

doi: 10.7690/bgzdh.2015.09.011

# 一种可伸缩的指挥显示数据处理模式

段慧芬, 张德华, 王 华, 刘焕敏, 赵建平

(中国卫星海上测控部试验技术部, 江苏 江阴 214431)

**摘要:** 为满足航天测控试验任务指挥信息的多样性和指挥控制需求的多变性需求, 提出一种可伸缩的指挥显示数据处理模式。通过分析影响处理的伸缩性要素, 将指挥显示信息处理的可伸缩性分解为处理能力的水平伸缩和垂直伸缩 2 个维度。构建数据模型、数据处理模型和数据处理过程树模型, 并选用 XML 文件对处理过程树模型进行记录, 设计实现了定制态、就绪态和运行态的数据处理生态模式。分析结果表明: 该方法使应用软件具备了数据处理可伸缩能力, 可以满足测控信息变化带来的可伸缩性需求。

**关键词:** 可伸缩; 数据处理; 指挥显示**中图分类号:** TP391 **文献标志码:** A

## Scalable Data Processing Model of Command and Monitoring System

Duan Huifen, Zhang Dehua, Wang Hua, Liu Huanmin, Zhao Jianping

*(Experiment Technology Department, Satellite Marine Tracking & Control Department of China, Jiangyin 214431, China)*

**Abstract:** In order to meet the demand of aerospace experiment tracking task command information diversity and command and control variability, a scalable command display data processing mode is proposed. By analysing the scalable factor which effect the processing results, the scalability of command and monitoring information processing is decomposed into two dimensions of horizontal expansion and vertical expansion of processing capacity. Through constructing data model, data processing model and data processing tree model, selecting XML file to record the process tree model, designed and realized the ecological model of data processing which with custom state, ready state and operation state. The analysis results show that the method make the application software have the data processing scalability, can meet the scalability demand of the change of measurement and control information.

**Keywords:** scalable; data processing; command and monitoring

### 0 引言

在试验任务中, 各种测量、控制设备采集的试验信息实时汇聚于指挥显示系统, 指挥显示系统要及时准确地将这些信息进行处理, 并将处理结果实时提供给指挥人员、总体技术人员, 作为验证飞行器各项试验指标是否达到设计要求、试验安全控制和试验指挥的决策依据。为了满足指挥信息的多样性和指挥控制需求的多变性, 需要设计一种可伸缩的实时数据处理架构。

可伸缩性 (scalability) 反映了系统可随系统需求和资源变化, 持续满足性能需求的能力<sup>[1-3]</sup>。可伸缩性分为硬件和软件 2 个方面: 硬件方面的可伸缩性指的是通过改变硬件资源来满足变化的工作负载, 比如改变处理器的数目、内存和硬盘的容量; 软件方面的可伸缩性是通过改变调度方式和并行化程度, 来满足变化的工作负载<sup>[4-5]</sup>。

对应用系统而言, 当系统需要承受更大的用户负载和改变用户行为时, 不用修改它现有的体系结

构和设计, 或至少保证这些修改带来尽可能小的影响, 否则会产生瓶颈, 降低软件的性能<sup>[6-7]</sup>。所以, 必须在软件设计过程中充分考虑可伸缩能力。

### 1 指挥显示数据处理可伸缩性分析

#### 1.1 数据处理过程

在航天测控任务中, 由部署于天基、海基、陆基测站的测控设备对目标跟踪测量, 采集各类测控信息, 通过测控网将信息汇聚于任务指挥中心, 经指挥中心的的处理服务器群处理后, 提供给指挥人员进行目标健康状况、目标运行轨迹、目标姿态变化、目标载荷系统工作情况的监视控制。测控系统的组成如图 1。测控系统采集的实时测控信息, 是以字节流的形式传输的。数据处理要解决的问题包括多目标、多信息源和多信息类型的信息汇聚, 对收到信息进行有效性检查, 对有效的信息按分类规则进行处理, 最后, 将处理结果分发到监视显示、辅助决策、飞行仿真等指挥应用系统。在处理中首

