第 55 卷 第 8 期 2006 年 8 月
1000-3290/2006/55(08)/3799-06

物 理 学 报
ACTA PHYSICA SINICA

Vol.55,No.8,August,2006

大规模软件系统的无标度特性与演化模型*

闫 栋 祁国宁

(浙江大学现代制造工程研究所,杭州 310027)

(2005 年 10 月 9 日收到;2006 年 4 月 10 日收到修改稿)

在软件工程中,常用类图来描述类之间的关系. 以软件系统网为研究对象,通过对 Sun 和 IBM 公司提供的大规模软件系统进行实证分析,发现 Java 软件系统网的度分布是无标度分布,标度指数 $\gamma \approx 2.5$. 在软件系统网的演化过程中,除加点之外,还存在边的添加、边的随机移除与边的重连等局部事件. 由此建立了软件系统演化模型. 由该模型演化生成的网络,其度分布服从幂律分布. 实际应用与数值仿真验证了该模型的有效性.

关键词:软件系统,复杂网络,度分布,无标度

PACC:0250,0565

1. 引 言

在这个复杂世界中,复杂系统蕴含着许多人类已知或未知的现象和规律. 这些现象和规律吸引并激励着人们对各种系统进行不懈的探索与对真理的追求. 通过对许多真实系统的实证分析与研究,人们发现了许多控制系统演化的机制. 1960 年,Erdós 和 Rényi[1] 基于随机机制提出了著名的随机图模型,在该模型中,节点(组成单元)以等概率成对连接形成网络. 随机图模型揭示了现实世界中具有随机特性的各种系统的随机原理. 1998 年,Watts 和 Strogatz[2] 提出了一种新的网络模型——小世界模型,该模型可以模拟人际关系演化过程. 然而,在现实世界中,还存在着许多困惑人们的"富者愈富"现象. 在研究规模不断膨胀的万维网(World Wide Web)的过程中,Barabási 和 Albert[3,4] 首先观测到网络中的幂律度分布特性,并基于增长与择优连接机制提出了著名的无标度模型. 由该模型演化生成的网络,其度分布 $p(k)$ 服从衰减幂律分布,即 $p(k) \propto k^{-\gamma}$,其中标度指数 $\gamma > 0$,k 表示结点的度. 择优连接指一个节点被连接的概率与其度成正比. 无标度模型很好地解释了"富者愈富"现象. 随后,Albert 和 Barabási[4,5] 发现,在许多系统中还存在边的单独添加和重连等局部作用,由此提出了局部事件(local-event)无标度模型,而上述无标度模型是局部事件模型的一个特例. 许多研究者也相继进行了更多系统的实证分析和网络特性研究[6—8]. 如 Myers[9] 分析了多个 C/C++ 软件系统,发现其度分布具有无标度特性. 其他被研究过的著名网络包括国际互联网[3,6]、电影演员合作网、科研合作网[7]、人类性接触网、细胞网、生态网、电话网、语言学网、电力网、神经网与蛋白质网等[4].

本文将分析并研究另一种大规模软件系统——Java 系统的复杂网络特性并给出一个演化模型.

2. 局部事件无标度模型

局部事件无标度模型模拟了在网络中现有节点间添加新连接(边)的作用、现有边的重新连接的作用以及带有新边的新节点与现有节点之间的连接作用.

局部事件无标度模型开始于 m_0 个孤立节点,在每个时间步骤都会发生下述三个过程之一.

过程 Ⅰ 以概率 $p(0 \leqslant p < 1)$ 增添 $m(m \leqslant m_0)$ 条新边. 新边的一端与随机选择的节点相连,另一端与择优选择的节点相连,择优连接概率为

$$\Pi(k_i) = \frac{k_i + 1}{\sum_j (k_j + 1)}. \tag{1}$$

重复过程 Ⅰ m 次.

* 国家自然科学基金(批准号:60374057,50575204)资助的课题.

