IBM Caribbean Project

IBM Tokyo Research Laboratory Agent Project

Caribbean: Technology of the Agent Server capable of Hosting Large Number of Agents

July 2001

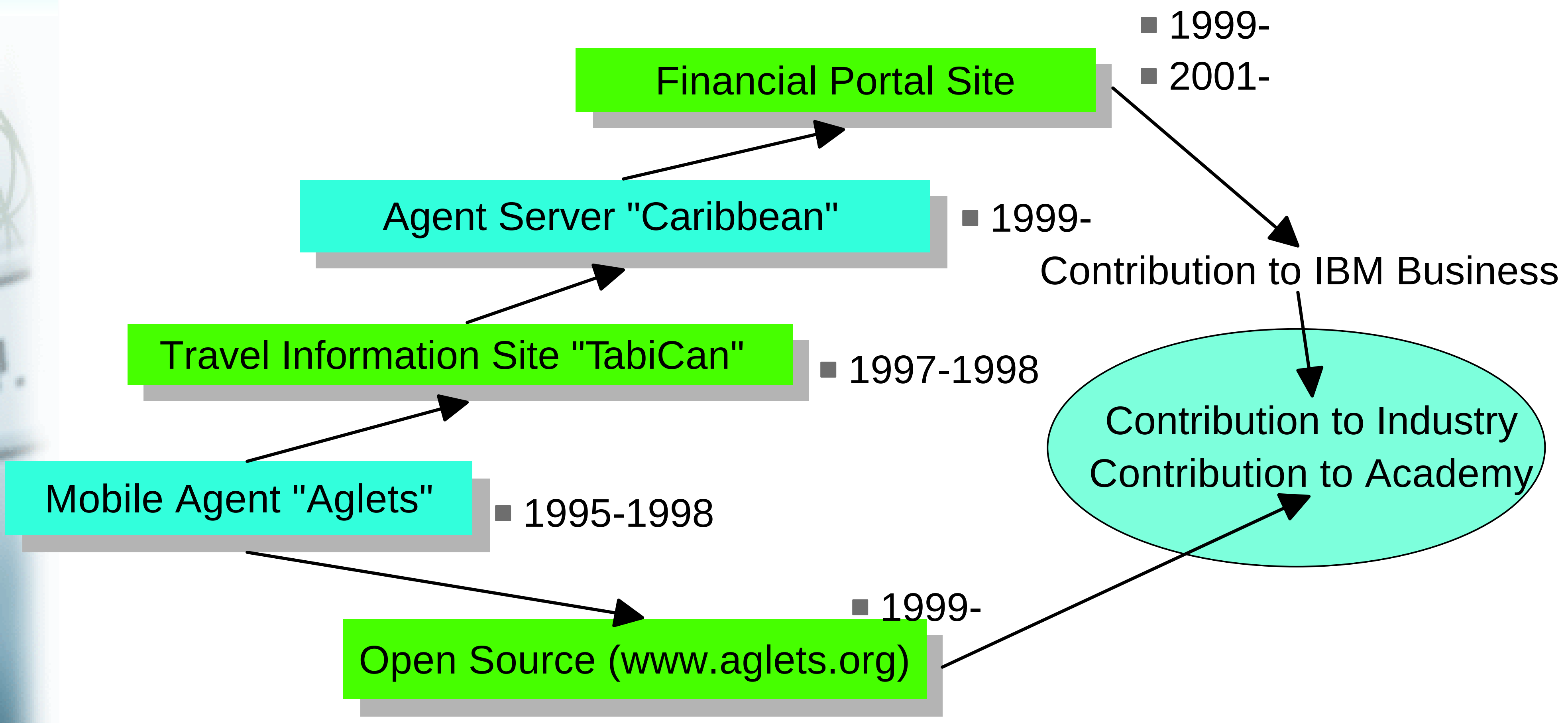
Gaku YAMAMOTO

yamamoto@jp.ibm.com

IBM Research,

Tokyo Research Laboratory

IBM Tokyo Research Lab. Agent Projects



IBM Caribbean Project

Change of Our Focul Points

Done

Mobile Agent "Aglets"
- Mobility
- Security

→ Research Purpose

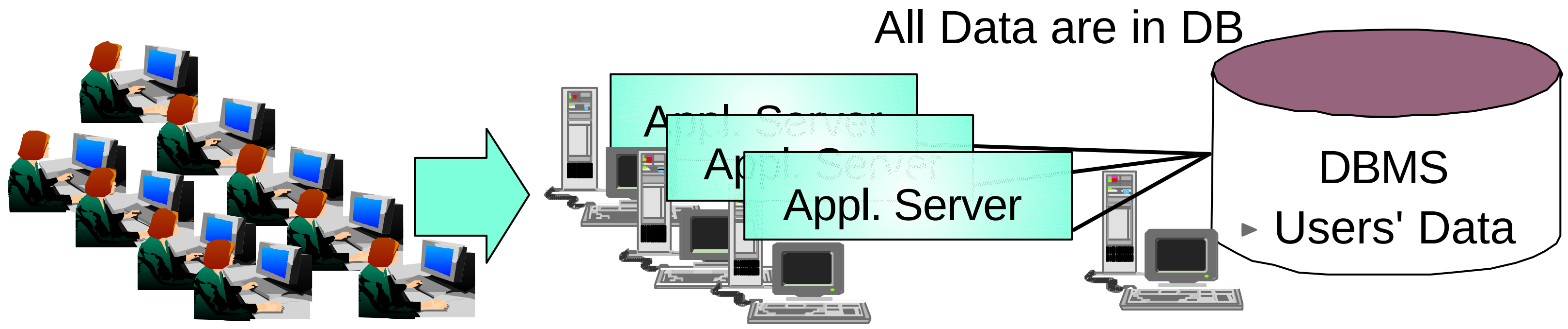