收稿日期: 2015-04-29; 修回日期: 2015-05-27

作者简介: 段慧芬(1963—), 女, 吉林人, 硕士, 高级工程师, 从事航天测控、计算机应用研究。

先要对每类数据的格式进行识别，对要处理的参数原码进行寻址定位和提取，将提取的原码依据业务处理要求处理转化为有效物理量或状态参数。

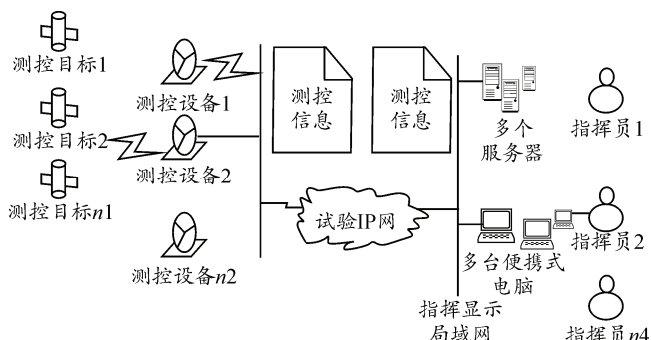


图 1 航天测控系统组成

### 1.2 影响处理的伸缩性要素

从图 1 中可以看出：对指挥显示数据处理可伸缩性产生影响的要素主要是测控目标的数量和型号变化、参试测控设备的数量和型号变化、测控信息种类和内容的变化、还有指挥员对数据最终处理结果要求的变化。这些变化最终反应在测控信息量、信息种类、信息内容和处理结果的扩展。把这些对

数据处理伸缩性能力的需求归纳为 2 个方面：一是处理能力的水平伸缩性，包括适应测控信息量、测控信息种类和测控信息内容变化的伸展处理方法；二是处理能力的垂直伸缩性，包括适应处理结果变化的伸缩性处理方法，例如同一个参数的多态处理、参数间的相关依赖处理等。

## 2 可伸缩数据处理方法

### 2.1 数据描述模型定义

要解决数据处理的可伸缩性，就不能将处理过程固化在程序中，需要对处理的数据对象进行形式化描述。为此，先定义基本的数据模型 (byte object model, BOM) 库，内容包括字段数据模型、位数据模型、时间数据模型、组合数据模型，数据模型库中包含的具体模型元素见表 1~表 4。这种形式化描述可以解决处理能力的水平伸缩，对于每一新型号任务的变化、测控设备的变化、测控信息的变化，都可以通过数据模型库中的基本元素进行重新定义，即使出现数据模型库中不能描述的数据，也可以通过增加数据模型元素的方式进行表示，而不会影响到处理程序的体系结构。

表 1 字段数据模型

字段方法	描述内容
SByteBOM	描述编码类型为有符号 8 位整数的字段
ByteBOM	描述编码类型为无符号 8 位整数的字段
Int16BOM	描述编码类型为有符号 16 位整数的字段
UInt16BOM	描述编码类型为无符号 16 位整数的字段
UInt24BOM	描述编码类型为无符号 24 位整数的字段
Int32BOM	描述编码类型为有符号 32 位整数的字段
UInt32BOM	描述编码类型为无符号 32 位整数的字段
Int64BOM	描述编码类型为有符号 64 位整数的字段
UInt64BOM	描述编码类型为无符号 64 位整数的字段
HexBOM	描述一段 16 进制源码数据字段，支持常量约束
PadBOM	描述一段不需要处理的字节流，不对任何结果

