



Introduction

ความท้าทายของนักวิศวกรซอฟต์แวร์ คือ

“การหาวิธีที่จะสร้างแอปพลิเคชัน ที่จะอำนวยความสะดวกในการติดต่อสื่อสาร และการกระจายสินค้า/บริการในระดับมวลชนโดยใช้แนวคิดใหม่”

ซึ่งความท้าทายใหม่ ๆ ดังที่กล่าวมา สอดคล้องกับกฎแห่งผลที่ไม่ได้คาดหวัง เพราะเรายังคงคาดถึงผลกระทบบางอย่างไม่ได้ในวันนี้ แต่นักวิศวกรซอฟต์แวร์สามารถจะเตรียมพร้อมได้โดยการเริ่มกระบวนการที่สามารถปรับเปลี่ยนให้เข้ากับเทคโนโลยีที่เปลี่ยนแปลงได้ในอนาคตข้างหน้า

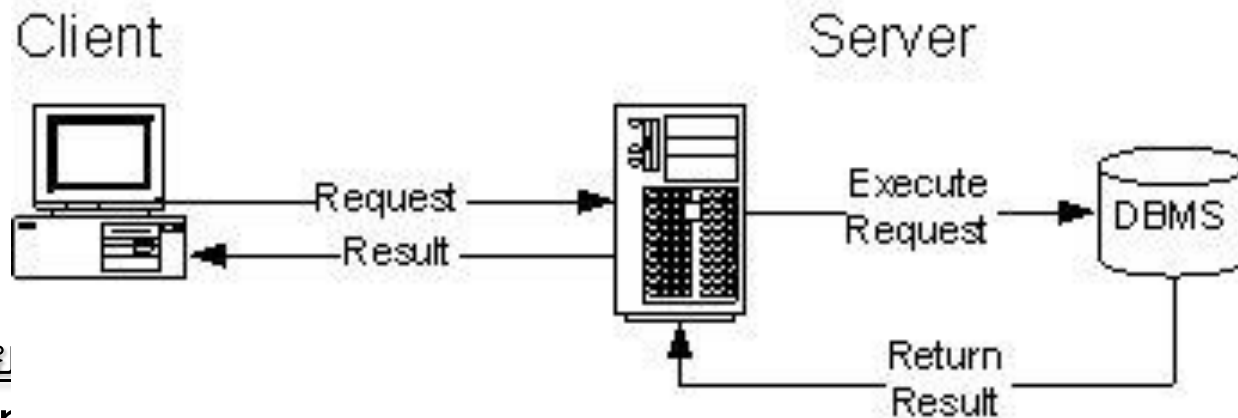




Introduction

Server - เครื่องที่ทำหน้าที่ในการให้บริการแก่ Client

Client - เครื่องที่ไปขอใช้บริการอย่างใดอย่างหนึ่งจาก Server



หมาย

Client... process

ที่อยู่บนต่างเครื่องกัน มีขอบข่ายการทำงานที่แบ่งแยกกันอย่างชัดเจน คือ Server process ที่ใช้เป็นตัวให้บริการอยู่ในเครื่องหนึ่ง ในขณะที่ Client Process เป็นตัวใช้บริการจะอยู่อีกเครื่องหนึ่ง





Introduction

3 Information Age of client-Server

ยุคที่ 1 : ยุคของ Main Frame กับ Terminal

ลักษณะสถาปัตยกรรมของระบบ

- ถูกเรียกว่า Host-Based Architecture หรือ Centralized System ระบบแบบรวมศูนย์
- ทุกตัวต้อง Connection Host วึ่งจาก Terminal ผ่าน Front- End Processor ที่ทำหน้าที่ Timesharing ให้ User รู้สึกว่าไม่สะดวกในการทำงาน

ลักษณะสถาปัตยกรรมของบุคคล

- แผนก MIS เป็นศูนย์กลาง
- ผู้เชี่ยวชาญในการพัฒนา Application, การวิเคราะห์ระบบ, การ Program ระบบ และสนับสนุนด้านเครือข่าย





Introduction

ยุคที่ 1 : ยุคของ Main Frame กับ Terminal (ต่อ)

ข้อดี :

- เมื่อเป็น Centralized control จะทำให้ง่ายต่อการ Code, Debug โปรแกรม

ข้อเสีย :