Agent Server "Caribbean"
- Agent Capacity
- High Performance
- Reliability

→ Business Purpose

On Going

Background

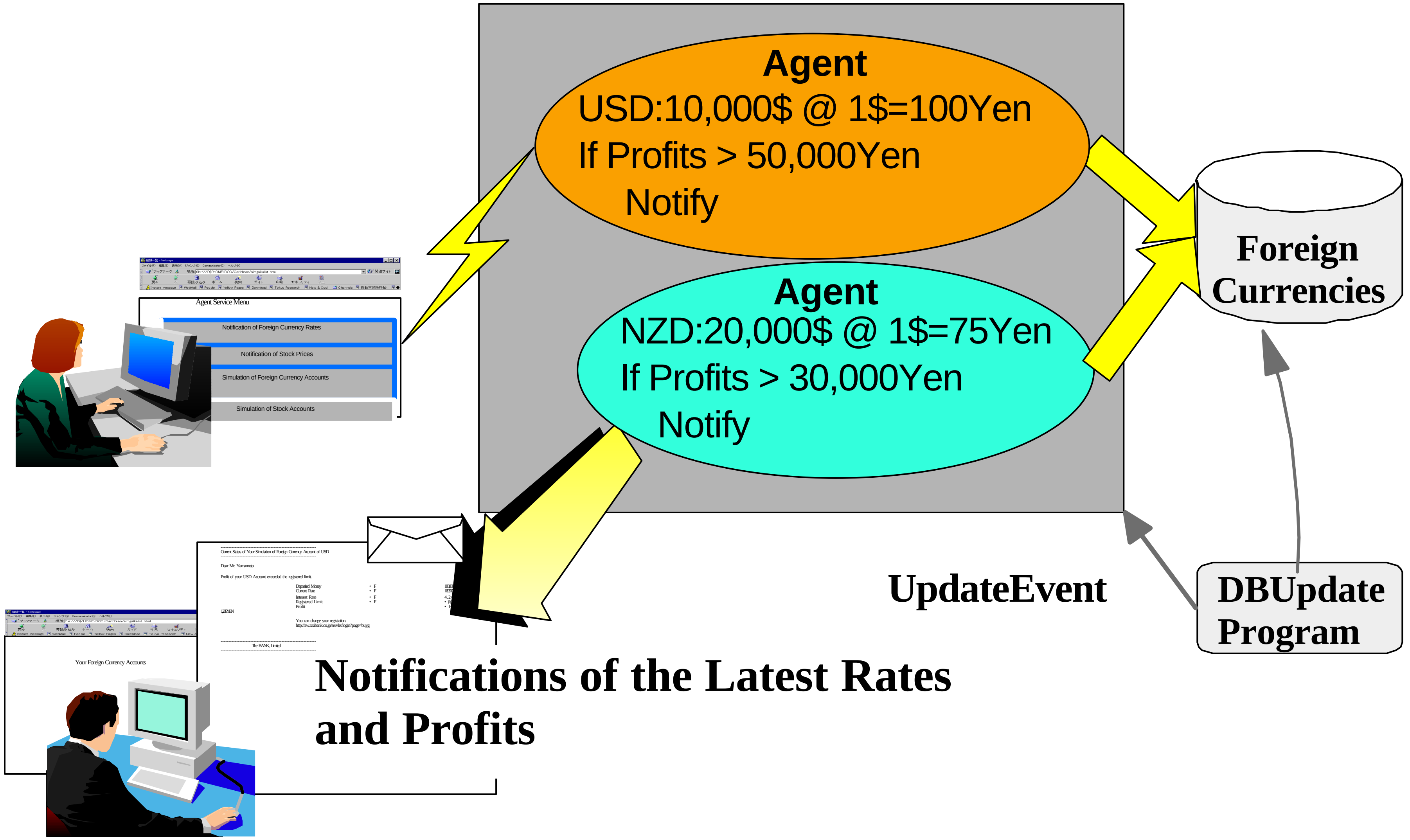
- Web applications become complex
  - Provide Notification services as well as Request-Response services
  - Perform tasks accessing users' data stored at a server
- DBcentric systems are not high performance
  - DBMS load to access individual user's data is heavy
  - High performance DBcentric systems are expensive



