有数据或任务到来时, 激发处理器开始进行拆包处理。

3 AADL 到 TPN 的转换及计算方法

AADL 属于高层建模语言, 不可直接用于能耗计算和分析, 因此首先需要将 AADL 模型转换到更低层的形式化模型, 进而再进行计算和分析。TPN 具有直观、形象等优点, 能够以图形化的方式形象地描述一个复杂系统及其动态行为, 且具有许多优良的数学性质, 适用于描述及分析异步并发系统。TPN 的结构元素包括库所 (place)、变迁 (transition) 和弧 (arc) 等。鉴于 AADL 组件与 TPN 元素间存在的对应关系, 定义 AADL 到 TPN 的转换规则成为节点建模分析的首要任务。

3.1 AADL 到 TPN 的转换规则

AADL 行为主要由各组件数据、事件端口之间的交互进行表征。由于线程具备行为描述能力, 因此软件组件转换主要专注于线程转换^[10]。AADL 模型基本元素和各组件状态之间的变迁可以用有限自动机模型描述, 该自动机模型和 TPN 模型之间存在映射关系, 表征如下:

定义 5: 用有限自动机描述的 AADL 模型表示为 $AM = (AS, AE, TR, IAS, F, T)$, 其中: AS 是所有组件

状态的集合, $AS = (as_1, as_2, \dots, as_n)$; AE 是所有事件、动作的集合, $AE = AE_i \cup AE_d$ 且 $AE_i \cap AE_d = \emptyset$, AE_i 是所有立即事件、动作集, AE_d 是所有延迟、周期事件动作集; TR 是所有状态间转移的集合, 转移函数 $TR(as_i, at_j) = as_k$; IAS 是初始状态集或满足条件的状态集; F 是 AM 的终止状态集合; T 是事件相关的时间属性, 包括 `period`、`Compute_Deadline`、`Compute_Execution_Time` 等。

根据定义 5 和 TPN 的定义, 可实现 AADL 模型基本元素到 TPN 模型所对应的函数 TPN_AM 的映射, 表征如下:

定义 6: AADL 基本模型元素到 TPN 模型元素转换:

(1) $TPN_AM(as) = p, TPN_AM(AS) = P$ 实现

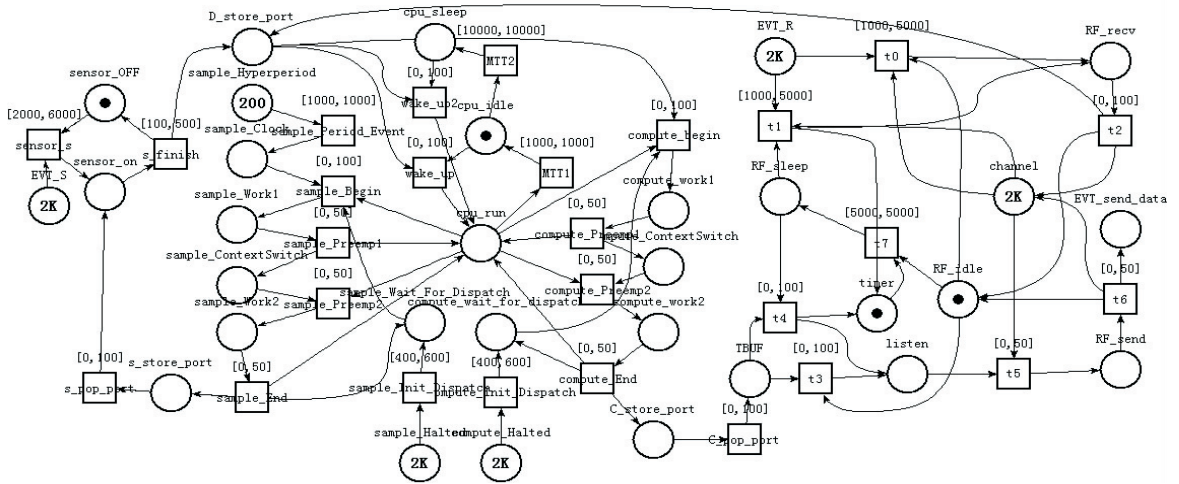


图 3 系统转换图

3.2 系统能耗计算方法

假设如下:

