一种可靠可伸缩组通信系统设计与实现*)

刘 畅 刘西洋 陈 平
(西安电子科技大学软件工程研究所 西安 710071)

摘 要 组通信系统是支持一致性和容错的分布式协同系统中非常重要的组成部分。为了满足大规模协同应用的需求,文中采用了基于流言的协议与确定性协议组合的方法设计并实现了一种可靠可伸缩组通信系统 SGCS。该系统主要包括可靠消息传输服务与组成员管理服务,其中基于流言的可靠多播协议和确定的消息恢复、流量控制、排序协议的组合,基于流言的失败检测协议与确定的视图一致化协议的组合以及乐观虚同步机制应用使系统具有良好的可伸缩性、可靠性和灵活性。

关键词 组通信,基于流言的协议,可伸缩性

The Design and Implementation of a Reliable and Scalable Group Communication System

LIU Chang LIU Xi-Yang CHEN Ping
(Software Engineering Institute, Xidian University, Xi'an710071)

Abstract Group communication systems are powerful building blocks for supporting consistency and fault-tolerance in distributed collaborative systems. In order to meet requirements of large-scale collaborative applications, a reliable and scalable group communication system SGCS is designed and implemented through the combination of gossip-based protocols and deterministic protocols. This system includes reliable message transport service and membership management service. The combination of gossip-based multicast protocol and the deterministic message recovery, flow control, ordering protocols, the combination of gossip-based failure detection protocol and deterministic view consistency protocol with the optimistic virtual synchrony mechanism make SGCS have the good scalability, reliability and flexibility.

Keywords Group communication, Gossip-based protocol, Scalability

1 引言

传统的通信模型和协议采用点到点的通信模式进行消息传递,此类协议适于在同一时刻仅涉及两个实体(客户/服务器)间通信的分布式应用。随着 Internet 与网络技术的发展,产生了计算机支持的协同工作这种新的分布式应用,此类系统中通信通常不仅仅涉及两个实体,而是多个地域分散的用户通过计算机网络以协作的方式来完成某项任务。为了保证系统多用户间的一致性和良好的容错性能,仅有点到点的通信模式是不够的,必须要有一到多与多到多的组通信模式来支持。典型的分布式协同应用包括军事指控系统、股票交易系统、航空交通管制系统、远程会议、教学、医疗系统以及网络在线游戏等。组通信服务为这些应用提供强有力的支持,已成为支持一致性和容错的协同系统中非常重要的组成部分。

组通信系统中的协议设计必须保证性能的可靠性,即在成员失效、网络暂时的丢包以及成员的连续加入/退出发生的情况下系统也能保证可靠地运行。此外,系统可伸缩性也很重要,协议算法应使系统施加在每个成员与整个网络上的负载随系统规模增加缓慢增长。

传统的组通信系统例如 ISIS, Horus, Esemble, Transis, Totem 等,普遍采用确定性的基于状态机的方法向用户提供容错与一致性保证,包括组成员关系的自动维护、通知用户成员关系的变化、可靠多播与各种次序排列性质。这些组通信系统有两个缺点:(1)在每个节点的开销随着系统的规模线性增加。(2)整个系统中的消息吞吐量会由于单个节点的扰动而大幅下降。因此虽然这些系统在小规模时均得到了较好的性能,但当规模扩大到一定程度时性能就出现了大幅的下降[1]。

近些年康奈尔大学的 Birman 等人提出一类基于随机流言的协议[4]。这类协议模仿流行病在密集人群中的传播,解决了大规模网络上的消息扩散问题,其典型代表就是基于闲聊的可靠多播协议 Bimodal[1]。这类协议即使在比较苛刻的环境中仍然可以保证较高的概率可靠性,有效克服了传统协议中制约可伸缩性的一些问题。但对于许多需要给用户提供确定可靠性保证的系统,仅能确保概率可靠性的基于随机流言的协议就不能满足应用需求了。

为了更好地满足大规模协同应用需求,文中采用了基于随机流言的协议与确定性协议组合的方法设计了一种可靠可伸缩组通信系统 SGCS(Scalable Group Communication System)。该系统主要包括可靠消息传输服务与组成员管理服务,其中底层采用的基于流言的协议为系统提供了概率可靠性和良好的伸缩性,而应用所要求的确定可靠性则由上层的一系列确定性的协议来保证。此外,多种消息提交次序的支持与乐观虚同步等机制的应用使该组通信系统具有良好灵活

