

EAST 即时通信系统^①

王 兰^{1,2}, 肖炳甲^{1,2}, 袁旗平¹

¹(中国科学院 合肥物质科学研究院, 合肥 230031)

²(中国科学技术大学, 合肥 230026)

通讯作者: 袁旗平, E-mail: qpyuan@ipp.ac.cn

摘 要: EAST 放电实验会产生大量的实验数据, 为了满足实验人员高效地获取实验数据、了解实验进展的需求, 提出了基于 Openfire+Spark 框架的 EAST 即时通信系统解决方案. 该系统具有状态呈现、即时沟通、共享文件、订阅实验数据等功能, 其中, 重点研究了实验数据的订阅与推送. 订阅/推送功能主要是由订阅网站系统和后台推送系统共同实现, 实验人员通过订阅系统订阅感兴趣的实验数据, 后台推送系统在监测到有新数据产生时, 其读线程即从 MDSPlus 数据库中读取实验数据并交由数据处理线程进行绘图处理并打包成带有订阅用户 JID 的 Message 格式消息, 最后由发送线程将其转发到服务器. 该系统解决了实验人员不能即时地获取实验状态和实验结果的问题, 为 EAST 实验提供了一个良好的信息交流平台.

关键词: EAST 即时通信; 订阅/推送; 数据处理; Openfire

引用格式: 王兰, 肖炳甲, 袁旗平. EAST 即时通信系统. 计算机系统应用, 2018, 27(8): 108-113. <http://www.c-s-a.org.cn/1003-3254/6482.html>

EAST Instant Messaging System

WANG Lan^{1,2}, XIAO Bing-Jia^{1,2}, YUAN Qi-Ping¹

¹(Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China)

²(University of Science and Technology of China, Hefei 230026, China)

Abstract: As EAST discharge experiments will produce a large amount of experimental data, it becomes necessary to organize these experimental data to make users can efficiently obtain them and understand the progress of experiments. In this study, we propose the EAST instant messaging system based on the Openfire and Spark framework, which can communicate instantly, share files, and subscribe the experimental results. But, we focus on the subscription and push function. The Subscription/Push includes two parts: front-end subscription website and the back-end push system. Researchers can subscribe their experimental data in website. The read thread of the back-end push system will read data from MDSPlus once new experimental data generates, then the data processing thread will handle them and package into Messages with the users' JID, finally the send thread will send Messages to the server. The system is designed to solve the problem that researchers cannot obtain the progress and the results of experiments in time and it provides a good communication platform for EAST experiment.

Key words: EAST instant messaging; subscription/push; data processing; Openfire

EAST (Experimental Advanced Superconductive Tokomak) 是中科院等离子体物理研究所设计的一个

环形长和极向场全超导的托卡马克装置^[1], 也是世界上第一个全超导核聚变实验装置, 该装置于 2006 年 3 月

① 基金项目: 国家磁约束核聚变能发展研究专项 (2014GB103000)

Foundation item: National Special Project of Magnetic Confinement Nuclear Converging Energy Research (2014GB103000)

收稿时间: 2017-12-12; 修改时间: 2018-01-04; 采用时间: 2018-01-16; csa 在线出版时间: 2018-07-28

成功地进行了首次通电实验. 在 EAST 实验中, 会产生大量的诊断数据和数值计算数据^[2], 不同的科研人员对数据的需求不尽相同. 因此, 基于该实验平台产生的实验数据, 我们设计并实现了 EAST 即时通信系统, 本文着重介绍该系统中订阅/推送^[3]功能的设计与实现.