过程Ⅱ 以概率 $q(0\leqslant q<1-p)$ 重连网络中已有的 m 条边. 随机选择网络中的某个节点 i 以及与该节点相连的任一条边 l_{ij},根据择优连接概率(1)式选择节点 j',然后用一条新边 $l_{ij'}$ 代替边 l_{ij},$l_{ij'}$ 连接节点 i 与 j'. 重复过程Ⅱ m 次.

过程Ⅲ 以概率 $1-p-q$ 添加一个新节点,该新节点有 m 条新边. 各条新边的另一端依择优连接概率(1)式连接到已存在于系统中的节点 i 上. 重复过程Ⅲ m 次.

根据 Barabási-Albert 连续域理论(continuum theory),局部事件无标度模型的度分布具有幂律形式,

$$p(k)\propto[k+K(p,q,m)]^{-\gamma(p,q,m)}, \tag{2}$$

式中,

$$K(p,q,m)=(p-q)\left(\frac{2m(1-q)}{1-p-q}+1\right)+1, \tag{3a}$$

$$\gamma(p,q,m)=\frac{2m(1-q)+1-p-q}{m}+1. \tag{3b}$$

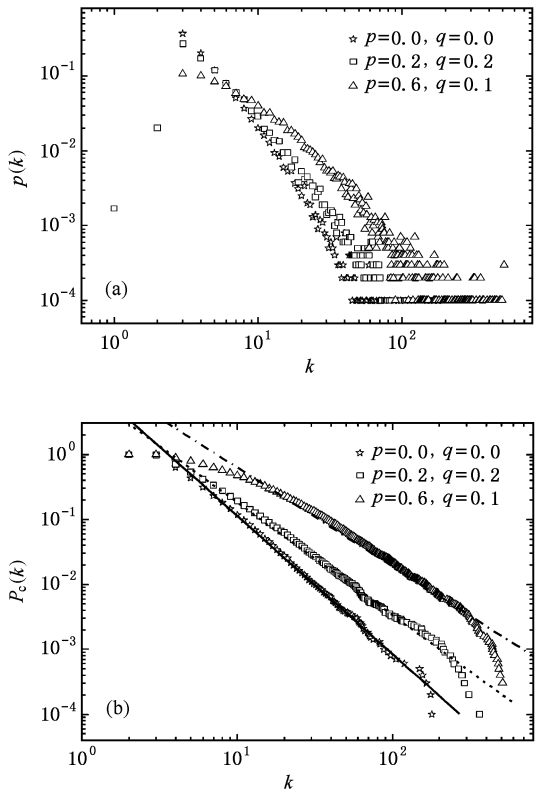

图 1 局部事件无标度模型的仿真结果 (a)度分布的双对数坐标图,(b)累积度分布的双对数坐标图

描述网络分布特性的另一种形式是采用累积度分布. 累积度分布 $P_c(k)$ 是指网络中节点度大于或等于 k 的概率,即

$$P_c(k)=\int_k^{+\infty}p(k)\,\mathrm{d}k.$$

累积度分布可以控制统计数据的噪声问题[10].

对局部事件无标度模型进行了数值仿真,仿真结果呈幂律形式(图 1). 图 1 中仿真网络的参数均有结点总数 $N=10000$,$m_0=10$,$m=3$.

3. Java 软件系统结构简介

计算机技术的发展,使得软件系统变得越来越庞大. 提供网络应用解决方案的 Sun 公司的 Java 语言是近几年发展较快的面向对象编程语言,它可以构建与平台无关的应用系统. 每个系统主要由大量的类及其类之间的相互作用构成. 在软件工程中,类之间的相互作用分为关联、泛化、实现和依赖四大类. 类之间的相互作用使得系统具有协同与自组织的网络特性. 类图(class diagram)常被用来描述类之间的各种相互作用关系,如图 2 实例所示. 在统一建模语言(UML)[11]中,定义一个类图就是一个静态声明模型元素(类与接口及其相互作用关系)的集合. UML 对软件系统架构的思想首先来源于分布式理念. 分布式理念倡导将一个软件系统要实现的功能逐步细化并分工成许多小的功能与函数,以此通过小的功能与函数的协同作用实现各种大的功能,即将一个大系统构建成一个具有大量基本组成单元并包含各种相互作用关系的网络. 分布式架构使得软件系统的结构清晰,并有利于系统的优化、升级和成长.