*)基金项目:人民解放军总装备"十五"预研项目(编号 413150501)。刘 畅 硕士研究生,主要研究方向是分布式系统,面向对象技术。刘西洋 副教授,主要研究方向为设计模式、特定域体系结构、形式化验证。陈 平 教授,博士生导师,主要研究方向为面向对象技术、特定域体系结构等。

性。

2 SGCS的体系结构与环境假设

假设系统环境是异步的，即系统对进程间的消息传递延时没有限制，对进程的执行速度没有限制，且系统中没有一个同步时钟或全局时钟。假设通信链路是不可靠的，系统中可能发生多种错误：进程可能失效并可能恢复，网络可能分区成为几个部分并可能重新融合，消息有可能丢失。但错误不能改变消息的内容，即不允许发生拜占庭错误。

SGCS的体系结构如图1所示。SGCS建立在提供不可靠多播与单播服务的网络层之上给在其上的应用层提供可靠的组通信服务。其中，通信层是以UDP与IP多播为基础的。应用使用SCGS提供的接口来加入/退出多播组，发送/接收多播或单播消息并接收来自SCGS的成员关系变化的通知消息即新的视图。SGCS还提供了会话服务，允许用户通过自定义字符串名字来表示不同多播组。用户通过发送一个目标为某组名的消息来与此组中成员通信，组通信服务则负责消息传送给组中所有成员。

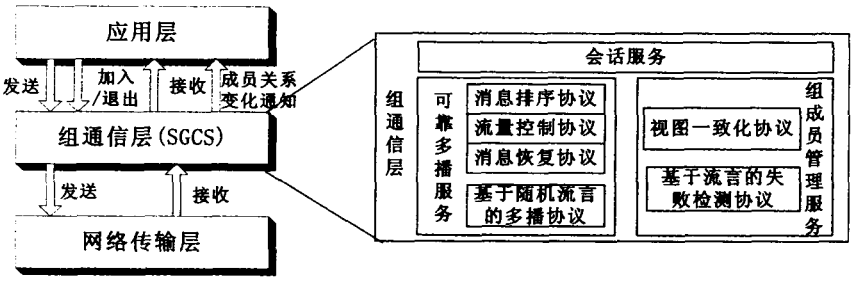

图1 SGCS体系结构图

SGCS主要由该系统主要由满足可靠消息传输服务与组成员管理服务组成。可靠消息传输服务的任务是确保在正常运行成员间发送的消息不会丢失，它由底层基于随机流言的多播协议与上层消息恢复协议、消息排序协议以及消息流量控制协议等确定性协议组合而成。组成员管理的任务是维护各组中当前正常运行的用户集合列表，在该集合发生变化时即时通知系统并向成员提供新的视图(组成员列表)，它由底层基于流言的成员失效检测协议和上层确定性的视图一致化协议组合而成。

3 SGCS的可靠消息传输服务

3.1 基于随机流言的多播协议

传统的可靠多播协议应用于大规模网络时可伸缩性就变差。其中某些协议看似具有一定的可伸缩性，但是一旦网络环境不再理想化之后，性能都急剧下降。这是因为在系统的运行过程中出现频率最高的错误为普通瞬时问题(Mundane Transient Problems)，包括网络或者处理器的调度延迟，丢包等等。在这种情况下，这些协议性能明显地下滑，而且这种小概率事件随着系统规模的扩大变得越来越容易发生，而且处理起来开销越来越大。随着网络规模的增大，普通瞬时问题制约着系统的性能，会出现吞吐量不稳定、微分裂、数据源"爆发"、消息请求和重发风暴等严重问题[1]。

为了解决上述制约系统可伸缩性的问题，借鉴Bimodal Multicast的设计思路，我们设计了基于流言的可靠多播协议PSRM[2]。该协议分为两个过程。第一个就是非可靠的数据分发过程，我们使用IP多播来实现。一旦消息到达接收者，该消息就会被放入消息缓冲区中。缓冲区中的消息都会以FIFO的次序提交到应用层，与此同时，每个一定时间就会运行垃圾回收将某些消息从缓冲区中去掉，避免缓冲区溢出。