表 2 位数据模型

字段方法	描述内容
BitBOM	描述由若干个 Bit 组成的数据，只能作为 BitBoxBOM 的子 BOM。
BitIntegerBOM	把 Bit 数据解析为十进制整数 (左边是高位)。
BitStateBOM	把 Bit 数据解析为状态文本，支持定义依赖关系，即解析不止取决于状态 BOM 自身，还要看其它的状态 BOM 的状态值。
BitPadBOM	描述一段不需要处理的 Bit 流，不对任何结果，类似 PadBOM。
BitStringBOM	把 Bit 数据解析为以“0”和“1”组成的字符串。

表 3 时间数据模型

字段方法	描述内容
DatetimeBOM	描述封装了时间数据字段：2 字节积日 + 4 字节积秒值 (0.1 ms 精度)，解析结果是时刻 (DateTime 类型)，日期是积日日期。
TimeBOM	描述了时间数据字段 (4 字节积秒值)，解析结果是时刻 (DateTime 类型)，其中的日期是 DP 的虚拟时统日期。
JYTimeBOM	从 TimeBOM 继承，描述封装了箭遥挑点时间字段，不带复用区别码，解析结果是箭遥时间指令对应的相对 T0 飞行时 Ti。
JYTimeMuxBOM	从 JYTimeBOM 继承而来，描述封装了箭遥挑点时间字段，前面带有 5 个 Bit 的复用区别码。

表 4 组合数据模型

字段方法	描述内容
RepeatBOM	属于组合 BOM，它的子 BOM 全部对应(处理)同一段字节序列。支持对同一字节序列同时应用多种不同的处理方式，得到不同的结果。
BitBoxBOM	一种组合容器 BOM，它的子 BOM 只能是 BitBOM，它对其下的 BitBOM 进行(按字节对齐)整体处理。容器中的子 BitBOM 是严格按顺序组织的，其顺序就是按照从左到右(从高 Bit 为到低 Bit 位)顺序。
SequentialBOM	组合 BOM，它的子 BOM 按照固定的顺序构成，顺序运行。该顺序是 desintime 决定的。
CaseBOM	CaseBOM 是一个特殊的顺序 BOM，定义了分支条件值，它只能作为 SwitchBOM 的子 BOM。
SwitchBOM	SwitchBOM 属于复用组合 BOM，它的子 BOM 只能是 CaseBOM，将分支判断条件与 CaseBOM 的分支条件值进行比对，选择其中一个 CaseBOM 执行。
VirtualBOM	虚拟 BOM 在物理上并不对应实实在在的 Data 字节序列，它的字节序列是从字节流中挑选出来，再顺序拼装好的是虚拟的字节流。通过适当的设置虚拟 BOM 的组成字节块，可以实现强大的任意拼装虚拟字节序列的功能。

2.2 数据处理方法模型定义

依据数据模型的描述，数据处理系统可以将数据流转化为有效的数据源码字段。这些有效的数据源码字段还需要进一步处理为有效的物理量，才能提供给指挥显示系统使用。处理为物理量的主要过程包括量纲转换、公式计算等内容。不同的任务、不同的监视显示要求，需要处理的最终结果也不同。

通过对数据处理方法的形式化描述，生成数据处理过程序列，动态调度数据处理方法，实现数据处理的垂直可伸缩。作为数据处理形式化描述的前提，用数据处理方法库先定义出数据处理所需要的基本处理方法。经过对各类测控数据处理要求的分析，笔者归纳出如表 5 所示的数据处理方法库内容<sup>[8]</sup>。