- เกิดคอขวดที่ Server เนื่องจากงานทั้งหมดถูกทำที่ Server ซึ่งจะส่งผลให้มี Performance จำกัด
- Upgrade ยุ่งยาก และมีราคาแพง (Mainframe)

หมายเหตุ :

สถาปัตยกรรมแบบนี้จะเรียกเครื่องแม่ข่ายว่า Host ซึ่งมีความหมายเดียวกันกับคำว่า Server





Introduction

ยุคที่ 2 : ยุคเริ่มต้นของ Personal Computer (PC) มีการผสมผสานกับ M/F อยู่

ลักษณะสถาปัตยกรรมของระบบ

- ถูกเรียกว่า Client-Based Architecture
- การทำงานทุกอย่างจะอยู่ที่ Client
- ส่วน Server ทำหน้าที่เป็น Data Storage แบบนี้ไม่ค่อยเป็นที่นิยม เนื่องจากคอขวดของระบบอยู่ที่เครือข่ายเพราะสถาปัตยกรรมแบบนี้ ใช้ Bandwidth ค่อนข้างสูง

ยกตัวอย่างเช่น ต้องการ Query ข้อมูลใช้ภาษา SQL ตัวข้อมูลจริงต้อง Load ผ่าน Network มาที่เครื่อง Client ทิ้งก่อนเพื่อประมวลผล ในการ update ก็เช่นกัน ต้องส่งข้อมูลทิ้งก่อน ไปที่ Server ในการใช้จริง จะมี Client หลาย เครื่องคอขวดของระบบจะเกิดที่ เครือข่าย (Network) ซึ่งแออัดไปด้วยข้อมูล



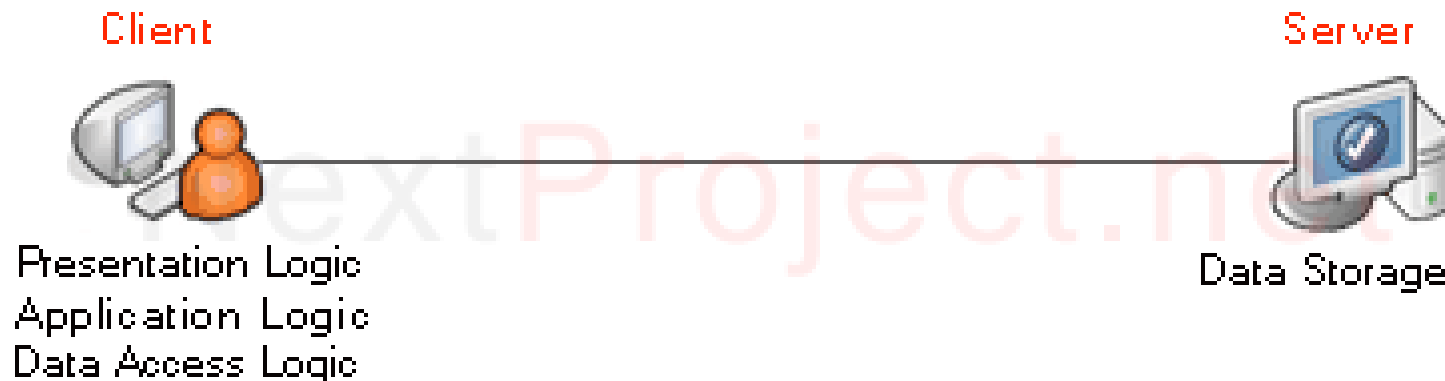


Introduction

ยุคที่ 2 : ยุคเริ่มต้นของ Personal Computer (PC) ต่อ.