第二个过程就是修复提交消息中的Gap(消息队列中不连续的部分)，具体操作如下，系统中的每个进程都要维护一个全局组成员信息的子集。为了有效控制这个子集的大小，我们在成员加入时采用可伸缩随机成员管理策略[3]使每个成员维护的子集大小控制在O(log$N$)，其中N为系统内总成员数。每隔一定时间，每个参与进程都要从自己的成员信息子集中选出某个进程，并且向它发送摘要报文(Digest)。该摘要报文表示出本地缓冲区中有哪些报文，没有哪些报文。一旦收到这种摘要报文，接收者便和自己本地的缓冲区中的相应报文相互比较。成员采用两种方式来恢复消息，一种是向发送者提出重传请求，请求对方发给它缓冲区中所没有的报文；另一种方式是主动发送给发送者所缺失的报文。

PSRM克服了制约可靠多播协议性能的问题，负载被分布在和系统规模成对数关系的成员上，使系统具有良好的可伸缩性。有关PSRM的详细情况请参考文[2,3]。

3.2 消息恢复协议

由于PSRM仅能确保消息传输的高概率可靠性，不能保证所有成员都能接收到消息，因此需要增加一个确定的消息恢复协议来保证消息传输的确定可靠性。因为PSRM已保证了很低的消息丢失率，所以我们使用基于NACK的消息恢复协议来实现消息的可靠传输，即成员只有在收到NACK时才进行消息重发。一条消息的传输首先通过基于随机流言的多播协议发送，每个成员给其发送的消息标记上连续增加的序号和成员标识一起作为消息的标识。为了能够对消息重发请求做出响应，成员将以前收到的多播消息当作备份保留在本地缓冲区内。这样一来，当某条消息的重发请求到达时，任何一个收到此条消息的成员均可做出响应，从缓冲区中读取相应消息并发给请求的成员。

由于缓冲区空间是有限的，成员不可能永远被保留消息。当所有成员收到某消息时，成员便可以安全地将这条处于稳定状态的消息从缓冲区删除。为了对此提供判断依据，成员在每次接收消息后要发送ACK，当收到所有成员对此消息的ACK时，一个成员就可以确定消息已达到稳定状态。为了减少消息数量、节省网络带宽，SGCS采用了"消息捎带"的方法，即成员不立即对消息进行响应，而是在准备发送的下一条多播消息中附加之前消息的ACK。消息附加的ACK信息直接重新构建了原本消息之间的因果序。因此每个成员可以通过分析本地收到消息形成的因果关系来得知消息的丢失。如果成员收到一个消息m，m含有m'的ACK，但成员还未收到

m'，出现了因果关系链的缺口，成员得知 m' 的丢失，就会发出重发请求。而 m 要一直等到 m' 恢复并被提交之后才能被提交。

通过这种 ACK 捎带方式 SGCS 的可靠消息传输服务不仅保证消息确定的可靠性而且在不增加额外消息的基础上实现对缓存空间的回收并形成了消息之间的因果关系。

3.3 流量控制协议

当通信负载很大的时候会导致网络与底层协议消息丢失率变高。而高丢失率又会使得消息恢复的代价巨大，这样会进一步导致雪崩效应。此外，丢失的消息不能得到及时的恢复会导致缓存的消息不能进行垃圾回收，越积越多，造成缓冲区的溢出。为了防止这些情况的发生，系统有必要控制网络中的消息流量。我们定义一个网络滑动窗口，窗口大小是还未被全部成员 ACK 的消息总数。每个成员仅根据本地信息就可计算这个窗口大小。与传统滑动窗口仅仅维护成员自身发出的消息不同，这里的窗口包括所有成员发出消息。系统可通过窗口的大小来确定一个合适的消息传输延时，范围从窗口很小时的最小延时到窗口达到最大上限时的无限延时，即停止发送应用消息，仅允许发送用于成员失效检测的“流言心跳”消息和其它控制消息。停止发送应用消息后，窗口不会卡住太长的时间，因为 SGCS 的组成员管理服务会及时地把失效的成员从新视图中剔除出去同时丢弃来自这些成员的不可提交的消息。这样将把阻塞中的窗口释放出来，成员就可以继续发送应用消息。这样，流量控制协议不仅控制了网络中消息的流量而且大体上确定了需要为消息重发保留缓冲区的最大值。

