

# Automated Conference Network

SIP Conference Server – Load balancing

---

**Huwei Zhang**

**IRT CINEMA Group**

# Automated Conference Network

## SIP conference server – Load balancing

---

### Background:

CINEMA need some mechanism to implement the load balancing in the SIP conference servers to achieve better performance. When the participants in the conference grows, we can dispatch the participants to the least loaded server and bring up new server when it is necessary.

### Long term goal of the project:

Build a system which automatically adds new servers and transfers existing participants to new servers as the group size grows. The new servers should be chosen to be well-positioned in respect to the members to be transferred, e.g., by using UDP round trip delay, path length or quality of service measurements.

### Short term goal of the project:

Find the server for the conference user from the conference servers in the system only based on the conference server load.

# Overview

SIP Conference Server – Load balancing

---

## *Overview*

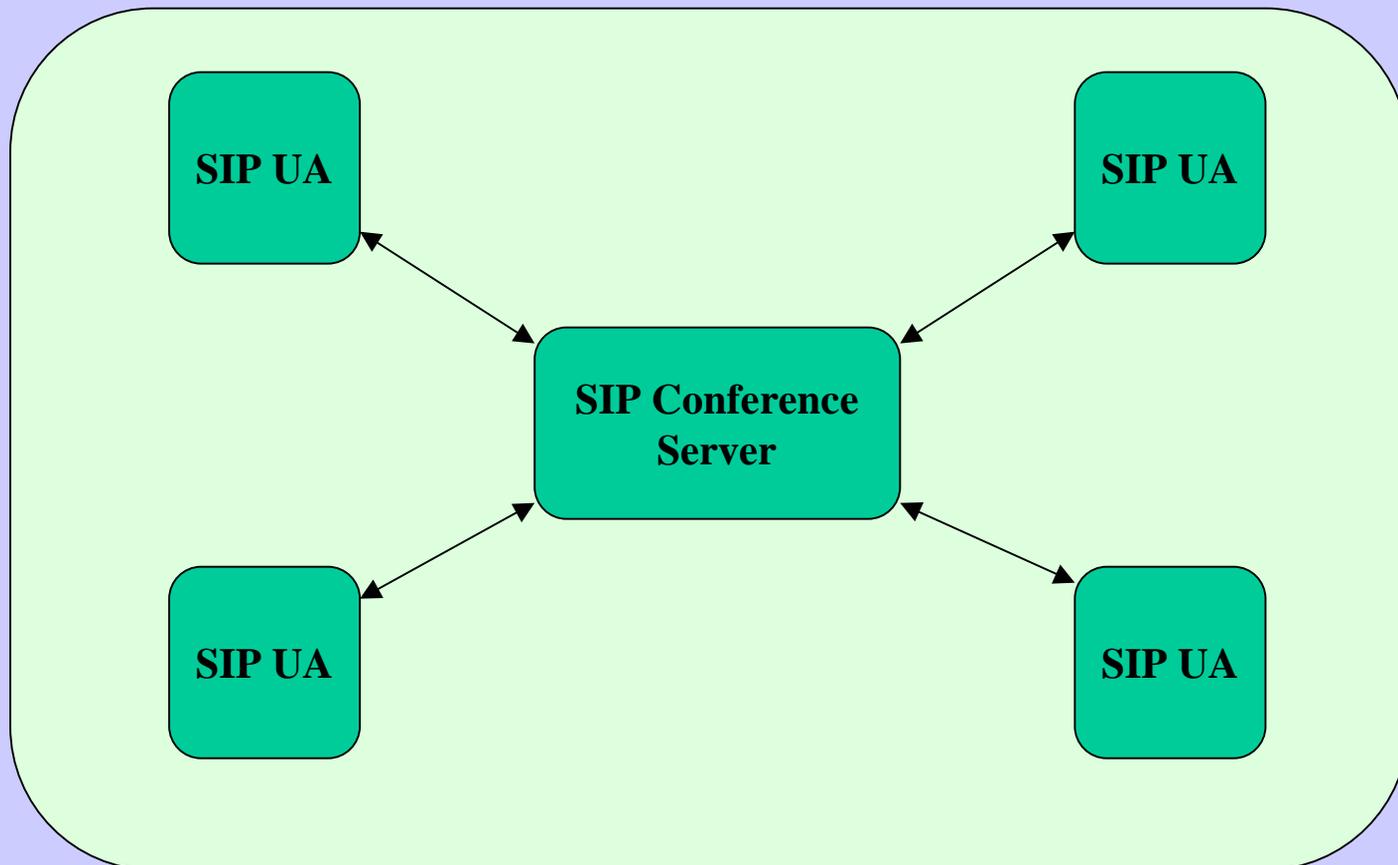
- **Background**
- **Design Architecture**
- **Basic Algorithm**
- **Running Sample**
- **Future work**