➡ We need a new high performance system architecture

Application Scenario Example

Financial Asset Simulation System


Agent
USD:10,000$ @ 1$=100Yen
If Profits > 50,000Yen
Notify
Agent
NZD:20,000$ @ 1$=75Yen
If Profits > 30,000Yen
Notify
Foreign Currencies
DBUpdate Program
UpdateEvent
Agent Service Menu
Notification of Foreign Currency Rates
Notification of Stock Prices
Simulation of Foreign Currency Accounts
Simulation of Stock Accounts
Your Foreign Currency Accounts
Notifications of the Latest Rates and Profits

Current Status of Your Simulation of Foreign Currency Account of USD

Dear Mr. Yamamoto

Profit of your USD Account exceeded the registered limit.

Deposited Money • F 100,000
Current Rate • F
Interest Rate • F 4.2
Registered Limit • F
Profit • F
1,205YEN

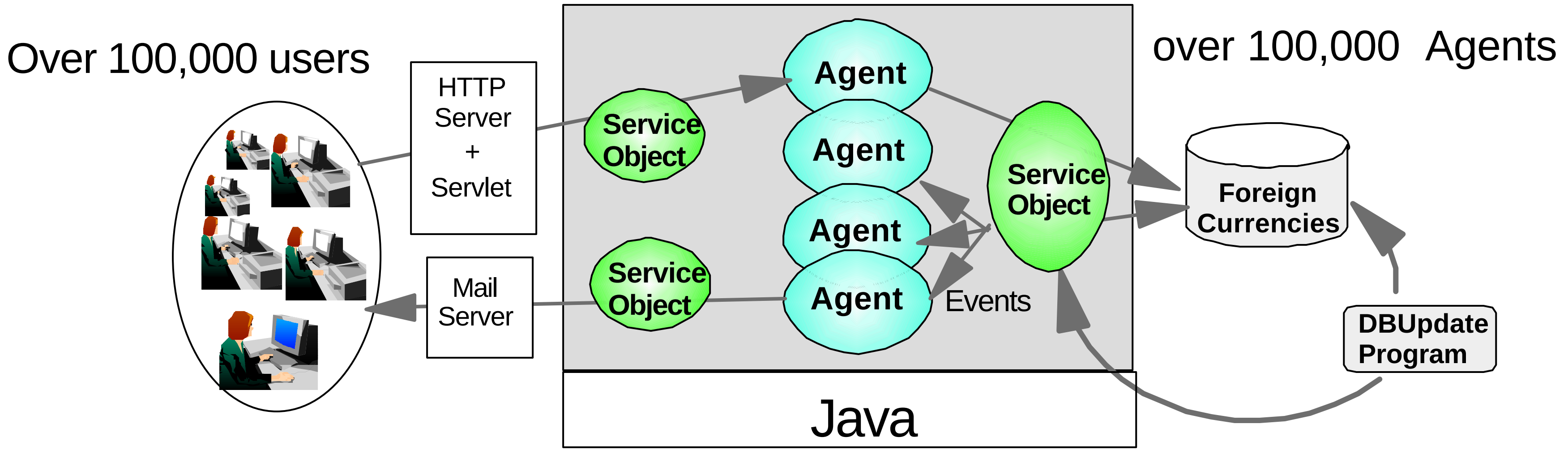
You can change your registration.
http://aw.xxibank.co.jp/servlet/login?page=buyg