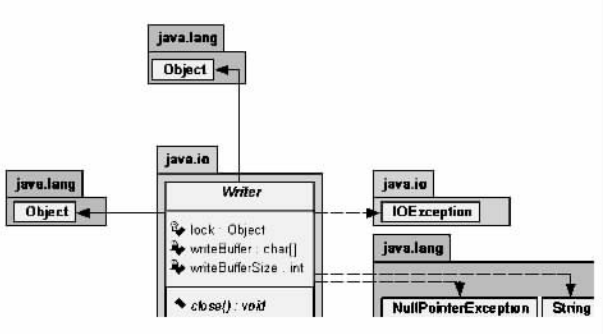

图 2 由 Java 的一些类生成的类图实例

为了尽量减少软件工程师的重复劳动,代码重用技术也被对象管理组织(OMG)以及软件工程师所提倡,并被广泛应用. 代码重用是指对具有所需功能的已有代码直接引用,而不需再次编写,尤其是对那些基本的常用代码和优秀代码而言. 因此,代码

重用有利于节约软件工程师的工作时间并提高信息业的效率，最终有利于企业降低软件开发成本.

如果将一个软件系统中所有类图组合起来，就形成了一个复杂网络. 将软件系统中的每个类看作一个节点，类之间的相互作用关系看作连接节点的边，节点之间不计重复连接. 每个类的度就是该类所有边的条数. 度的频数就是系统中具有相同度的节点数. 度的概率就是具有相同度的节点数占总节点数的比率. 那么，根据度的概率便可以获得度的概率分布图.

4. 实证分析

我们统计分析了 Sun 公司所有版本的 Java 开发包 JDK 系统以及 IBM 公司的 Netbeans 3.6 与 Netbeans 4.0. 实证表明这些软件系统的度分布都服从衰减幂律分布. 下面仅给出一个代表性实例——Sun JDK1.4.0(3883 个类/节点)的网络度分布图，如图 3 所示. 在图 3(a)的线性坐标中，度分布呈 L 形；而在图 3(b)的双对数坐标中，度分布具有肥大的尾部. 从图 3(a)可以看出，在线性坐标中度分布图的拐角区应存在一个类似拐点的贫富分界点. 在贫富分界点的左侧，度分布 $p(k)$ 随 k 的增大而出现较大幅度的下降，而其分布曲线表现为与纵坐标几乎平行. 在贫富分界点的右侧，度分布 $p(k)$ 随 k 的增大而下降幅度较小，而其对应曲线表现为与横坐标几乎平行. 图 3(a)在贫富分界点的右侧具有较大度的节点表现出‘富者愈富’现象. 相反，在贫富分界点的左侧具有较小度的节点表现出‘穷者愈穷’现象. 图 3(b)表明，在双对数坐标中度分布图的尾部随概率 $p(k)$ 的减小而越来越肥大.

为了获得度分布的线性图形以利于直线拟合，我们给出双对数坐标的累积度分布，如图 3(c)所示. 图 3(c)表明，只有极少量的节点具有较大的度，而大部分节点的度都较小，这就是‘富者愈富’现象. 如在 JDK1.4.0 中度大于 25 的节点数只占总数的 10%，而其最大度远大于 25. 度较大的少数节点对应于软件系统中那些最常用的类，这些类通常被软件工程师大量重用(代码重用). 所以，在软件系统中‘富者愈富’现象归因于代码重用技术. 在双对数坐标中，累积分布曲线近似于一条直线，这显示了度分布的幂律特性. 在实际的软件系统架构过程中，软件工程师基于分布式理念添加类(加点)并基于代码重用(加边)技术实现对系统的升级. 在升级过程中，还可能伴随着对个别已有类的优化，从而在网络上引起边的添加、移除和重连. 根据统计，Java 软件系统如 JDK 等随着版本的升级，其规模(如节点数与边数)也变大. 这表明该类系统是一个增长网络. 因此，大规模 Java 软件系统是一个无标度网络.