1 研究背景

随着互联网的快速发展, 即时通信早已成为继 Web 和 Email 之后最主要的基础应用之一. 近年来, 企业级即时通信具有很强的竞争力, 很多企业和机构都选择搭建内部即时通信系统来办公, 以此提高安全性和办公效率. 搭建企业内部的即时通信系统目前有三种较流行的方式, 一是通过软件服务商提供的即时通信云服务, 比如融云、环信、网易云信等, 这种方式的优点是技术成熟、部署快速、节省时间和人力成本, 缺点是费用随着功能的增加而增加, 无法实现更深层次的功能定制, 数据和服务存在一定的安全隐患; 二是选择专注于团队办公、提高办公效率的成熟软件, 比如腾讯的 TIM, 阿里的钉钉, 这种方式的优点是产品成熟且免费, 功能十分齐全, 但是对于研究机构来讲, 很多功能是冗余的, 也无法对系统拥有完全的控制权; 三是通过开源软件来搭建内部即时通信系统, 这种方式的优点是开源免费、可扩展性强, 可以根据内部需求进行二次开发和功能定制, 缺点是性能不如商业系统.

EAST 是基于上世纪末托卡马克最新成果而设计的, 它的目标就是针对近堆芯等离子体稳态先进运行模式的科学和工程问题. 在 EAST 托卡马克实验中产生的大量的实验数据都直接或间接地存入 MDSPlus 数据库^[4]和二级库中. 在 EAST 放电实验过程中, 参与实验的科研人员擅长的领域各不相同, 对实验数据的需求也不一样, 实验过程中实验人员需要实时地获取实验状态、实验结果并进行讨论, 以发现问题或制定下一步实验方案. 现有的基于 Web 的实验数据分析与可视化系统 WebScope^[5]虽然提供了详尽的实验结果查看平台, 但却需要用户主动并有选择地去获取. 针对这种情况, 西南物理研究院提出了基于 HL-2A 实验的即时通信系统 BCOS, BCOS 系统投入使用后取得了不错的效果, 鉴于此构建 EAST 即时通信系统也将具有重要的现实意义. 通过选型调研和需求分析, EAST 即时通信系统采用对开源软件二次开发的方式进行搭建, 这样不仅可以节省人力物力, 也更能专注核心功能的开发. 经过多方面考虑, 最终选择基于

Openfire+Spark 框架^[6], 其中的订阅/推送功能作为 WebScope 的扩展满足了科研人员即时便捷地获取实验信息的需求.

2 系统介绍

2.1 系统介绍

EAST 即时通信系统是在开源即时通信解决方案 Openfire+Spark 上进行二次开发搭建的, 系统框架如图 1 所示.