The BANK, Limited

Java based Agent Server "Caribbean"

- Agent
  - Event-Driven Coarse-grained Object ("Reactive and Lightweight Agent")
  - Created for each user and stays at a server long time
  - Keep an user's data
  - Exchange messages with other agents in asynchronous manner
  - Accesse B/E systems and DBMSs through "Service Object"
- Agent Server manage hundreds of thousands of agents
  - Achieve high performance by keeping agents in physical memory

A Financial Portal Site using an Agent Server



Framework of Caribbean

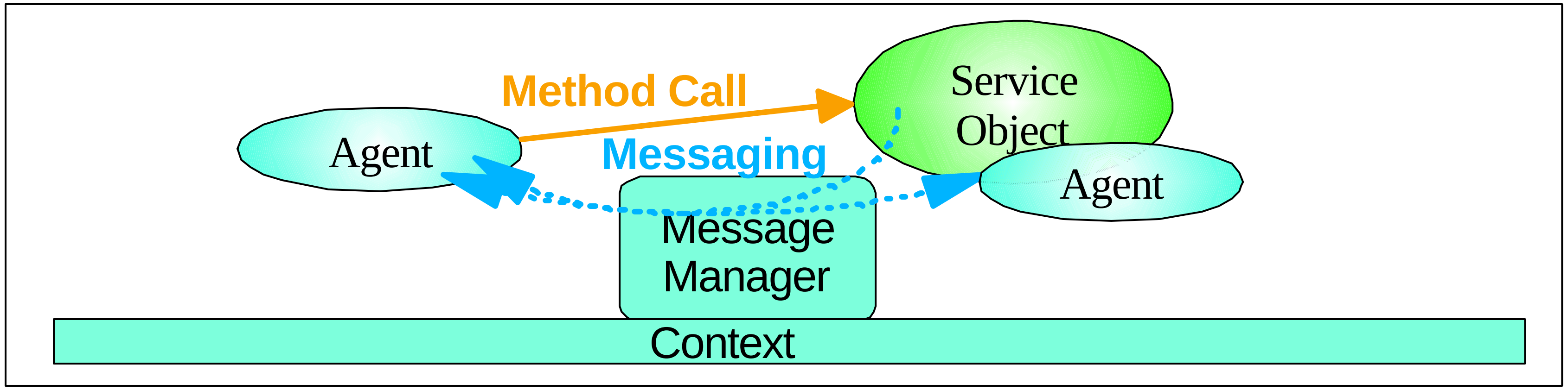
```
public abstract class ObjectBase implements Serializable {
        public boolean handleMessage(SID sid, OID sender, Message msg, MessageManager mng) {}
        public abstract void onCreation(Object args);
        public abstract void onDisposing();
        public abstract void onActivation();
        public abstract void onDeactivating();
        ...
}
```

```
public abstract class Context {
        public abstract OID create(String classname, String group, Object args);
        public abstract OID[] getAllOIDs(String group);
        public abstract SimpleMessageManager getSimpleMessageManager();
        public abstract ServiceObjectBase lookupService(String service);
        ...
}
```

```
public abstract class SimpleMessageManager extends MessageMan
        public void post(SID sid, Message msg) {}
        public void post(SID[] sids, Message msg) {}
        public SID startSession(String session, OID oid) {}
        public SID[] startSessions(String session, OID[] oids) {}
        ...
}
```

```
public class Message implements Serializable {
        public Message(String type) {}
        public void setArg(String name, Object value) {}
        public Object getArg(String name);
        ...
}
```

```
public abstract class ServiceObjectBase extends ObjectBase {...}
public abstract class MessageManager {...}
and others
```