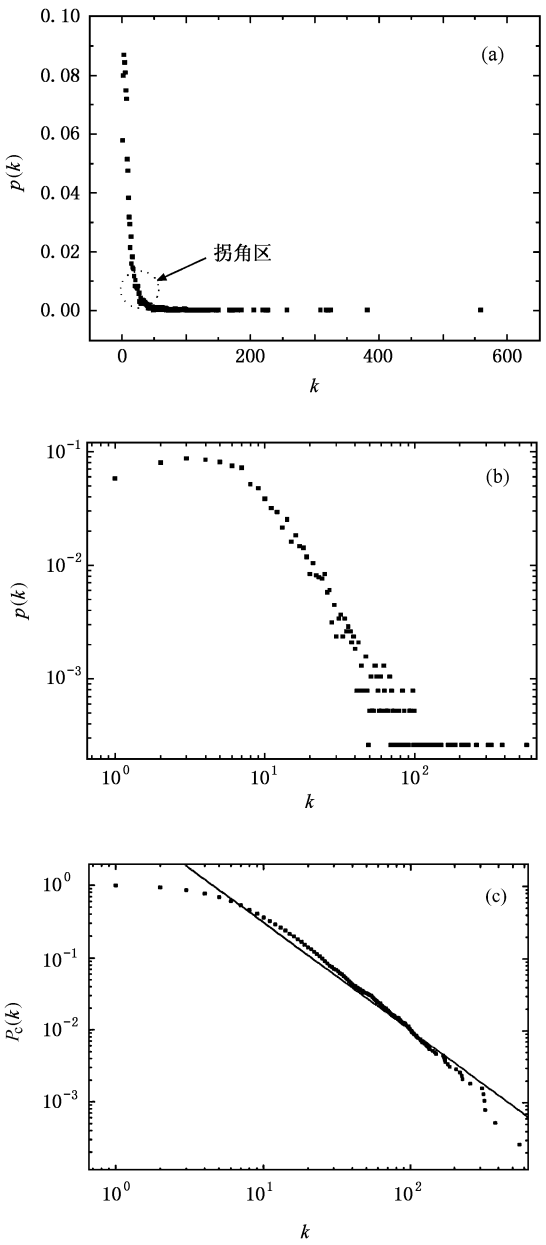

图 3　Sun JDK1.4.0 的度分布图与累积度分布图　(a) Sun JDK1.4.0 的度分布线性坐标图，(b) Sun JDK1.4.0 的度分布双对数坐标图，(c) Sun JDK1.4.0 的累积度分布双对数坐标图

通过对累积度分布图进行直线拟合(图 3(c))，可以得到大规模软件系统的网络度分布的标度指数 γ. 用同样的方法，对以上提到的所有系统进行统计

与分析. 根据统计,大规模软件系统度分布的标度指数 $\gamma \approx 2.5$,其中 JDK1.4.0 的标度指数为 $\gamma = 2.49$.

5. 软件系统演化模型

既然大规模软件系统的度分布服从无标度分布,那么在系统演化过程中节点的添加与边的择优连接必然发挥着作用. 此外,在大规模软件系统中还存在如下现象 边的添加、边的随机移除和边的重连. 由此,基于局部事件无标度模型给出以下的软件系统演化模型.

软件系统演化模型开始于 m_0 个节点,e_0 条边. 在每个时间步骤依概率执行下述过程之一.

过程Ⅰ 以概率 p 增添 m_1 条新边. 每条新边的一端与随机选择的节点相连,另一端与择优选择的节点相连,择优连接概率为

$$\Pi(k_i) = \frac{k_i + \alpha}{\sum_j (k_j + \alpha)}, \tag{4}$$

式中 α 代表初始吸引度. 重复过程Ⅰ m_1 次. 如果 $\alpha = 0$ 且 $k_i = 0$,则 $\Pi(k_i) = 0$,孤立节点 i 永远不能增加连通度.