表 5 数据处理方法模型

处理方法	描述内容
SectionMethod	字段 BOM 的处理方法，对字段的编码值进行解析，得到数据值。
PacketMethod	PDXP 包处理方法，待字段解析完成后对整个包的解析结果进行“二次处理”，如发送回复消息、新增处理结果等。
SingleMethod	单精度浮点数方法，将字段编码值处理成 Single 类型。
DoubleMethod	双精度浮点数方法，将字段编码值处理成 Double 类型。
IntegerMethod	整型方法，将字段编码值处理成 Integer 类型。
DateMethod	积日值处理方法，将字段编码值处理成 DateTime 类型(不含 Time 数据)。
TiMethod	相对 T0 飞行时方法，将字段编码值转换成相对 T0 飞行时，单位：s，类型 float.精度为 ms，即 3 位浮点小数。
NewResultMethod	新建结果包处理方法，添加一个新的结果(由 BID 标示的新的一些字段序列，如箭遥事件)。
ActionMethod	包动作方法，执行某种操作，动作的方法，不会改变现有的字段值。
AddSectionMethod	添加字段方法，在现有的字段解析基础上，增加若干字段值。
ChangeSectionMethod	修改字段方法，在现有的字段解析基础上，修改若干字段值。
JYTimeMethod	箭遥时间处理方法，对解算出来的箭遥时间参数，进行再处理，得到箭遥时间参数累积帧。
SZRRMethod	瞬时站址测距方法，根据统一系统瞬时站址中的测距数据，结合测角数据(测速数据推算目标经纬度速高和圈号(DX-2))。
AccumulateMethod	累积包处理方法，对当前解算出来的数据结果进行持续缓存。如果当前包中没有解算出某结果，但之前解算出来过，那就用原来的结果作为最终的解算结果，最终的结果帧中的数据结果可能是来自多个次处理解算结果。
ReplyMethod	回答方法，根据当前数据的比对结果，设置回答状态，向当前包的发送方反馈回答。
SZAEMethod	瞬时站址测角方法，把统一系统瞬时站址中的测角数据添加到瞬时站址同步器中。
Check3P2Method	3 判 2 数据比对方法，对重复发送的关键数据进行 3 判 2 比对，即连续 2 个 PDXP 数据包一致，则认为比对一致，比对成功后，忽略后续重复数据。
GDMMethod	轨道处理方法，专门处理轨道参数，触发外推。
FCMethod	分离点参数处理方法，提取分离点时刻，添加到全局环境中，以触发计算运行时。
T0Method	T0 处理方法，提取并设置有效的 T0 值。
SZRRMethod	瞬时站址测速方法，把统一系统瞬时站址中的测速数据添加到瞬时站址同步器中。
ForwardMethod	转发方法，把当前数据的处理结果，转发给第三方(PDXP 协议)。
RawCodeMethod	原码处理方法，把整个数据原码提取出来，添加在结果中。
ArithmeticMethod	算术运算方法，实现以算术脚本的形式对解析出来的若干字段进行复杂算法计算，得到新的字段结果。
SZJWSGMethod	瞬时站址经纬速高方法，处理统一系统瞬时站址类数据，从 RR,R,A,E,L,B,K,Vp,计算出 V,H,L,B,Q。
DDJWSGMethod	弹道经纬速高处理方法，处理弹道类数据，从 X,Y,Z,Vx,Vy,Vz,计算出 V,H,L,B,Q。

3 数据处理过程

3.1 数据处理过程模型

为了满足数据处理的水平伸缩性和垂直伸缩性，在数据模型库和处理方法模型库的基础上，选取树形结构作为数据处理过程模型，通过增加处理过程树和树的分枝实现水平伸缩，通过增加树的结点或树的分枝深度，实现垂直伸缩。除了可以满足伸缩性的需求外，还有结构清晰、易于描述、方便对处理过程进行序列化处理等优点。

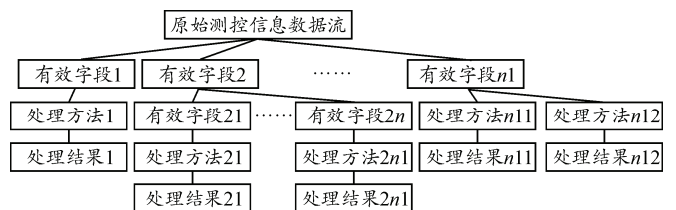


图 2 数据处理过程树模型

图 2 给出了数据处理基本过程树模型，有 3 种分枝情况，第 1 个分枝代表是最简单的分枝处理过程，第 2 个分枝代表每个节点还可以有若干子处理分枝，第 n1 个分枝代表一个有效字段可以通过不同