ลักษณะสถาปัตยกรรมของบุคคล

- แผนก MIS เป็นศูนย์กลาง
- มีการพัฒนาผู้เชี่ยวชาญ PC ในแต่ละแผนก
- เริ่มเกิดการโต้เถียงกันว่า Information นี้เป็นของใคร ของแผนก MIS หรือของ User ในแต่ละแผนก





Introduction

ยุคที่ 3 : ยุคเริ่มต้นของสถาปัตยกรรม Client-Server

ลักษณะสถาปัตยกรรมของระบบ

- ถูกเรียกว่า Client-Server Architecture หรือ Distributed System ระบบแบบกระจาย
- PC/MF อยากได้ข้อมูลที่ข้ามกัน เป็น Enterprise Network Connect ผ่านกันเกิดลักษณะ Cross Function
- การพัฒนา Application ต้องเชื่อมกัน เช่น การเงิน ต้องใช้ข้อมูลบุคลากร หรือ งบประมาณต้องใช้ข้อมูลการเงิน เป็นต้น

ลักษณะสถาปัตยกรรมของบุคคล

- เริ่มมอบอำนาจสู่ User
- การพัฒนาโปรแกรมบนพื้นฐานความต้องการของแต่ละแผนก, Cross Function
- คนของแผนก MIS สวมบทบาทในฐานะที่ปรึกษา
- เปลี่ยนบทบาทจากแผนก MIS เป็นศูนย์กลาง





Introduction

ระบบ Client/Server เกิดขึ้นเพราะ

1. มีการเปลี่ยนแปลงยุคหลักของ ๆ คอมพิวเตอร์
 - การเข้ามาของเครื่องคอมพิวเตอร์ส่วนบุคคล (PC)
 - การเข้ามาของเทคโนโลยีด้านสื่อสาร ที่นำมาเป็นเครือข่าย PC
 - การเข้ามาของเทคโนโลยีด้านสื่อสาร WWW
2. มีการแบ่งหน้าที่ระหว่าง เครื่อง Clients จำนวนมาก กับเครื่อง Server จำนวนน้อย เพื่อ
 - ทำให้แบ่งแยกหน้าที่ของ Application ต่างประเภทกันได้ ทำให้ออกแบบได้ชัดเจน
 - ระบบมีความยืดหยุ่นมากขึ้น และการดูแลรักษาระบบง่ายขึ้น





Introduction

ระบบ Client/Server เกิดขึ้นเพราะ (ต่อ)

3. Low cost PCs made ideal clients
4. LANs เป็นมาตรฐานการสื่อสารที่ทำให้ง่ายต่อการเชื่อมต่อ Clients กับ Server
5. Hardware และ Software ที่ใช้ร่วมกัน
 - ทำให้ราคาลดลง และ ง่ายในการเชื่อมต่อกับผู้ขายอุปกรณ์และ Software หลากหลาย





ลักษณะการทำงานของ Client/Server

1. การให้บริการ (Service)

- Server เป็นผู้ให้บริการ (Server provider of service)
- Client เป็นผู้ใช้บริการ (Client consumer of service)
- การเข้ามาของเทคโนโลยีด้านสื่อสาร WAN และ WWW

2. มีการแบ่งปันทรัพยากร (Shared Resources)

- Server สามารถให้บริการหลาย ๆ Client ในเวลาเดียวกัน
- Server สามารถควบคุม Client ในการเข้าถึง Shared resources ได้

3. Asymmetrical Protocols (Protocols แบบอสมมาตร)

- ความสัมพันธ์ของ Client-Server จะเป็นแบบ M:1
- Client จะเป็นผู้ร้องขอบริการเสมอ
- Server จะรอการร้องขอจาก Client





ลักษณะการทำงานของ Client/Server

4. Location Transparency

- Server อาจจะเป็นเครื่องชนิดเดียวกันกับ Client หรือแตกต่างกันก็ได้
- สถานที่ตั้งของ Server สามารถเปลี่ยนแปลงได้
- Software Client/Server จะทำให้ Client ค้นหา Server เจอแบบ dynamically
- เครื่องสามารถเป็น Client หรือ Server หรือเป็นทั้งคู่

5. Mix and Match

- ในอุดมคติแล้ว Software Client/Server ควรจะเป็น Platform independent ดังนั้นจึงสามารถ Mix and Match Client and Server Platforms





ลักษณะการทำงานของ Client/Server

6. Message-based Exchange

- Clients and Servers loosely coupled system
- ติดต่อกันด้วยการใช้ Message-passing mechanism เพื่อ Service request and replies.