过程Ⅱ 以概率 q 重连网络中已有的 m_2 条边. 随机选择网络中的某个节点 i 以及连接该节点的任一条边 l_{ij},根据择优连接概率(4)式选择节点 j',然后用一条新边 $l_{ij'}$ 代替边 l_{ij},$l_{ij'}$ 连接节点 i 与 j'. 重复过程Ⅱ m_2 次.

过程Ⅲ 以概率 r 随机移除网络中已有的 m_3 条边.

过程Ⅳ 以概率 s 添加一个新节点,该新节点有 m_4 条新边. 依择优连接概率(4)式选择节点 i,并与新节点一新边的另一自由端相连. 重复过程Ⅳ m_4 次.

在软件系统演化模型中,概率 p, q, r, s 必须满足条件 $0 \leqslant p, q, r, < 1, 0 < s \leqslant 1$ 以及 $p + q + r + s = 1$,而边数 m_1, m_2, m_3 与 m_4 分别满足条件 $m_1, m_4 \leqslant m_0$ 以及 $m_2, m_3 \leqslant m_4$. $s > 0$ 保证该模型产生的网络是一个增长网络.

根据 Barabási-Albert 连续域理论,设定 k_i 连续变化,并且概率 $\Pi(k_i)$ 随 k_i 的变化而变化. 采用连续域理论对过程Ⅰ—过程Ⅳ分别进行求解.

当以概率 p 添加 m_1 条新边时,k_i 的变化率方程为

$$\left(\frac{\partial k_i}{\partial t}\right)_{\mathrm{I}} = pm_1 \frac{1}{N(t)} + pm_1 \Pi(k_i). \tag{5}$$

在(5)式中,$N(t)$ 表示系统的当前节点总数. (5)式等号右端第一项表示新边的一端按随机选择进行连接,第二项表示新边的另一端按择优选择(4)式进行连接.

当以概率 q 重连网络中已有的 m_2 条边时,k_i 的变化率方程为

$$\left(\frac{\partial k_i}{\partial t}\right)_{\mathrm{II}} = -qm_2 \frac{1}{N(t)} + qm_2 \Pi(k_i). \tag{6}$$

(6)式等号右端第一项表示边的随机移除引起相应节点连通性的减少,第二项表示择优选择(4)式的节点连通性的增加. 边的重连不改变当前网络中边的数量.

当以概率 r 随机移除网络中已有的 m_3 条边时,k_i 的变化率方程为

$$\left(\frac{\partial k_i}{\partial t}\right)_{\mathrm{III}} = -rm_3 \frac{1}{N(t)} - rm_3 \frac{1}{N(t)}. \tag{7}$$

(7)式等号右端两项均代表边的随机移除引起相应节点连通性的减少.

当以概率 s 添加一个有 m_4 条边的新节点时,k_i 的变化率方程为

$$\left(\frac{\partial k_i}{\partial t}\right)_{\mathrm{IV}} = sm_4 \Pi(k_i). \tag{8}$$

(8)式等号右端表示对新边的自由端进行择优连接.

将(5)—(8)式相加,获得软件系统演化模型总的 k_i 的变化率方程

$$\frac{\partial k_i}{\partial t} = (pm_1 - qm_2 - 2rm_3)\frac{1}{N(t)} + (pm_1 + qm_2 + sm_4)\Pi(k_i). \tag{9}$$

在(9)式中,系统的当前节点总数 $N(t)$ 和总度数 $\sum_j k_j$ 都随着时间的变化而变化,有

$$N(t) = m_0 + st,$$

$$\sum_j k_j = p2m_1 t - r2m_3 t + s2m_4 t + 2e_0.$$

系统的演化往往使 t 远大于各初始参数,因此可以忽略起初始化作用的常数项 m_0 和 e_0.