# Background

SIP Conference Server – without redirect service

---

## Conference Server System

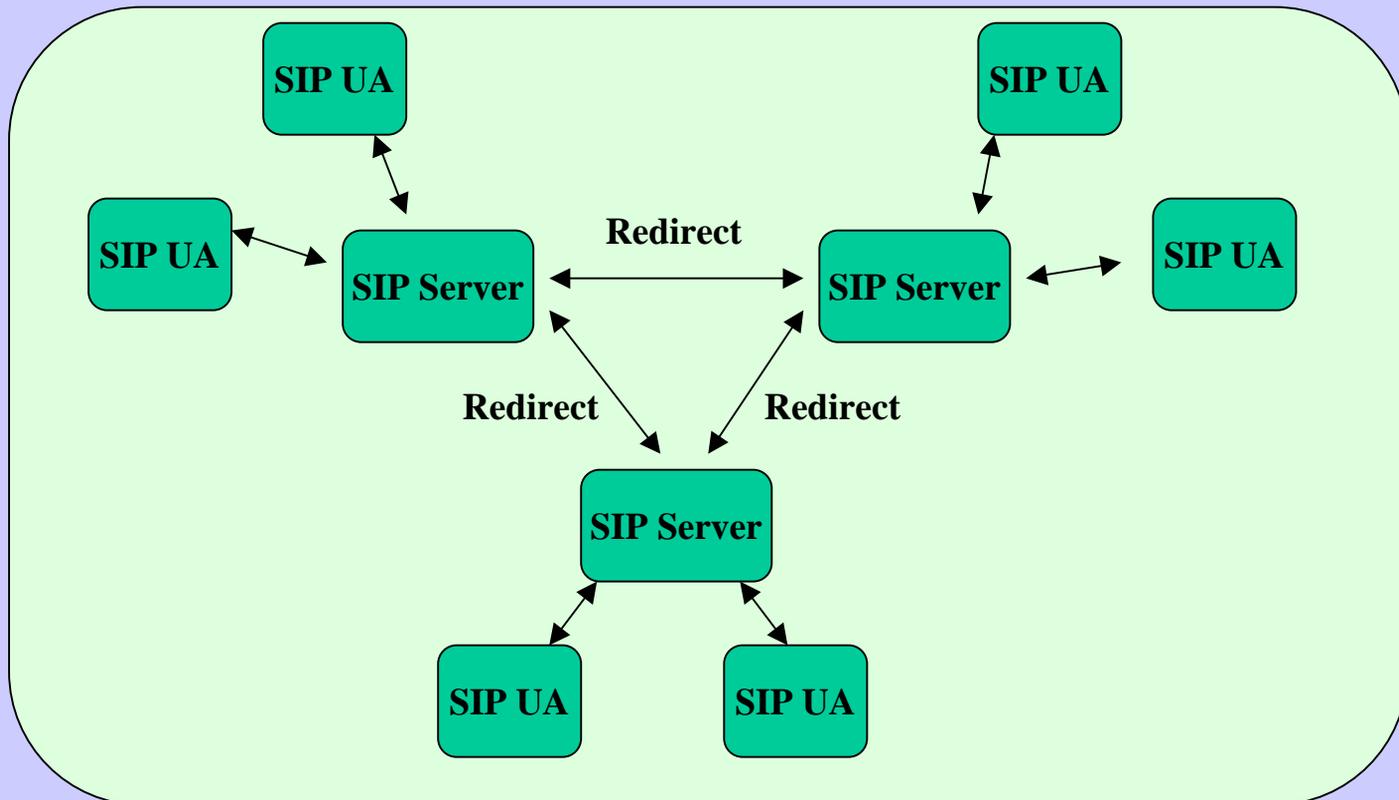


# Design Architecture

## SIP Conference Server with Redirect Service

*A distributed architecture to implement load balancing in the conference server system using REDIRECT service.*