7. Encapsulation of Services

- Server เป็น Specialist
- Server รับการร้องขอ Service และตอบสนองโดยตัดสินใจว่าจะทำอย่างไรงานจึงจะเสร็จโดยไม่เกิดผลกระทบต่อ Clients ตราบที่ Message Interface ไม่มีการเปลี่ยนแปลง





ลักษณะการทำงานของ Client/Server

8. Performance Scalability

- Horizontal Scaling : การเพิ่ม Clients ส่งผลกระทบต่อประสิทธิภาพเพียงเล็กน้อย
- Vertical Scaling : การเพิ่ม Server หรือ Upgrade Server ที่เร็วขึ้นเพื่อเพิ่มประสิทธิภาพ

9. Maintainability

- เครื่อง Server จะถูกบริหารจัดการโดยส่วนกลาง ซึ่งค่าใช้จ่ายในการบำรุงรักษาถูกกว่าและปลอดภัยกว่า
- เครื่อง Client จะเหลือเพียง Personal and independent





ชนิด Sever ที่ใช้ในระบบ Client/Sever

1. File Server

- ข้อมูลจะถูกเก็บไว้ที่ Server
- Client ขอใช้บริการข้อมูลที่ Server คือ File ที่อาจเป็น Program หรือ Data
- เมื่อ Client ส่งสัญญาณ Request แล้ว Server จะ Copy ส่ง File ที่ขอให้
- Process ส่วนใหญ่เกิดขึ้นที่ Client เป็นหลัก Server มีหน้าที่แค่บริการ File
- เป็น Model ที่ง่ายและราคาถูก
- Print Server หรือ Fax Server เป็นกลุ่มที่ทำงานแบบ File Server เช่นกัน เพราะเป็นการส่งผ่านข้อมูลลักษณะ File เช่นเดียวกัน



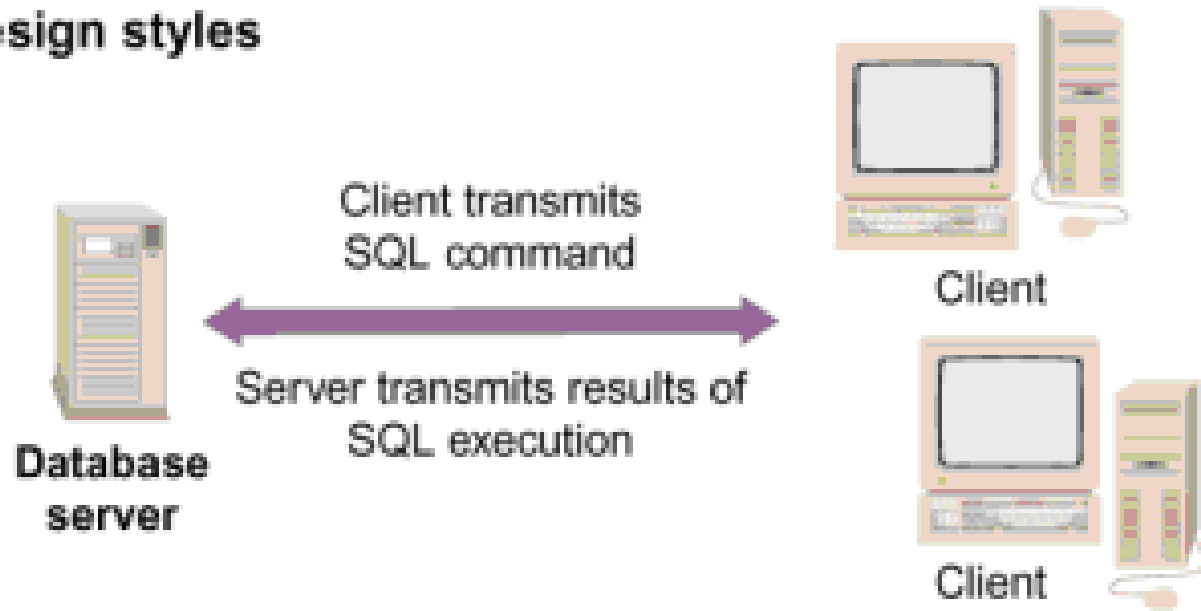


ชนิด Sever ที่ใช้ในระบบ Client/Sever

2. Database Server

- เหมาะกับ Relational Database Model
- Client จะเรียกโดยการส่ง SQL request message ไปยัง Server จากนั้น DBMS จะทำการประมวลผลคำสั่ง คั้นหรือประมวลผลข้อมูลใน Database แล้วส่งผลลัพธ์กลับไปยัง Client - มีปริมาณ Traffic น้อยมากใน Network เพราะเป็นการส่งผ่านแค่คำสั่งกับผลลัพธ์ถึงกันเท่านั้น