3.4 消息排序协议

消息排序协议为用户灵活地提供了多种消息提交次序支持，包括先进先出序(FIFO)、因果序(Causal)、全序(Total)与稳定(Safe)。

- “先进先出”序保证来自同一节点的消息在所有节点被提交的次序是根据它们被发送的次序而定，即保证消息按照序号由小到大的连续顺序提交。
- 因果序提交可以在有消息间 ACK 关系图支持的情况下直接完成。
- 全序保证在所有节点上消息均按照相同的顺序被提交。我们采用一个完全分布式的算法来实现，它通过本地的信息来建立全序并达成一致。基本思路是：(1)等待直到从每个成员至少收到一条消息后(2)按照成员 id 的升序提交那些在所有因果关系上没有未提交的前驱消息的消息集合，不能提交则等待重试。
- 稳定消息提交保证消息被所有节点收到后才被提交，即通过本地的信息得知每个成员均 ACK 了这条消息时进行提交。

4 SGCS 的组成员管理服务

4.1 基于流言的成员失效检测协议

正确的成员组成信息是组通信实现的基础，它保证系统在用户的动态加入、退出或发生故障时能够继续保持正常运行，避免服务器在判断消息稳定状态时因未考虑新加入成员的接收情况而过早地将消息从缓冲区清除，或因未及时排除失效用户而一直等待无法到来的消息 ACK 信息，使缓冲区空间不能被正常回收，影响系统向组成员提供可靠传输服务。因此，系统需要对成员进行失效检测，以便在变化发生时即时通知成员视图的改变，保证用户在正确的成员关系中进行通信。

为了解决传统的基于 all-to-all 心跳消息的成员失效检测协议开销大、可伸缩性差的问题。康奈尔大学的 Van Renesse 等人提出了基于流言的成员失效检测协议[5]，该协议在无需知道网络拓扑的情况下能高效地在网络中扩散失效检测信息。其基本算法思想是：每个成员维护一个它所知道的所有成员的列表。列表的每一项包括每个成员的 id 与“心跳”计数器。在每一个回合(round)即每隔 T_{gossip} 的时间，每个成员增加其自己对应的心跳计数器，并且从列表中随机选择一个成员将列表发给此成员。每当一个成员收到这样一个流言消息后，它将此消息的列表并入其自己本地的列表中，取每个成员“心跳”计数器的最大值。每个成员还为列表中每一个成员维护一个上次其对应“心跳”计数器增加的时间。如果在 $T_{suspect}$ 的时间过去后还未收到一个使能某成员对应“心跳”计数器增加的流言消息，这个成员将被怀疑为失效。

该流言协议存在几个缺陷。首先，随机“流言心跳”消息可能导致错误的检测。一个成员有可能因为长时间收不到含有另一成员“心跳”计数器更新信息的流言消息而错误地把此成员检测为失效。其次，网络的带宽有可能被浪费。在特定的 T_{gossip} 时间内，多个成员有可能向同一成员发送“流言心跳”消息而其它一些成员则收不到一条“流言心跳”消息。

为了解决这些问题，我们对基本流言协议进行了一些优化。首先，在每一个回合，每个成员发送流言的目的地将由下面的方程决定，其中 r 为当前的回合次数，N 为成员数：

目的地成员 ID=发送成员 ID+2^{r-1}，$1 \leqslant r < \log N$

这样一来，冗余的“流言心跳”消息就可以被完全消除，流言消息的发送和接收变得均匀。在特定的 T_{gossip} 时间内，每个成员将发送和接收各一条流言消息。而且这种流言消息的通信模式确保所有成员将在一个有限的 $\log N$ 回合内收到一个指定成员的“心跳”计数器更新值，即可以确定 $T_{suspect}$ 的最小值为 $(\log N) \cdot T_{gossip}$。实际上，由于各个成员的时钟不是同步的，虽然成员们以一个预先指定的次序来发送流言消息，但并不是所有节点都在指定给每个回合的时间内的同一时刻发送流言消息。一个成员有可能在发送它自己的流言消息之前先收到第二个成员的流言消息，并将此消息的列表并入其自己本地的列表中，这样第一个成员就将第二个成员流言信息在同一回合中传递给另一成员。因此，在异步环境中一个成员的流言信息很有可能在少于 $\log N$ 的回合内传递到所有成员，$(\log N) \cdot T_{gossip}$ 只是 $T_{suspect}$ 在最坏情况下的最小值。