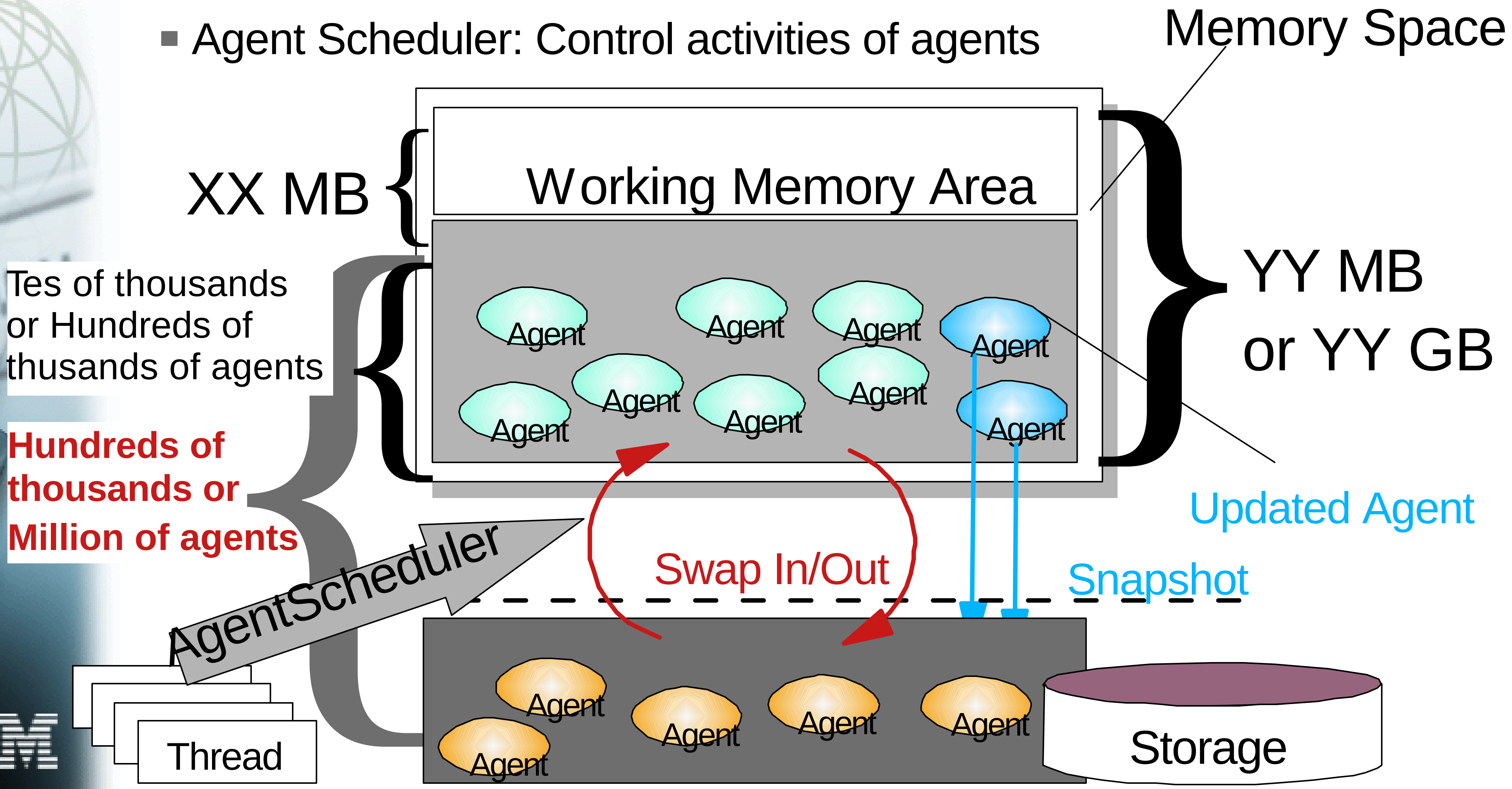
Technical Problems

- Memory Management
  - Agents are basically in memory, however too many agents may break memory limitation
- Agent Persistency
  - Agents in memory may be lost because of system failure
- Agent Scheduler
  - Assign threads to agents to maximize performance