### Conference Server System

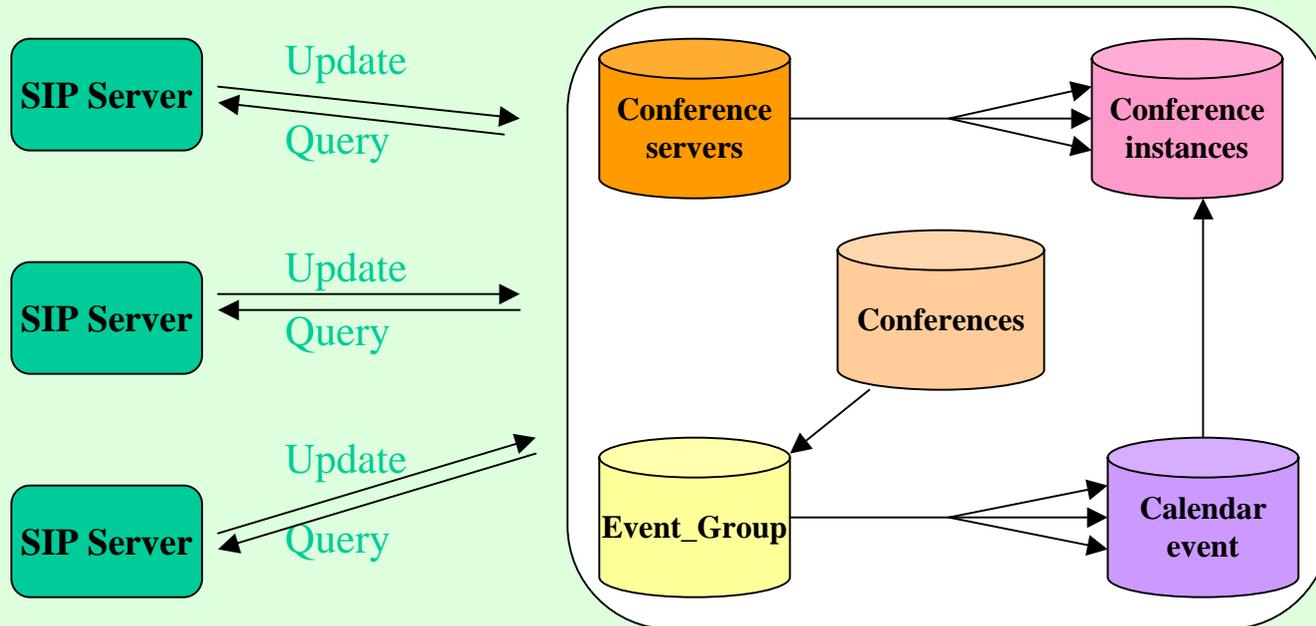


# Design Architecture

## SQL database design - SIP Conference Server

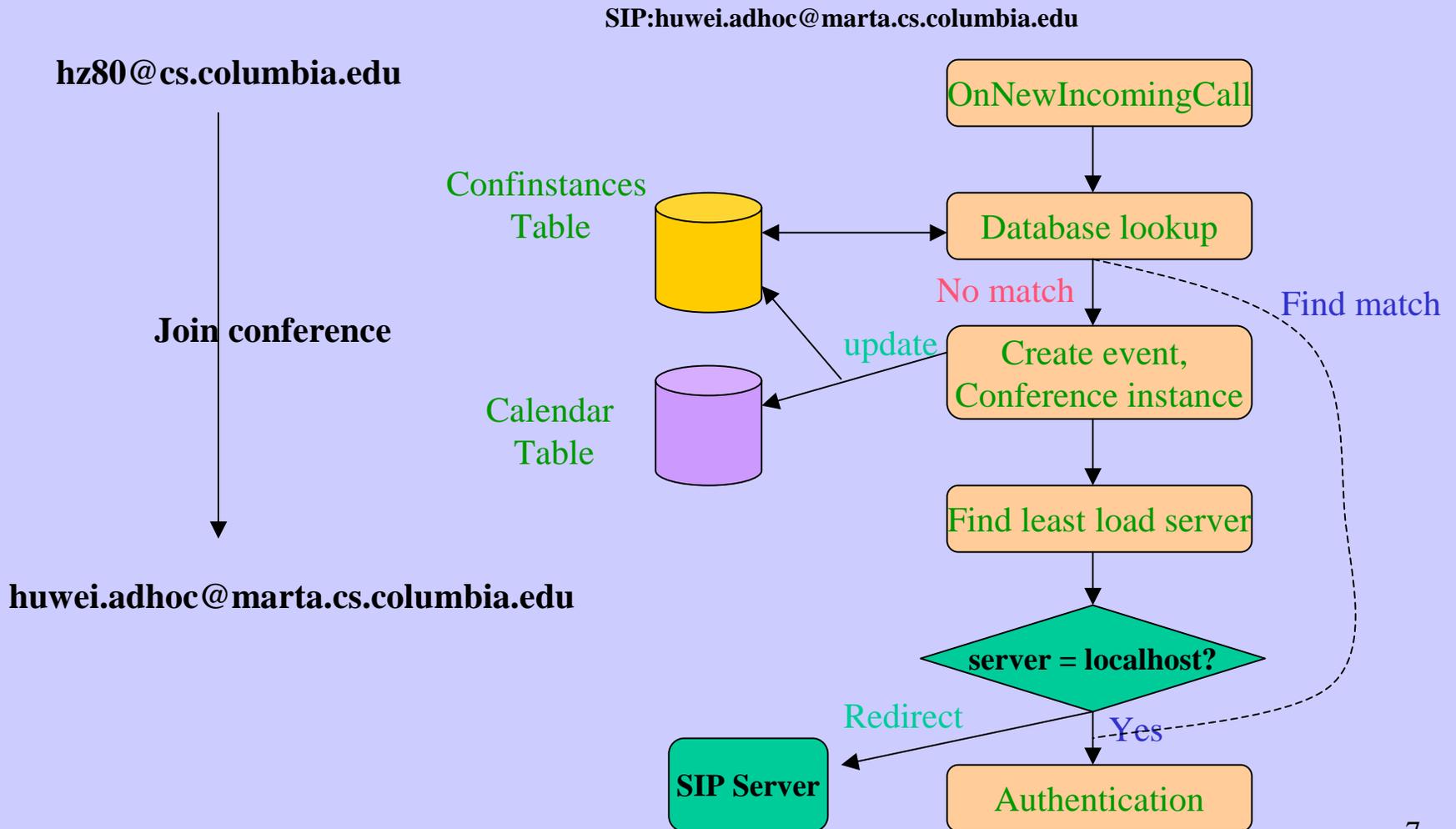
*Add new database table to store the server information, conference instance information, and record QOS information to support decision make.*