的处理方法产生不同的处理结果。模型中的每个节点都可以用数据描述模型、数据处理模型库中的元素来表示。

构建完处理过程树，按照树的索引算法对处理过程进行序列化，序列化的原则选择从左到右深度优先的索引原则，就完成了某类测控信息的处理过程的动态构建。

### 3.2 处理过程树之间的关系

为了确保处理能力的可伸缩性，在调度处理树的运行中，在每个处理过程树之间设计了3种关系：

1) 独立关系，只要有符合要求的原始数据输入，就可以启动处理过程树，树中的处理过程遍历结束后，就完成了该处理；2) 顺序关系，处理过程树需要等待其他某一个树的处理结果产生后，才能开始启动处理；3) 同步关系，处理过程树需要等待其他多个处理树的处理结果都产生后，才能开始启动处理。通过这3种关系，建立起处理树之间的调度运行关系，便于操作系统进行资源均衡分配。

### 3.3 处理过程的描述

可扩展标记语言XML，具有跨平台通用性好，规则严谨，使用简单，程序易读等优点<sup>[9]</sup>，选用XML文件，记录保存处理过程树模型结构及其之间的关系。下面是定制生成的雷达测量信息处理的资源文件片段，记录了时间字段处理方法和对瞬时站址处理结果进行再次处理，计算出目标的速度、高度、经度、纬度的描述：

```
<?xml version="1.0" encoding="utf-8"?>
<DataSchema>
  <MID>不限</MID>
  <SID>不限</SID>
  <BIDI>70</BIDI>
  <Version>1.0.2014.0127</Version>
  <BOM Type="SequentialBOM">
    <Name>瞬时站址脉冲雷达数据</Name>
    <Caption></Caption>
    <Comment></Comment>
    <ForeignKeys />
    <Endian>LittleEndian</Endian>
    <Priority>Auto</Priority>
    <Address>0 : 37</Address>
    <BOM Type="BitRawBOM">
      <Name>ZT</Name>
      <Caption>设备状态</Caption>
      .....
    </BOM>
  <BOM Type="UInt32BOM">
```

```
    <Name>T</Name>
    <Caption>时标</Caption>
    <Comment></Comment>
    <ForeignKeys />
    <Endian>LittleEndian</Endian>
    <Priority>Auto</Priority>
    <Address></Address>
    <PrimaryKey>False</PrimaryKey>
    <InvalidCode />
    <TTL>1</TTL>
    <Methods>
      <Method Type="DateTimeMethod">
        <Name>M</Name>
        <Return>True</Return>
      </Method>
    </Methods>
  </BOM>
  .....
</BOM>
<Methods>
  <Method MethodType="SZMethod">
    <MVSection>MV</MVSection>
    <MHSection>MH</MHSection>
    <MLSection>ML</MLSection>
    <MBSection>MB</MBSection>
    <MQSection>MQ</MQSection>
    <RRSection>RR.M</RRSection>
    <RSection>R.M</RSection>
    <ASection>A.M</ASection>
    <ESection>E.M</ESection>
    <LSection>L.M</LSection>
    <BSection>B.M</BSection>
    <KSection>K.M</KSection>
    <VpSection>Vp.M</VpSection>
  </Method>
</Methods>
</DataSchema>
```

## 4 软件实现

### 4.1 软件运行模式

为了满足指挥显示业务可伸缩的处理需求，将数据处理构建一个具有定制态、就绪态、运行态的生态系统，见图3。

定制态完成数据处理方法的定义，就绪态加载数据处理方法，运行态依据解析的处理方法进行实时处理，并生成处理结果。系统具备运行态既可以返回到就绪态重新加载数据处理方法，又可以直接