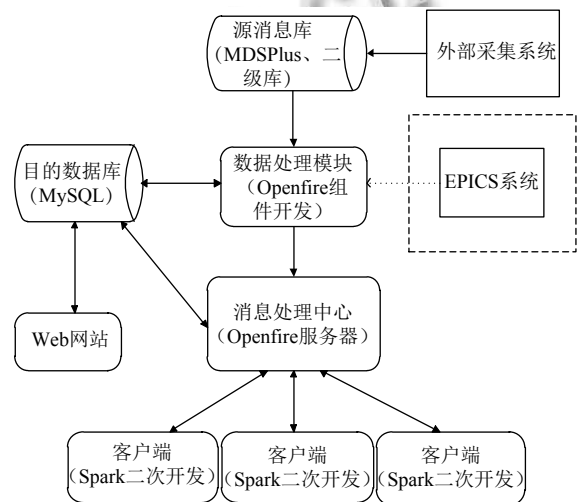


图1 EAST 即时通信系统框图

系统主要由五个部分组成, 分别是数据处理模块、消息处理中心、目的数据库、客户端、Web 网站. 在 EAST 托卡马克实验过程中, 外部采集系统采集实验过程中产生的工程数据并存入 MDSPlus 数据库中, 实验数据存储完后 MDSPlus 将会触发存储完毕事件; 数据处理模块主要实现的功能是在监听到 MDSPlus 数据库每炮 (聚变实验放电一次)^[1]实验数据存储完毕的事件后, 根据当前炮号 (每次实验产生的数据以炮号为单位) 获取需要推送的数据进行处理打包后发送给消息处理中心; 消息处理中心的功能是完成用户间消息的转发以及将数据处理模块发送过来的消息进行相应地推送; 目的数据库是用来存放用户信息、消息类别、用户订阅、历史消息记录等与用户操作相关的表, 数据处理模块、消息处理中心、Web 网站模块通过各自的数据库接口与目的数据库交互, 进行存取等操作; 客户端承担着用户操作接口的角色, 用户通过客户端来使用系统的功能, 与好友互发消息、共享文件, 接收来自服务器的推送消息; Web 网站为用户提供个性化设置, 用户可以通过 Web 端个性化订阅

消息, 设置个人信息, 查找历史消息, 部分有权限的用户还可以进行消息类别的增删改操作.

2.2 框架介绍

EAST 即时通信系统是在开源的即时通信系统解决方案 Openfire+Spark 的框架上进行开发搭建.

其中, Openfire 是开源的、基于 XMPP 协议^[7](可拓展通讯和表示协议)、采用 JAVA 语言开发的即时通信服务器, 具有开源、稳定、高性能等优点, EAST 即时通信系统开发初期选用 Openfire 服务器也是基于此考虑. 此外, Openfire 提供了内部组件与外部组件两种开发方式: 内部组件主要是以 jar 包的形式, 能够直接访问和获取 Openfire 上的资源, 但是依赖于 Openfire, 不适合复杂的业务; 外部组件则是一个独立的应用程序, 以 TCP 形式连接到 Openfire 进行通信. 在 EAST 即时通信系统的开发过程中, 数据处理模块最终选择的是外部组件开发形式, 该种方式优点是应用程序独立于 Openfire 不占用服务器的资源, 适合处理复杂耗时的业务逻辑. 数据处理模块有大量的数据读取和绘图操作, 因此, 若采用内部组件开发的形式, 会增加服务器的负担, 严重情况下, 会造成服务器崩溃. Igniterealtime 官网^[8]提供了 Openfire 外部组件开发包 whack, 可以方便地进行外部组件开发.

Spark 是一个开源的跨平台的即时通信客户端, 具备单聊、群聊功能, 支持文本、图片消息, 在 EAST 即时通信系统中 Spark 扮演着客户端框架的角色, 系统客户端部分是基于 Spark 进行二次开发, 在基本的通信模块基础上, 增加了解析后台推送系统推送的图片消息模块、电子白板、消息配置等功能以满足实验人员的需求, 另外, 对 Spark 的界面布局也进行了相应的调整并删掉了部分多余的功能.

使用 Openfire+Spark 的框架构建内部即时通信系统不仅可以节约时间成本、金钱成本、获得更稳定的服务, 还能够使开发者更专注于核心需求, 开发出高可用性的即时通信软件.

3 系统实现

3.1 系统流程

系统的消息订阅/推送功能整体流程如图 2 所示. 一次完整的订阅推送过程为:

(1) 用户在 Web 网站选择要订阅的消息, Web 系统将用户的订阅操作保存到目的数据库 MySQL 的

用户订阅表 useropt 中;

(2) 一炮实验结束后, MDSPlus 数据库向数据处理模块发送 MdsplusCompleted 事件;

(3) 数据处理模块监听到事件后, 从 MDSPlus 中获取最新一炮的数据进行处理;

(4) 数据处理模块根据目的数据库 MySQL 的用户订阅表 useropt 将处理后消息以及订阅用户 JID(用户唯一标识) 打包成 Message 发送给消息处理中心 Openfire;

(5) 消息处理中心将消息推送给目标用户.

客户端接收到服务端推送的 Message 后, 将其解析为文本或者图片格式在推送面板进行显示. 若用户不在线, 服务器则将该推送消息暂时放入用户的离线消息队列 OfflineQueue 中, 待用户上线后, 再推送给用户.