IBM Caribbean Project

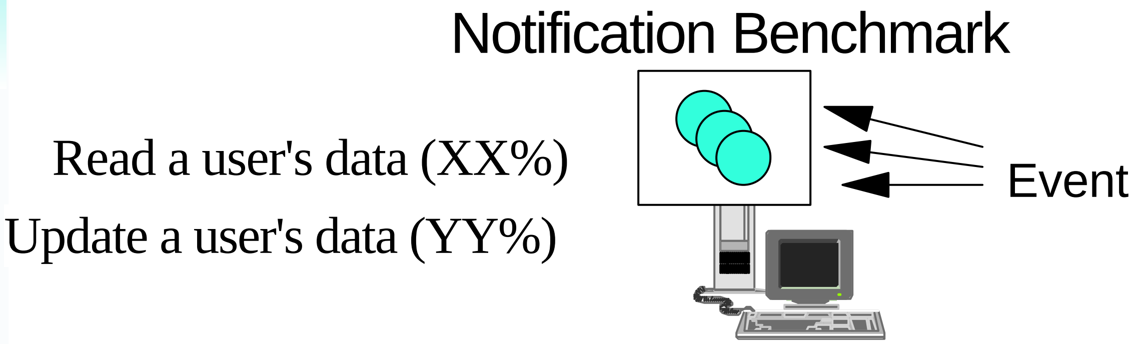
Agent Management Mechanism

- Memory Control: Swap agents in and out
- Agent Persistency: Take snapshot of agents
- Agent Scheduler: Control activities of agents

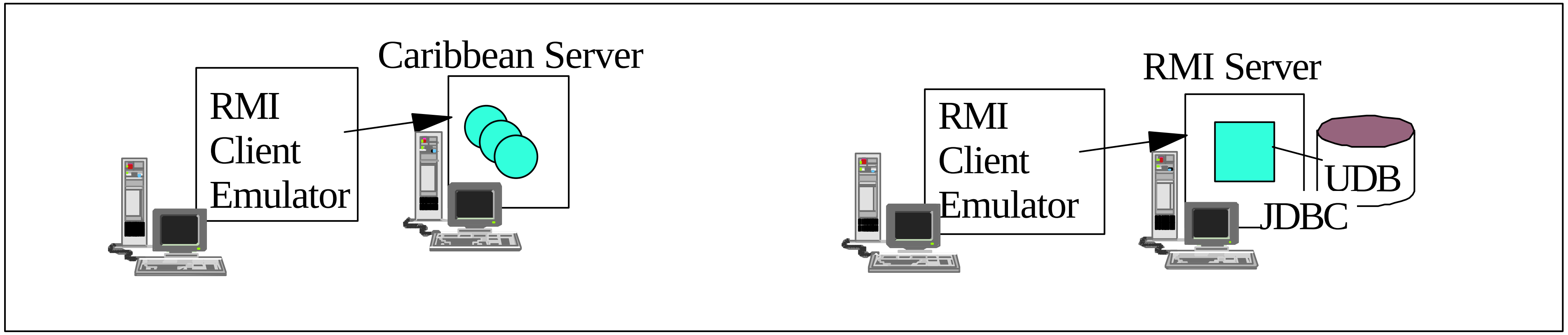


IBM

Performance of Caribbean



PentiumII 600MHz, 1GB memory, WindowsNT 4.0, IBM JDK1.1.8
100,000 users
A user's data size = 5KB



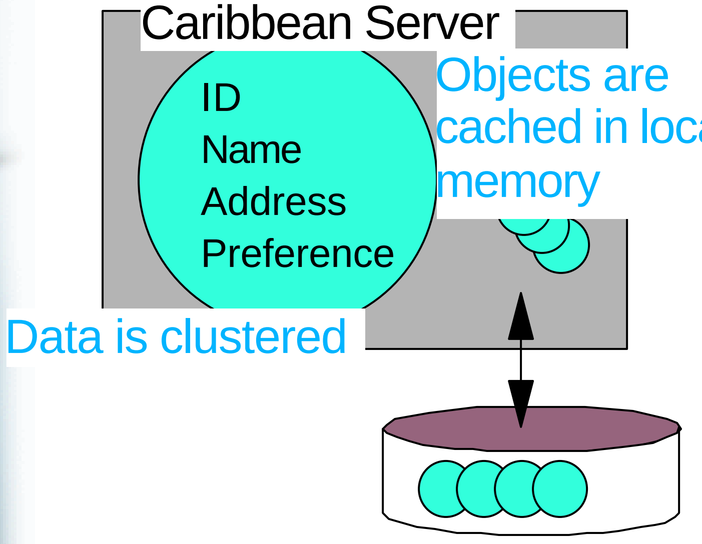
processes/sec

	100% read	50% read 50% update	100% update
A Caribbean System	22301.50	2077.90	1,069.33
DBcentric System	168.05	125.40	100.61