其次，为了减少流言消息的发送间隔时间并进一步降低发送流言消息所占用的网络带宽，我们尽可能地把流言消息的信息附加在应用消息之上一起发送。这样每当附加有流言信息应用多播消息发送时，该成员一次便可将流言信息传给所有的成员。

4.2 视图一致化协议

当成员失效、成员的加入/退出或网络发生分区/重新融合等引起视图的变化时，SGCS 系统就要执行视图一致化协议来发起一个视图变化过程来在互连成员间达成统一的新当前视图 CV(CurrentView)。视图一致化协议将特殊的视图改变消息穿插在应用消息流中发送，这就提供了 CV 变化的信息，使每一个普通消息都在一个特定的视图中被发送和提交。视图一致化协议需要满足虚同步性质[6]，它要求系统确保经

(下转第 142 页)

tems. Artificial Intelligence, 1982, 19(1): 39～88
3 http://www.sics.se/isl/configuration/configurators.html
4 Mittal S, Frayman F. Towards a generic model of configuration tasks. In: Sridharan N S, ed Proc. of the 10th intl. joint conf. on artificial intelligence, San Mateo Morgan Kaufmann publishers, 1989. 1395～1401
5 Sabin D, Weigel R. Product configuration frameworks-A survey. IEEE Intelligent Systems, 1998, 14(3): 42～49
6 Friedrich G, Stumptner M. Consistency-based configuration. www.ifi.uni.klu.ac.at/IWAS/staff/Gerhard.Friedrich
7 Stumptner M. An overview of knowledge-based configuration. AI Communications, 1997, 10(2): 1～16
8 Kokeny T. a new arc-consistency algorithm for csps with hierarchical domains. In: workshop Notes of the ECAI'94
9 Mittal S, Falkenhainer B. dynamic constraint satisfaction problems. In: proc. of the 8th national conf. on artificial intelligence, AAAI Press, 1990. 25～32
10 Freuder E. Constraint solving techniques. Constraint Programming, 1992, 131: 51～74
11 Sabin D, Freuder E. Configuration as composite constraint satisfaction. In: Faltings B, freuder E, eds. Configuration-Papers from the 1996 Fall Symposium, AAAI Press, 1996. 28～36
12 Heinrich M, Jüngst E. A resource-based paradigm for the conf. of technical systems form modular components. In: Proc. of the 7th IEEE Conf. on AI Applications(CAIA), 1991. 257～264
13 Stumptner M, Haselböck, Friedrich G. COCOS-a tool for constraint-based, dynamic configuration. In: Proc. of 10 th IEEE Conf. on AI Applications(CAIA), San Antonio, 1994. 373～380
14 McGuinness D L, Wright R. An industrial-strength description logic-based configuratior platform. IEEE Intelligent Systems & Their Applicatins, 1998, 13(4): 69～77
15 Wache H, Kamp G. Using description logic for configuration problems. http://www.informatik.uni-bremen.de/～wache/publications.html
16 McGuinness D. Configuration. In: Baader F, Mcguinness B, narli D, eds. The Description Logic Handbook: Theory Implementation, and Application, Cambridge university Press, 2002. 397～414
17 Junker U, Mailharro D. The logic of ILOG(J) Configurator: combining constraint programming with a description logic, IJCAI on configuration, 2003
18 Soininen T, Niemela I, et al. Represnting configuration knowledge with weight constraint rules.
19 Geneste L, Ruet M. Fuzzy case based configuration. In: Stumptner M., ed. Papers from the workshop at ECAI 2000 on Configuration, Berlin, Germany, 2000. 71～76
20 李占山，王涛，孙吉贵，张朝辉．产品配置器的工作机理研究．计算机应用研究，2004，21(10)
21 李占山，寇飞红，孙吉贵，王崧，任键．一种产品配置知识的图形表示与配置求解
22 Stumptner M, ed. Papers from the workshop at ECAI 2000 on Configuration, Berlin, Germany, 2000
23 Aldanodo M, ed. Papers from the workshop at ECAI 2002 on Configuration, Lyon, France, 2002

(上接第 48 页)