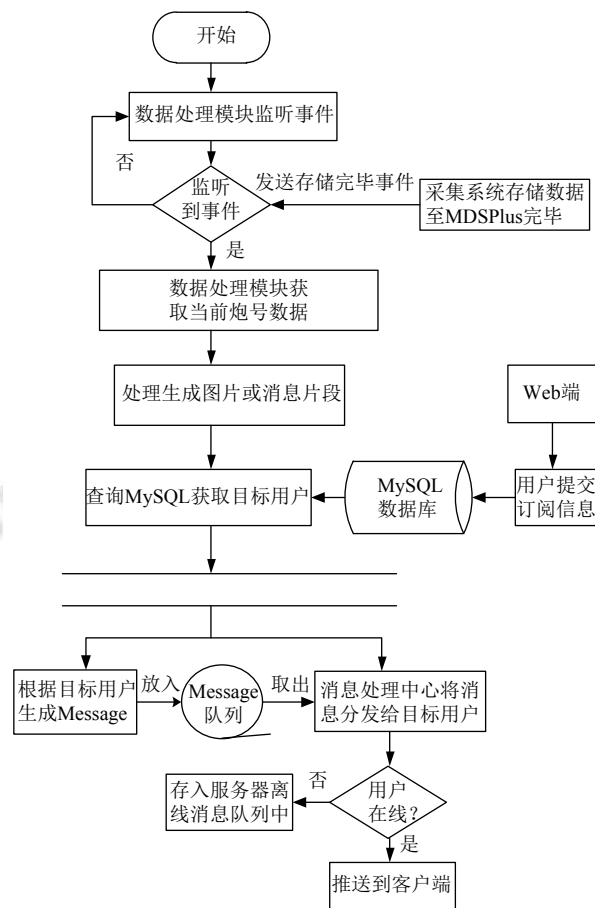


图 2 系统流程图

3.2 消息订阅

在消息订阅系统中, 用户通过与 Web 页面的交互完成指定消息的订阅, 订阅信息将提交至数据库的

户订阅表 useropt 中保存, 后台推送根据此表对用户进行目标推送.

EAST 每炮的实验数据都以信号量的形式保存在 MDSPlus 数据库中, MDSPlus 是一种脉冲驱动的数据库系统, 其数据以树状结构分层存储, 不同类型的实验数据保存在不同的树上^[1], 每个树的节点对应相应存储的信号量. 整个实验中的信号量数量非常庞大, 因此我们选取部分来展示我们信号量的分类过程, 信号量分类示意如图 3 所示. 我们比较关心的是 pcs_east、efit_pcs、east 树上存储的信号量, 其中 pcs_east 树上存储的是与 PCS 相关的数据, efit_east 树上存储的是 efit 计算数据, east 树上存储的是采集到的 EAST 数据. 信号量分类是按照 MDSPlus 树来进行组织的, 这样更符合实验人员的习惯, 更方便地找到要订阅的信号量. 另外, 当树上的信号量太多的时候, 我们进一步对信号量进行了分类, 如图 3 中的单匝环信号、快控电流、等离子体位形等, 这样的组织方式不仅能够方便用户, 后台数据库的组织层次也更加清晰. 最终订阅页面也是按照这种组织关系实现的.