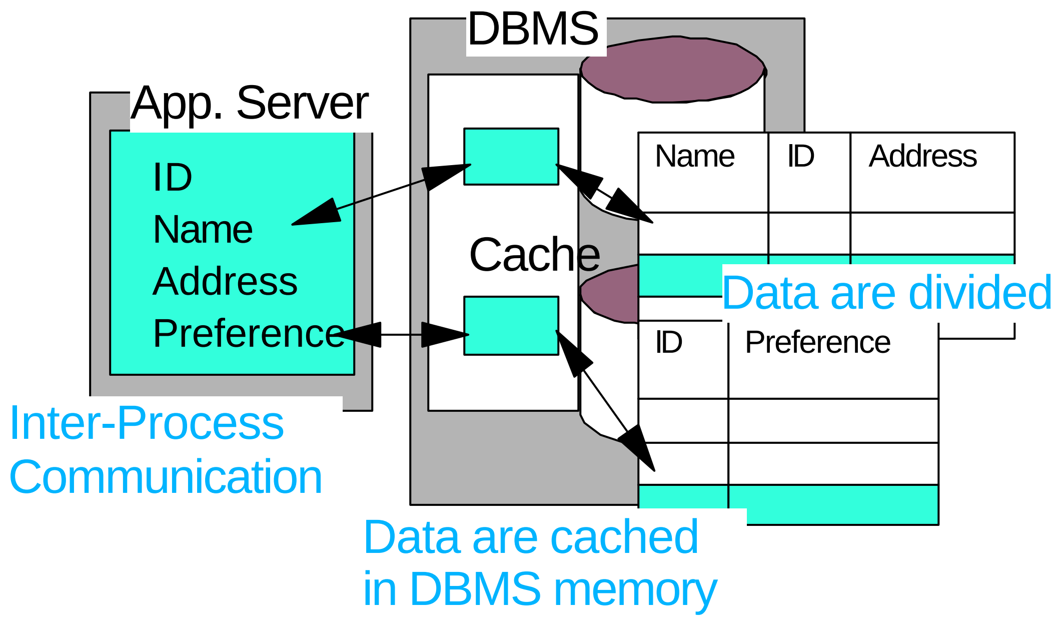
Why is Caribbean Agent Server fast?

- Agents are kept in local memory of a server
- Data related to a user is clustered in an agent