定义模型的时间单元为加边、重连边、移除边和加带边点的过程之一. 时间单元 t_i 的概率密度为

$$p_i(t_i) = \frac{1}{m_0 + t}.$$

在时刻 t_i,添加一个有 m_4 条新边的节点 i,则该节

点的初始度为

$$k_i(t_i) = m_4,$$

那么(9)式关于 $k_i(t)$ 的解为

$$k_i(t) = \left(\frac{A}{B} + \alpha + m_4\right)\left(\frac{t}{t_i}\right)^B - \frac{A}{B} - \alpha, \quad (10)$$

式中，

$$A = \frac{pm_1 - qm_2 - 2rm_3}{s},$$

$$B = \frac{pm_1 + qm_2 + sm_4}{2m_1p - 2m_3r + 2m_4s + \alpha s}.$$

根据 Barabási-Albert 连续域理论，求得相应的度分布为

$$p(k) = \frac{t}{m_0 + t}\frac{1}{B}\left(\frac{A}{B} + \alpha + m_4\right)^{1/B} \times \left(\frac{A}{B} + \alpha + k\right)^{-\left(\frac{1}{B}+1\right)}. \quad (11)$$

当 $t \to \infty$ 时，

$$p(k) = \frac{1}{B}\left(\frac{A}{B} + \alpha + m_4\right)^{1/B}\left(\frac{A}{B} + \alpha + k\right)^{-\gamma}, \quad (12)$$

式中标度指数

$$\gamma = \frac{1}{B} + 1 > 1.$$

如果对参数 $p, q, r, s, \alpha, m_1, m_2, m_3$ 和 m_4 进行调整，总可以获得适当的标度值 γ.

为了说明该模型的有效性，采用目前通用的验证标度指数 γ 的方法，对软件系统 JDK1.4.0 进行标度指数分析. 根据初步统计，$m_1 = m_2 = m_3 = m_4 = 6$，相应的概率 $p = 0.13$，$q = 0.01$，$r = 0.09$，$s = 0.77$，同时令 $\alpha = 0$，那么系统 JDK1.4.0 的理论标度指数 $\gamma \approx 2.78$. 该值与实际值 2.49 有一个误差，该误差可以通过修正参数解决.

软件系统演化模型的数值仿真如图 4 所示. 在图 4(a)，(b)中，都有 $e_0 = 0$. 图 4 中的四方形、三角形和圆形分布图所对应仿真网络的参数都有 $N = 10000$，$m_0 = 10$，$m_1 = m_2 = m_3 = m_4 = 3$，$\alpha = 1$；而星形(对 Sun JDK1.4.0 的仿真结果)分布图所对应的仿真网络的参数 $N = 4000$，$m_0 = 20$，$m_1 = m_2 = m_3 = m_4 = 6$，$\alpha = 0$. 由图 4 可见，仿真结果与实证分析以及该模型的结论相一致，亦都呈现出幂律形式. 由于在该模型中引入了小概率的边的随机移除作用，导致了小部分节点的度小于 m_4，但大部分节点的度大于或等于 m_4. 图 4(a)，(b)中的星形图形是对 JDK1.4.0 系统进行仿真的结果. 对图 4(b)中星形图形进行直线拟合，获得仿真网络的标度指数为 2.8，接近理论标度指数 2.78. 由此可知，根据软件系统演化模型对实际的软件系统 JDK1.4.0 所进行的理论分析及仿真结果都表明了该模型的有效性.