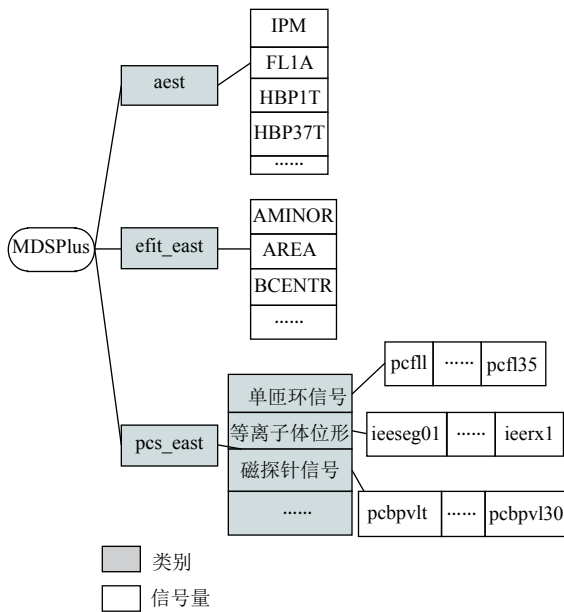


图 3 信号量分类

本系统中消息订阅系统的 web 网站我们采用了 Tomcat+Mybatis+Spring+SpringMVC 技术来进行实现. 其中 Tomcat 作为应用服务器, 运行 JSP 应用; Mybatis 则作为数据库访问的框架, 完成数据库的交互部分; Spring 和 SpringMVC 则用于封装 Web 层的开发, 以达

到低耦合、高效易用的目的.

SpringMVC 的前端控制器将用户请求通过映射的方式, 传给页面控制器进行处理; 页面控制器根据用户的请求类型, 通过调用 mybatis 实现的访问接口, 对数据库进行对应的查询、修改或删除操作; 在完成数据库交互后, 将得到的数据进行相应的整合与进一步的封装, 并对其指定视图名; 最后通过选定的视图控制器对返回的数据进行渲染, 将数据按需求的规格展示在页面上, 供用户访问与获取. 以消息订阅页面为例, 勾选自己需要的消息后, checkbox 的数据将映射到页面控制器, 通过判断每个选项的选中值 (true 或者 false), 将选中的项目更新为 SQL 语句的参数, 利用事先定义好的 mybatis 接口对数据库的订阅项进行更新, 更新结束后, 指定返回到订阅页面, 并以 tree 结构展示更新后的订阅结果. 此外, 整个工程通过 Maven 对依赖的 jar 包进行配置和管理, 避免了繁琐的依赖项导入.

数据库我们使用 MySQL 来进行存储, 主要包括用户信息和消息描述类信息. 其中, 与订阅/推送相关的关键表有用户订阅表、signal 类别表、signal 描述表等, 其中 signal 描述表和用户订阅表的结构如表 1 和表 2 所示. 类别表之间通过外键约束实现级联更新和删除, 类别表与 signal 描述表则使用了触发器来进行级联. 此外, 为了防止并发修改带来的影响, 我们采用了乐观锁机制对表的修改进行了版本控制, 即增加了一个 version 字段, 当修改时 version 与查询到的 version 不同时, 表示修改冲突, 则放弃此次修改, 否则修改成功.

表 1 signal 描述表

字段名	类型	是否为空	说明
id	int	否	id 号
mname	varchar	否	信号量名称
dscr	varchar	是	信号量描述
unit	varchar	否	单位
meaning	varchar	否	物理意义
belongs	varchar	否	用于分类
belongslevel	int	否	用于分类
version	int	否	版本控制

表 2 用户订阅表

字段名	类型	是否为空	说明
id	int	否	id 号
name	varchar	否	用户名
mname	varchar	否	信号量名
belongslevel	int	否	信号量所属 level

3.3 数据处理与推送

鉴于系统的扩展性,在设计消息类的时候将消息的共同行为和属性抽象出来,设计 MessageBase 类作为消息类的父类,其它消息类都继承这个类,再添加自己的属性和方法,如图 4 所示,其中 SignalMessage 类表示信号量类型的消息,EpicsMessage 类表示 EPICS 相关的消息类型,这样可以降低消息存储与处理间的耦合性。