A Caribbean System

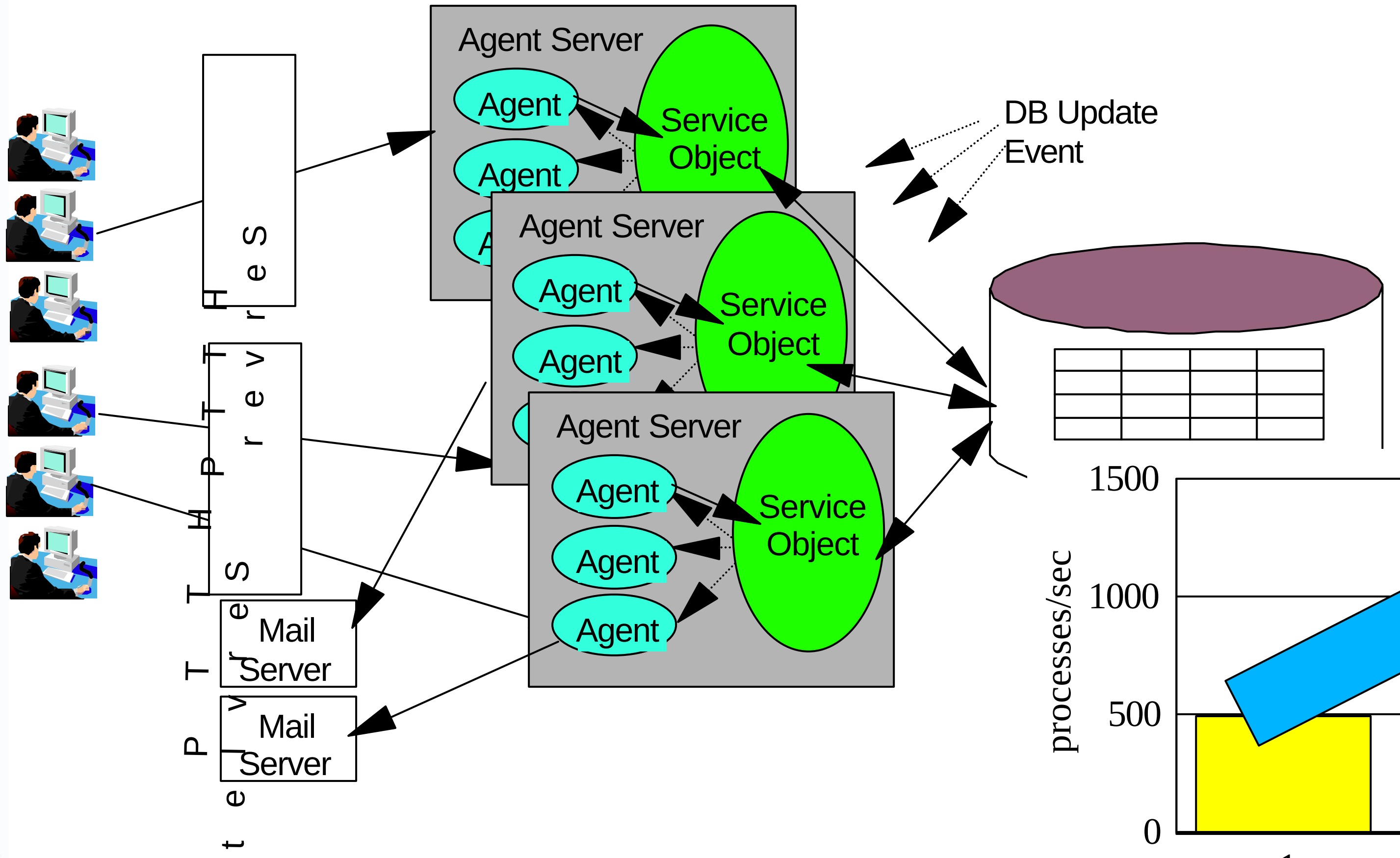


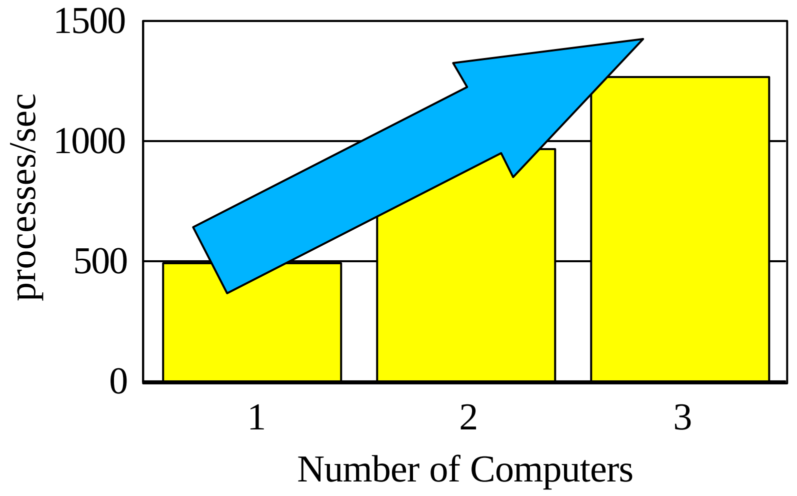
A DBcentric System



Clustered Agent Server

- Clustered Agent Servers enable to develop large systems
  - Support over millions of users
  - Enable highly scalable systems





IBM Caribbean Project

Summary

Agent Server

- Is an application server based on "Agent-oriented Programming Model"
- Provide a framework and runtime for developing high performance systems that use computer resources efficiently
- Achieve high performance by keeping agents in local memory
- Provide reliability for commercial systems
  - Agent Swapping Mechanism
  - Agent Persistency Mechanism
  - Agent Scheduler

Important for Commercial Systems

Caribbean Project Status and Plan

- Published Caribbean v2.4 as a Solution Core S/W included in IBM Japan SI projects
- Deploy the Agent Server Technology through customer's commercial systems
  - Already adapted to two commercial systems of major Japanese Banks
  - Proposing to several customers
- Next Step
  - Develop a system that supports millions users