返回到准备态重新定义数据处理方法。通过处理方法的定义实现处理能力的可伸缩。

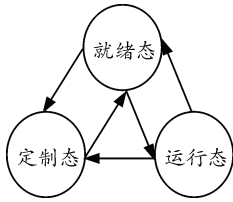


图 3 数据处理生态系统

### 4.2 软件流程

在 Windows 环境下，采用微软提供的 Visual 2010 开发平台，用 C#语言开发出了一个试验任务指挥显示实时数据处理系统。系统以组件库的形式分别建立了对应数据描述模型的数据模型组件库，对应数据处理方法模型的处理方法组件库，通过数据模型组件和方法模型组件配置，生成了每类测控信息的处理过程定义，用数据资源文件动态记录保存了每类数据的处理过程和处理过程的相互关系。图 4 给出了软件主要流程及状态转换关系。软件在定制态的用户界面如图 5 所示，软件在运行态的监视界面如图 6。

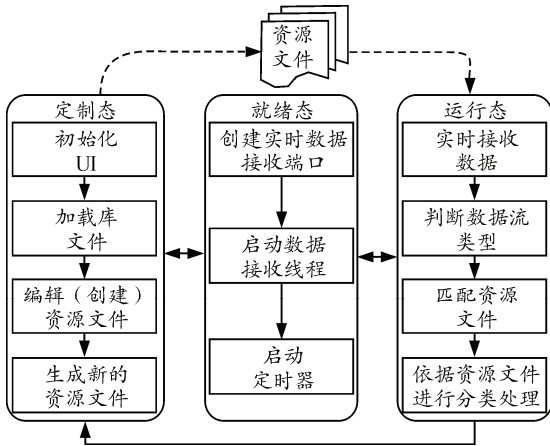


图 4 数据处理主要流程



图 5 定制态 UI



图 6 运行态监视 UI

### 5 结论

笔者以航天试验任务指挥显示数据处理为应用背景，根据测控信息的特点，构建了以信息描述为基础的字段数据模型、位数据模型、时间数据模型、组合数据模型，以信息流处理结果为基础的处理方法模型，在此基础上构建了以处理过程为基础的处理过程树模型，在架构上设计了定制态、就绪态和运行态 3 种生态模式，定制态主要完成处理过程树的生成，就绪态主要完成数据通信接口、处理线程的创建，运行态主要依据处理过程树的描述完成实时数据接收和处理。通过以上设计，使应用软件具备了数据处理可伸缩能力，可以满足测控信息量、测控信息种类、测控信息内容和处理结果变化带来的可伸缩性需求。开发的应用系统，经过多次试验任务的检验，有效地提高了指挥显示数据处理的适应能力。

### 参考文献:

- [1] 章文嵩. 可伸缩网络服务的研究与实现[D]. 长沙: 国防科学技术大学, 2000: 10.
- [2] 陈斌, 白晓颖, 马博, 等. 分布式系统可伸缩性研究综述[J]. 计算机科学, 2011, 38(8): 17-24.
- [3] 叶伟, 赵进, 叶军. 互联网时代的软件革命: SaaS 架构设计[M]. 北京: 电子工业出版社, 2009: 132-133.
- [4] 李万红. 构建高度可伸缩的 .NET 数据库应用程序[M]. 北京: 清华大学出版社, 2003: 7-9.
- [5] 马俊. 基于 .NET 平台的可伸缩性研究与实现[D]. 武汉: 武汉理工大学, 2006: 11.
- [6] 周奇. 可伸缩性软件构架的研究与其在 .NET 中的实现[D]. 重庆: 重庆大学, 2004: 11.
- [7] DamonAllison, 等. vb.net 可伸缩性技术手册[M]. 北京: 清华大学出版社, 2003: 272-300.
- [8] 耿金良, 王劲, 孙千里, 等. 自动化设备的数据采集与计算机处理技术[J]. 机电工程, 2014, 31(5): 616-619.
- [9] 李晓光, 宋宝燕. XML 非完全结构查询技术[M]. 北京: 北京师范大学出版社, 2011: 6.