Client/server design styles





ชนิด Sever ที่ใช้ในระบบ Client/Sever

2. Database Server (ต่อ)

- ภาระทั้งหมดตกอยู่ที่ Server เนื่องจากต้องรองรับงานจากทุก client บางช่วงเวลาที่มีการส่ง command จาก client หนักๆจะทำให้ Server ไม่สามารถประมวลผลได้ทัน
- ในระดับ Centralized Host SQL Command ก็จะถูกสร้างขึ้นที่ Server ซึ่งจะต่างจากใน PC Client ที่จะส่งมาเป็น Command เนื่องจาก Client เป็น Terminal ที่ไม่มี CPU และ Memory ทุกตัวอักษรที่กดจะถูกส่งไปที่ Host ทั้งหมด และจะเริ่มประมวลผลทันทีที่ Submit คำสั่ง
- การตอบสนองจะเป็นไปในลักษณะ Interactive แต่ไม่สามารถคาดหมายเวลาในการตอบสนองได้ เนื่องจากขึ้นอยู่กับช่วงเวลานั้น



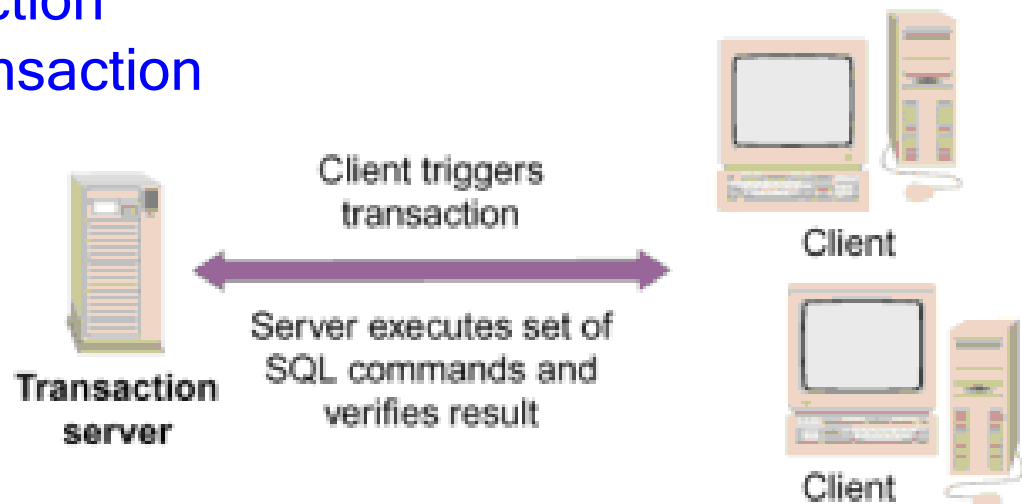


ชนิด Sever ที่ใช้ในระบบ Client/Sever

3. Transaction Server

- สร้างขึ้นมาเพื่อแก้ปัญหา Crash ของ Server ในช่วงเวลา Busy หนัก ๆ
- แบ่งกลุ่ม Transaction Server เป็นสองกลุ่มคือ
 - (1) TP Monitor (TP-Heavy) เหมาะสำหรับงานที่มีปริมาณ Transaction มากๆ
 - (2) Store Procedure เหมาะสำหรับ งานที่มีปริมาณ Transaction ไม่มาก

- ลักษณะงานของ Transaction Server คือ รองรับ Transaction ทั้งหมดจาก User แล้วส่งไป Process ที่ Server ในอัตราสูงสุด ไม่เกินปริมาณที่ Server สามารถจะรับได้ในแต่ละช่วงเวลา





ชนิด Sever ที่ใช้ในระบบ Client/Sever

4. Groupware Server

- ใช้งานกับ Data ชนิดที่ไม่มีโครงสร้าง (Un-Structure) หรือกึ่งโครงสร้าง (Semi-Structure)
- รองรับงานจำพวก Email , Scheduling , Conferencing , Workflow หรือบางครั้งก็เป็น Multimedia หรืองานที่มีลำดับ การทำงานไม่แน่นอน