历了两个连续视图中的成员能够提交相同的介于这两个视图之间的消息集，使成员状态在视图发生变化时得到同步。成员的失效和脱离由基于流言的成员失效检测协议来完成。新成员的加入与网络融合以一个统一的对称方式处理，均可看成原本未连接的部分融合成为一个更大部分。这种处理方式也解决了系统启动时的问题：每个成员启动时其视图只含有其本身，所有启动的成员相互融合成为一个集合视图。视图的变化通过一系列的多播通信操作完成。

视图一致化协议包括 3 个阶段：

(1)变化发起：成员通过发送视图变化消息来发起一个视图变化过程。

(2)达成一致：所有的成员动态地对下一个要进行的视图变化达成一致。而且，所有的成员还必须对在下一个视图变化前需要提交的消息集合达成一致。这个消息集合包括在视图变化消息之前或同时发送的消息。这样做是为了满足前面提到的虚同步的性质。

(3)新视图提交：当在上一阶段达成一致的集合中的所有消息被提交之后，所有成员通过提交达成一致的视图变化消息来安装新的视图。

为了防止协议无法达成一致而无限制地运行下去，协议的“变化发起”阶段和“达成一致”阶段均设置制定时器来控制时间，它们开始执行后会在一个有限时间内终止。为了满足虚同步的性质，“新视图提交”阶段有可能包括上一视图的消息提交过程。这种情况发生在一个待提交的消息等待一个失效的成员 q 来发送 ACK 确认收到这条消息。当新视图提交把失效的成员从新视图中剔除出去后这个消息便可以顺利地被提交。

大多数的传统组通信系统为了满足虚同步的性质在从得知视图开始发生变化直到新视图被提交的时间段内不允许成员发送应用消息，这样会浪费掉宝贵的运算时间和网络资源。为了满足虚同步的性质并且无需在视图变化期间阻止成员发送应用消息，我们采用了乐观虚同步机制[7]。在乐观虚同步中，每个新视图安装之前都会有一个乐观视图，乐观视图是对下一可能视图的估计。当成员接收到乐观视图便进入乐观模式。在这种模式下，成员接收上一视图的消息，也可以乐观地发送消息，以便在下一个视图中传递。在乐观模式发送的消息叫乐观消息。当成员安装新视图后回复到正常模式并检查是否可以在新视图里提交乐观消息。一条乐观消息发送时所在的乐观视图如果是新视图的子集，这条乐观消息便可以提交；否则将被回滚(rollback)。

结束语 SGCS 组通信系统采用了基于流言的协议与确定性协议组合的设计方法。可靠消息传输服务中的基于随机流言的多播协议与组成员管理服务中的基于流言的成员失效检测协议使系统负加在每个在成员上与网络上的负载随系统规模增长以对数的关系(logN)缓慢增长，及具有良好的可伸缩性，而消息恢复协议、流量控制协议与视图一致化协议保证了系统确定的可靠性。另外，多种消息提交次序的支持与乐观虚同步机制的应用时系统具有了良好的灵活性。综上所述，SGCS 系统为大规模协同应用提供了高性能的可靠可伸缩群组通信服务。

参 考 文 献

1 Birman K P, Hayden M, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, Yaron Minsky. Bimodal Multicast. ACM Transactions on Computer Systems, 1999, 17(2): 41～88
2 范晓鹏. 基于随机闲聊的可靠多播的可靠性研究:[硕士论文]. 西安电子科技大学，2004. 1
3 王伟波. 可伸缩随机成员管理策略的研究:[硕士论文]. 西安电子科技大学，2004. 1
4 Birman K P, Gupta I. Building Scalable Solutions to Distributed Computing Problems using Probabilistic Components: [Cornell Technical Report]. 2004
5 Renesse V, Minsky RR, Hayden M. A Gossip-Style Failure Detection Service. In: Proc. of the IFIP Intl. Conf. on Distributed Systems Platforms and Open Distributed Processing Middleware ' 1998
6 Birman K, Joseph T. Exploiting Virtual Synchrony in Distributed Systems. In: 11th ACM Symposium on Operating Systems Principles, 1987
7 Sussman J, Keidar I, Marzullo K. Optimistic virtual synchrony. In: 19th IEEE Intl. Symposium on Reliable Distributed Systems (SRDS), 2000, 10: 42～51