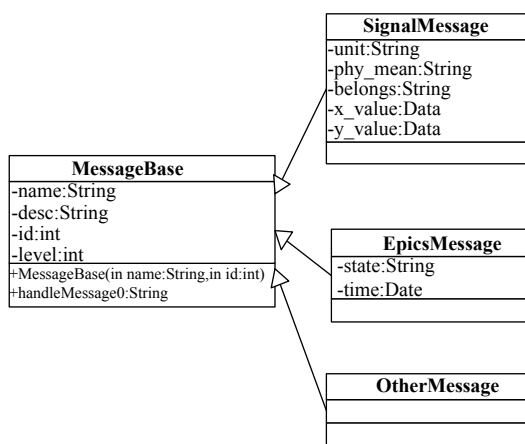


图 4 消息类设计

EAST 即时通信的数据处理模块采用 Openfire 外部组件开发形式,组件是独立于服务器运行的,即组件的运行状态不会影响 Openfire 服务器的运行,这也是采用组件进行二次开发的重要原因,一方面不占用服务器的资源来处理数据,另一方面组件不依赖于服务器,组件崩溃或者启动/关闭都不需要对服务器采取进一步操作,确保了服务器的稳定性。

数据处理模块的功能主要包括监听源消息库的新一炮存储完毕的事件、从源消息库中读取数据、处理数据、转发生成的 Message。数据处理流程如图 5 所示。

数据处理模块主要由四个线程组成,监听线程用于监听 MDSPlus 数据库是否有新炮号存储完毕(当存储完毕的时候会触发 MdsplusCompleted 事件),当监听到对应事件后,数据处理模块的取数据、处理数据、发送消息操作即采用多线程机制以异步的方式进行,读线程通过 MDSip 协议远程访问 MDSPlus^[1],从中读取数据并放入数据阻塞队列 DataQueue; 处理线程从 DataQueue 中取出数据进行绘图处理,然后根据用户订阅表 useropt 将处理后的数据和订阅用户的 JID 打包成 Message 放入消息队列 MessageQueue; 最后,发送

线程从消息队列里取出 Message 转发给消息处理中心 Openfire。整个过程是一个典型的生产者-消费者模式,读线程是数据的生产者,处理线程是数据的消费者同时也是 Message 的生产者,发送线程是 Message 的消费者。其中,DataQueue 和 MessageQueue 是通过 Java 阻塞队列 BlockingQueue 实现,二者都是固定大小的阻塞队列,当线程向队列里存储元素的时候,若队列满则线程阻塞直到队列有空闲位置;当线程从队列里获取元素的时候,若队列为空则阻塞等待直到队列里有新的元素。发送线程在将 Message 发送给处理中心的时候,在 Openfire 服务可用的情况下,则代表该条 Message 消费成功,Openfire 的消息可靠性机制保证了这点;Openfire 服务若不可用,Message 的可靠性由数据处理模块保证,发送线程在向 Openfire 发送消息的时候,如果连接不可用,将抛出异常,Message 进一步交给异常处理程序处理,异常处理程序将其放入 TempQueue 里,并不断重发直到发送成功。客户端接收订阅消息功能实现如图 6 所示。