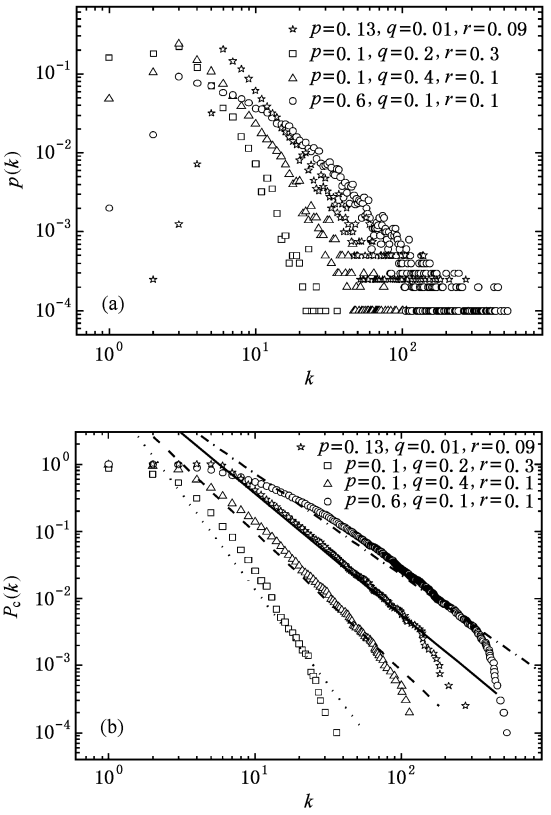

图 4　软件系统演化模型的仿真结果　(a)度分布的双对数坐标图，(b)累积度分布的双对数坐标图

6. 结　论

本文以 Java 软件系统为对象研究了大规模软件系统的复杂网络特性. 同时探讨了架构软件系统的分布式理念和代码重用技术在系统演化过程中的重要意义. 分析了 Sun 和 IBM 公司提供的大规模软件系统的复杂网络特性及其机制. 实证表明，大规模 Java 软件系统的度分布服从衰减幂律分布和无标度分布. 分布式理念和代码重用技术在软件系统的成长过程中发挥着重要作用，其中代码重用技术引起了‘富者愈富’现象的产生. 在软件系统网的演化过程中，还伴随着边的添加、移除和重连现象. 为了描述软件系统的演化过程，构建了一个软件系统演化模型. 该模型的度分布与实际相符，也是幂律分布. 根据该模型，可以对大规模 Java 软件系统的网络连接结构进行模拟、预测及评价. 如果一个实

际的大规模 Java 软件系统具有本文的复杂网络特性,它的整体软件架构就具有 Sun 和 IBM 风格. 具有衰减幂律度分布的系统,其度分布在线性坐标中呈现 L 形. 由此表明在 L 形分布图中应当存在贫富分界点. 如何对该点进行精确的数学求解与量化是需要进一步研究的问题.

[1] Erdós P, Rényi A 1960 *Publ. Math. Inst. Hung. Acad. Sci.* **5** 17

[2] Watts D J, Strogatz S H 1998 *Nature* **393** 440

[3] Barabási A L, Albert R 1999 *Science* **286** 509

[4] Albert R, Barabási A L 2002 *Rev. Mod. Phys.* **74** 47

[5] Albert R, Barabási A L 2000 *Phys. Rev. Lett.* **85** 5234

[6] Li Y, Shan X M, Ren Y 2004 *Acta Phys. Sin.* **53** 11 (in Chinese) [李 、山秀明、任 勇 2004 物理学报 **53** 11]

[7] He Y, Zhang P P, Xu T *et al* 2004 *Acta Phys. Sin.* **53** 1710 (in Chinese) [何 阅、张培培、许 田等 2004 物理学报 **53** 1710]

[8] Huang Z X, Wang X R, Zhu H 2004 *Chin. Phys.* **13** 273

[9] Myers C R 2003 *Phys. Rev.* E **68** 046116

[10] Newman M E J 2003 *SIAM Rev.* **45** 2

[11] Booch G, Rumbaugh J, Jacobson I 1998 *The Unified Modeling Language User Guide* (Boston: Addison Wesley) p96

The scale-free feature and evolving model of large-scale software systems*

Yan Dong Qi Guo-Ning

(*Institute of Contemporary Manufacturing Engineering, Zhejiang University, Hangzhou* 310027, *China*)

(Received 9 October 2005; revised manuscript received 10 April 2006)

Abstract

In software engineering, class diagrams are generally used to describe the relationship of classes. Software systems as networks are studied in this paper. By the demonstration and analysis of the large-scale software systems provided by Sun and IBM, it is found that the degree distribution of software systems written in Java is characterized by the scale-free distribution, and its scaling exponent γ is about 2.5. In the evolving process of software systems, in addition to addition of nodes, there are some other local events as follows: addition of edges, random removal of edges and rewiring edges. The evolving model of software systems is established consequently. As for the network generated by this model, its degree distribution follows the power-law distribution. The actual application and numerical simulations validate this model.

Keywords: software system, complex network, degree distribution, scale-free

PACC: 0250, 0565

* Project supported by the National Natural Science Foundation of China (Grant Nos. 60374057, 50575204).