(1) 节点总能耗 E_{total} 为各个能耗部件 E_{sensor} 、 E_{cpu} 、 E_{rf} 之和^[14-15]:

$$E_{total} = E_{sensor} + E_{cpu} + E_{rf} \quad (1)$$

(2) 各个组件能耗为该组件各个状态下能耗之和, 对于 CPU, 如下所示:

$$E_{cpu} = E_{cpu_run} + E_{cpu_idle} + E_{cpu_sleep} \quad (2)$$

对其他耗能部件, 能耗计算方法式(2)类似。

(3) 在整个 TPN 模型中, 存在多个库所和变迁, 只考虑对该状态有影响的库所和变迁。

(4) 在 token 到达某库所且其他条件都满足时, 变迁才可发生; 一旦 firing 之后, token 到达下一个库所为瞬时的。

(5) 对于某一库所, 从 token 流入到流出时即为此次变迁在该库所停留时间 t_i ; 对应的实际含义为此次执行在该状态的消耗时间。在仿真一定时间后, 每一库所消耗的总时间, 即为该状态在这段时间内消耗

AADL 模型各组件状态到 TPN 中库所之间的转换。

(2) $TPN_AM(at) = t, TPN_AM(AT_i) = T_i$, $TPN_AM(AT_d) = T_d$, 即 AADL 模型中立即事件、动作到 TPN 的瞬时变迁, AADL 中延时事件、动作到 TPN 中的延时变迁。

(3) $TPN_AM(TR) = A_{st} \cup A_{ts} = F$, 将 AADL 中状态之间的转换对应于 TPN 中库所到变迁的弧和变迁到库所的弧。

(4) $TPN_AM(ias) = p$, 是包含 token 的库所, 将初始状态 ias 元素转换到 TPN 中只含 token 的库所。

(5) $TPN_{AM(t)} = (t_{min}, t_{max})$, $TPN_{AM(T)} = D$ 将 AADL 中时间属性转换为 TPN 的时间延迟。

通过上述规则, 可以得出 AADL 各组件的 TPN 模型。以传感器节点为例, 系统转换如图 3 所示。

的总时间 T_{s_i} , 该状态的功率为 p_{s_i} , 则该状态的耗能为:

$$E_{s_i} = T_{s_i} * p_{s_i} \quad (3)$$

$$T_{s_i} = \sum_i t_i \quad (4)$$

(6) 对于 CPU, 在运行时有六个任务, 分别用来对传感器采集的数据和 RF 接收数据进行拆包、处理、封装。能耗计算公式为:

$$E_{cpu_run} = \sum_{i=1}^6 E_{cpu_task_i} \quad (5)$$

至此, 结合 AADL 和 TPN 的能耗建模已完成, 通过 Tina 工具可对节点 TPN 模型进行仿真, 通过计算出各个耗能部件所有状态中 Token 所停留的时间, 即可计算出整个系统在一定时间内的能耗情况。

4 结束语

为了能及时预测分析嵌入式系统整体能耗, 提出了一种结合 AADL 和 TPN 的嵌入式系统能耗建模方

法。该方法综合考虑系统软硬件,建立了能耗模型,并通过严格的数学推导公式预测整个系统的能耗。以某传感器节点为实例的验证结果表明,所提出的方法有效、可行。该方法在嵌入式系统设计初期对系统架构的能耗分析具有很好的参考价值,方便设计人员根据能耗要求及时调整系统架构,节省设计的时间、物力成本。

参考文献:

[1] 郭 兵,沈 艳,邵子立. 绿色计算的重定义与若干探讨 [J]. 计算机学报,2009,32(12):2311-2319.

[2] 赵 霞,郭 耀,陈向群. 软件能耗优化技术研究进展[J]. 计算机研究与发展,2011,48(12):2308-2316.

[3] Tiwari V, Maliks S, Solfe A. Power analysis of embedded software: a first step towards software power minimization [J]. IEEE Transactions on Very Large Scale Integration, 1994, 2(4):437-444.

[4] 杨志斌,皮 磊,胡 凯,等. 复杂嵌入式实时系统体系结构设计与分析语言: AADL[J]. 软件学报,2010,21(5):899-915.