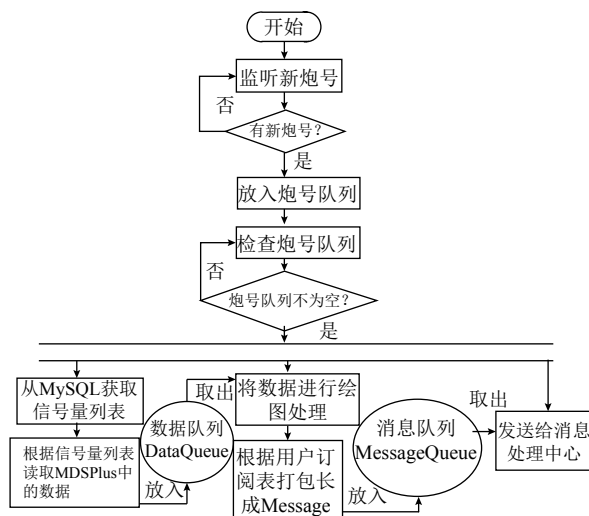


图 5 数据处理流程图

数据处理模块是以炮号为单位进行处理,即处理完一炮的数据并全部发送成功后才会处理下一炮的数据,因此在整个处理过程中可以很容易地对处理状态进行标记,Message 发送成功也会有对应的消息发送成功标记。在数据处理模块因为某种原因重启时首先会检查状态记录表,如果上一炮的数据没有完全处理(即没有处理完毕的状态记录)且距离当前时间不超过十分钟(由于实验数据的时效性,时间间隔久即失去了推

送的意义),先读取该炮的实验数据进行处理,推送的时候根据状态记录表将未推送的消息重新推送,已成功推送的消息舍弃,该方式在一定程度上确保了消息送达用户.

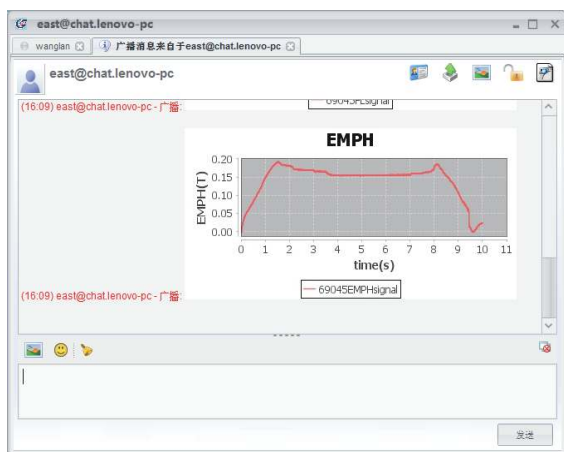


图6 用户接收推送消息

4 结论与展望

本文主要介绍了基于 Openfire+Spark 框架的即时通信系统中订阅/推送功能的设计与实现过程.该功能是 EAST 即时通信系统的核心功能,根据实验人员的需求进行消息定制和推送,实验人员获取信息的方式由传统的主动检索转变为即时推送,这不仅方便了实验人员查看实验结果,也提高了消息的利用率.与此同

时,该即时通信系统也存在很多需要改进的地方,系统目前处理的订阅消息种类仅有信号量类型,比较单一,后续将根据实际情况接入 EPICS 系统消息,由于系统在最初消息类设计时就考虑到后续的扩展情况,因此增加消息类别将比较方便.消息丢失问题也还需要进一步研究高效的方法,另外,消息推送算法随着消息量的增加也需要进一步的优化.下一步的工作重点是进一步完善该即时通信系统.

参考文献

- 1 杨飞,肖炳甲,刘连忠,等.基于 MDSplus 的 EAST 工程数据获取及显示.微计算机信息,2010,26(13):20-22. [doi: 10.3969/j.issn.2095-6835.2010.13.008]
- 2 李斌,肖炳甲,刘连忠.基于 Java 的 EAST 实验数据可视化系统.计算机应用,2010,30(S1):248-250.
- 3 盖荣丽,钱玉磊,李鸿彬,等.基于 MQTT 的企业消息推送系统.计算机系统应用,2015,24(11):69-75. [doi: 10.3969/j.issn.1003-3254.2015.11.011]
- 4 MDSplus. <http://www.mdsplus.org/>.
- 5 杨飞. EAST 实验数据系统研究[博士学位论文].北京:中国科学院研究生院,2011.
- 6 沈奎林,杜瑾.基于 Openfire+Spark 构建 IM 实时交流平台.现代图书情报技术,2011,(5):83-87. [doi: 10.11925/infotech.1003-3513.2011.05.13]
- 7 Saint-Andre P. Extensible messaging and presence protocol (XMPP): Core [Thesis]. Helsinki: Department of Computer Science, University of Helsinki, 2011.
- 8 Ignite realtime. <http://www.igniterealtime.org/>.