**Confservers, Confinstances, Conferences, Calendar, Event\_Group**



# Basic Algorithm

Scenario I - "ad hoc" conference: Find the least loaded server

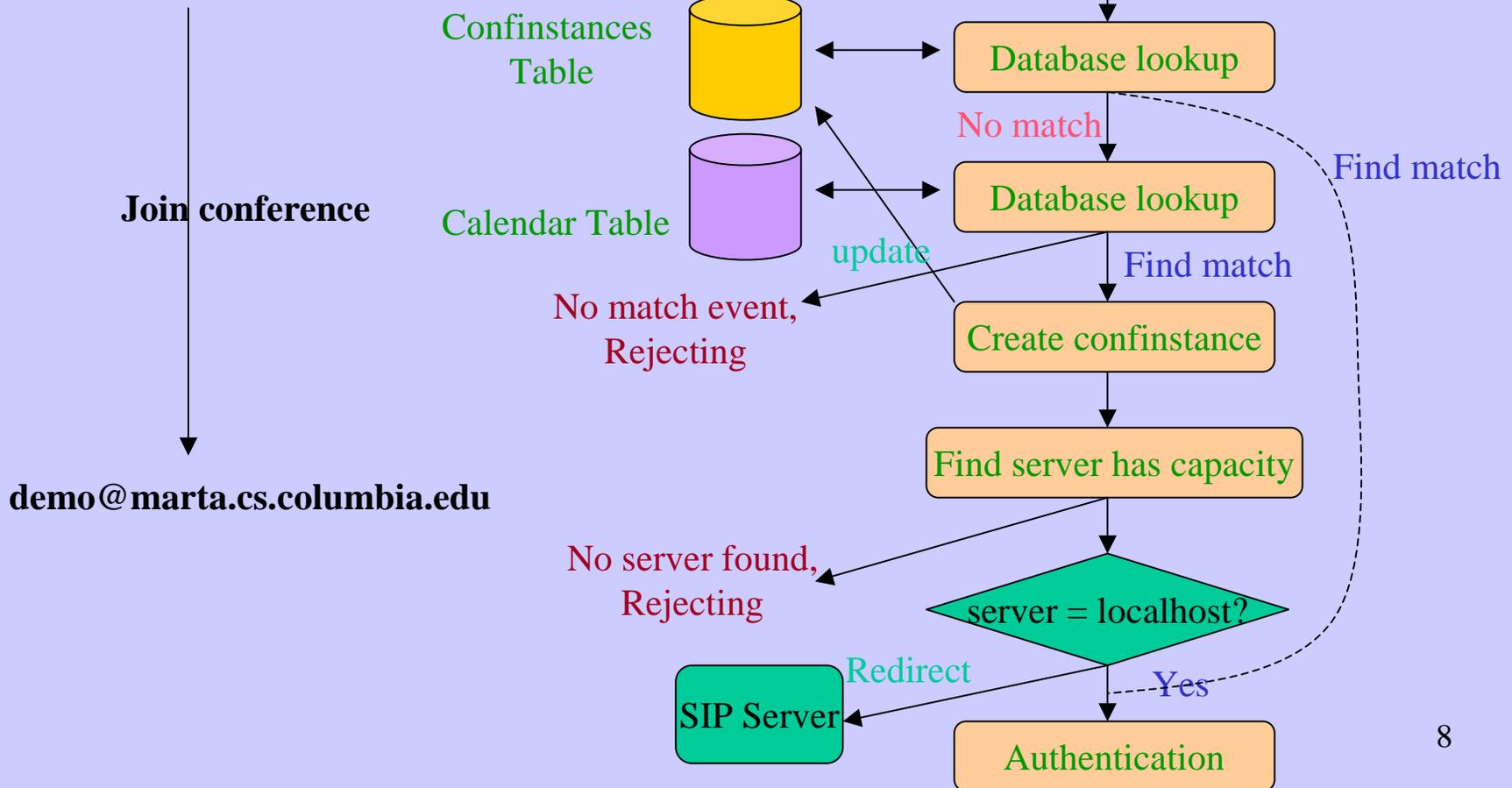


# Basic Algorithm

Scenario II, III: Find the server has capacity  $\geq$  bandwidth

SIP:demo@marta.cs.columbia.edu

hz80@cs.columbia.edu



# Basic Algorithm

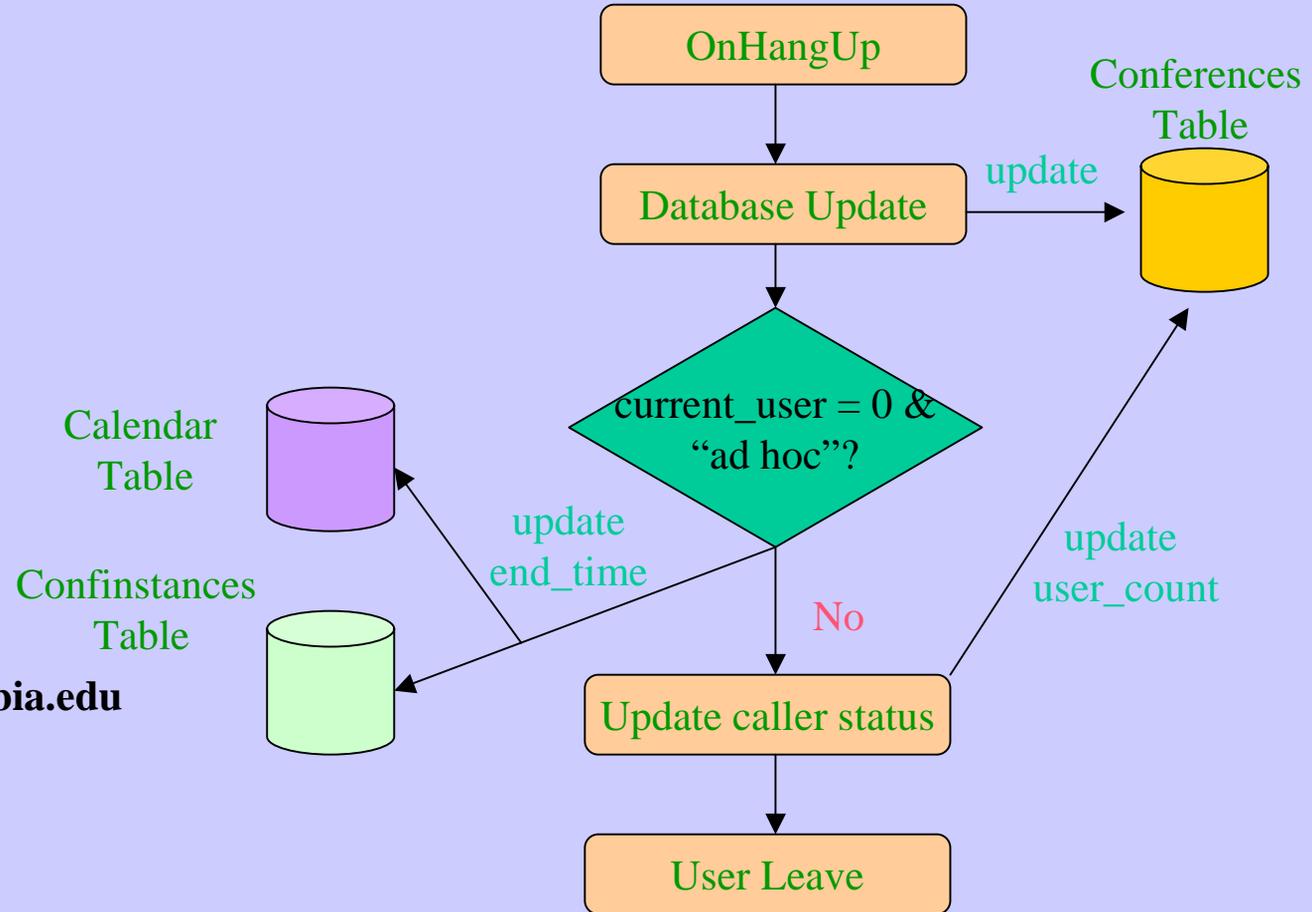
Scenario I, II, III: OnHangUp

SIP:huwei.adhoc@marta.cs.columbia.edu

hz80@cs.columbia.edu

Leave conference

huwei.adhoc@marta.cs.columbia.edu



# Running Sample

## SIP Conference Server – Load balancing

---

### Setting up

1. Conference servers: marta.cs.columbia.edu (50), cisalpino.cs.columbia.edu (50)
2. Conferences: huwei.adhoc, hz80.adhoc, demo (10), test (1000), test-demo (2)
3. User : hz80@cs.columbia.edu, huwei@cs.columbia.edu

### Test Case

1. hz80 join huwei.adhoc@marta.cs.columbia.edu
2. hz80 join hz80.adhoc@marta.cs.columbia.edu
3. huwei join huwei.adhoc@cisalpino.cs.columbia.edu
4. hz80 join demo@marta.cs.columbia.edu
5. huwei join test@marta.cs.columbia.edu
6. huwei join test-demo@marta.cs.columbia.edu

### Running Result

1. Least load server = marta, marta host the conference “huwei.adhoc”
2. Least load server = cisalpino, user is **REDIRECT** to cisalpino which host “hz80.adhoc”
3. huwei.adhoc already has instance on marta, user is **REDIRECT** to marta
4. cisalpino has maximum left capacity (48), user is **REDIRECT** to cisalpino which will host the conference “demo” (bandwidth = 10)
5. Bandwidth for “test” = 1000, no server has enough capacity, **Reject** the call
6. No event found for conference “test-demo”, “**Not in the scheduled time**”, **Reject call**.<sup>10</sup>

# Future Work

## SQL database design - SIP Conference Server

***Add new database table to store the Quality of service information by recording the RTCP packets for each conference user. Decision make based on path property and Qos information.***

**Confservers, Confinstances, Conferences, Calendar, Event\_Group, Confqos**