[5] Blouin D, Senn E. CAT: an extensible system-level power consumption analysis toolbox for model-driven [C]//Newcas conference. [s. l.]:[s. n.], 2010:33-36.

[6] Senn E, Douhib S, Blouin D, et al. Power and energy estima-

tions in model-based design [M]. Netherlands: Springer, 2009:3-26.

[7] 罗 增. 一种基于 AADL 语言的移动软件能耗评估方法 [D]. 福州:福建师范大学,2015.

[8] Wu Zhehui. Petrinetwork introduction [M]. Beijing: Mechanical Industry Press, 2006:47-64.

[9] Zhang Hongmei, Liu Fei, Yang Ming, et al. Simulation of time petri nets [C]//Fourth international conference on system science, engineering design and manufacturing. Guiyang: [s. n.], 2013:26-27.

[10] Hugues J, Zalila B, Pautet L, et al. From the prototype to the final embedded system using the Ocarina AADL tool suite [J]. ACM Transactions on Embedded Computing Systems, 2008, 7(4):1-25.

[11] Feiler P H, Gluch D P. Model-based engineering with AADL [M]. [s. l.]:[s. n.], 2012.

[12] Girault C, Valk R. Petri nets for systems engineering [M]. [s. l.]:Springer-Verlag, 2003.

[13] 吴育春,李蜀瑜. 基于时间 Petri 网的 AADL 模型[J]. 计算机技术与发展, 2014, 24(2):88-91.

[14] 谢和平. 基于排队 Petri 网的感知节点能耗建模技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2013.

[15] 周海鹰,徐 杰,高 妍,等. 基于状态转移的感知节点能耗模型研究与设计 [J]. 计算机应用研究, 2012, 29(9):3432-3436.

(上接第 150 页)

较大规模的共享数据的相关问题,以在 Hadoop 集群中引入 Redis 分布式缓存的方式,为该类问题提供了一种简单、高效的解决方案。实验结果表明,Redis 分布式缓存存在高并发访问时仍具有优异的性能,同时还具有扩展性好、使用简单等特点,这些使得该方案能够很好地与实践相结合,解决 Hadoop 任务中共享数据的访问问题。

参考文献:

[1] Viktor M S, Cukier K. 大数据时代 [M]. 杭州:浙江人民出版社, 2013.

[2] 严霄凤,张德馨. 大数据研究 [J]. 计算机技术与发展, 2013, 23(4):168-172.

[3] 王彦明, 奉国和, 薛 云. 近年来 Hadoop 国外研究综述 [J]. 计算机系统应用, 2013, 22(6):1-5.

[4] 杜 江,张 铮,张杰鑫,等. MapReduce 并行编程模型研究综述 [J]. 计算机科学, 2015, 42(6A):537-541.

[5] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51

(1):107-113.

[6] Redis [EB/OL]. 2016-01-28. <http://redis.io>.

[7] How fast is Redis [EB/OL]. 2013-08-20. <http://redis.io/topics/benchmarks>.

[8] Redis cluster specification [EB/OL]. 2014-10-09. <http://redis.io/topics/cluster-spec>.

[9] Rai P, Lal A. Google PageRank algorithm: Markov chain model and hidden Markov model [J]. International Journal of Computer Applications, 2016, 138(9):9-13.

[10] 李远方,邓世昆,闻玉彪,等. Hadoop-MapReduce 下的 PageRank 矩阵分块算法 [J]. 计算机技术与发展, 2011, 21(8):6-9.

[11] Och F J, Jahr M E, Thayer I E. Minimum error rate training with a large number of features for machine learning: US, 2013/0144593 A1 [P]. 2013-06-06.

[12] 胡爱娜,蔡晓艳. 基于 MapReduce 的分布式期望最大化算法 [J]. 科学技术与工程, 2013, 13(16):4603-4606.

[13] 曾超宇,李金香. Redis 在高速缓存系统中的应用 [J]. 微型机与应用, 2013, 32(12):11-13.

[14] Leskovec J, Rajaraman A. Mining of massive datasets [M]. Cambridge: Cambridge University Press, 2014.