5. Object Application Server

- เป็นเทคโนโลยี Client/Server ในยุคต่อไป
- 1 Application เกิดจากการนำ object หลายๆตัวที่มี relation ต่อกันและใช้วิธี Object Oriented Programming Language ในการเขียน





ชนิด Sever ที่ใช้ในระบบ Client/Sever

5. Object Application Server (ต่อ)

- 1 Object คือ Unit อีกระยะที่สามารถ process application เองได้ - ถ้าต้องการสร้างโปรแกรมใหม่ก็เพียงแต่นำ Object ที่มีอยู่แล้วมาสร้างใหม่ได้
- เป็นตัวรวบรวม object ทั้งหมด และมี ORB (Object Request Broker) จะเป็นตัวเชื่อมต่อระหว่าง Client กับ Server โดยเมื่อ Client จะติดต่อไปหา Object ก็จะต้องติดต่อไปหา ORB เพื่อบอกว่าต้องการ Object อะไร ORB ก็จะจัดสรรให้ตามต้องการ
- OMG : Object Management Group เป็นกลุ่มงานที่กำหนด frame work ของการสร้าง object เพื่อกำหนดว่าบริษัทผู้ผลิต object จะออกมาอย่างไรให้เข้ากันได้ 100%

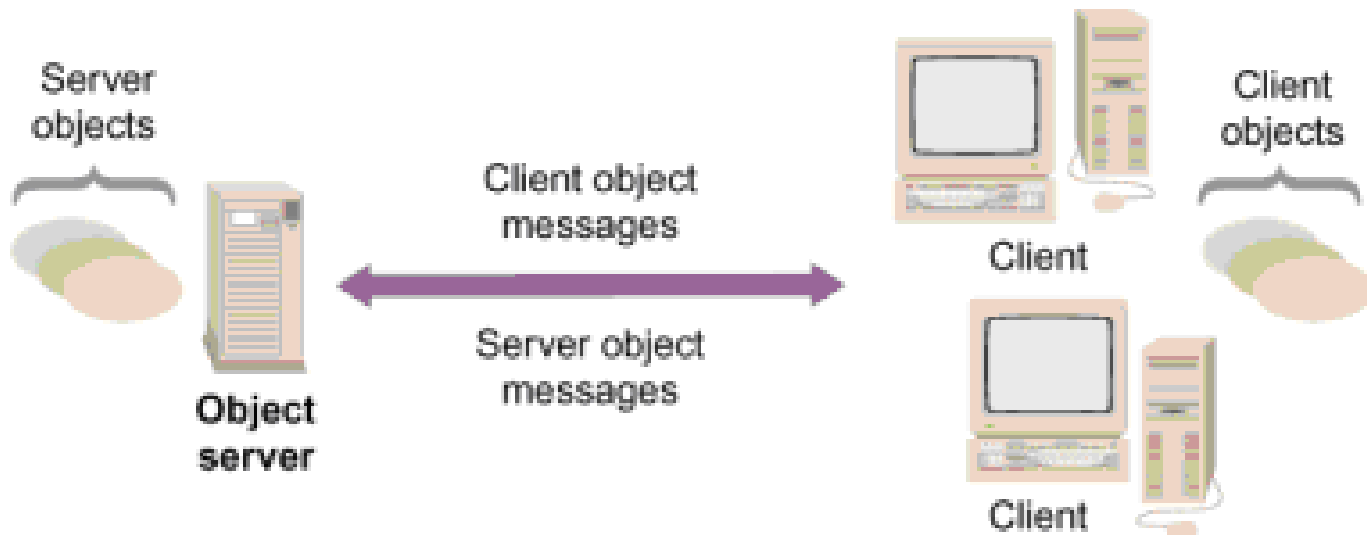




ชนิด Sever ที่ใช้ในระบบ Client/Sever

5. Object Application Server (ต่อ)

- COBRA : Common Object Broker Adapter เป็นแกนกลางที่ทำให้ object ที่มาจากบริษัทที่ต่างกันแลกเปลี่ยนหรือใช้ร่วมกันได้
- Component : คือชิ้นส่วนที่สามารถนำไปใช้ได้หลายๆอย่าง แต่มันจะมีหน้าที่ของมันเอง โดยเราจะนำเอา Component เหล่านี้มาสร้างเป็น Application

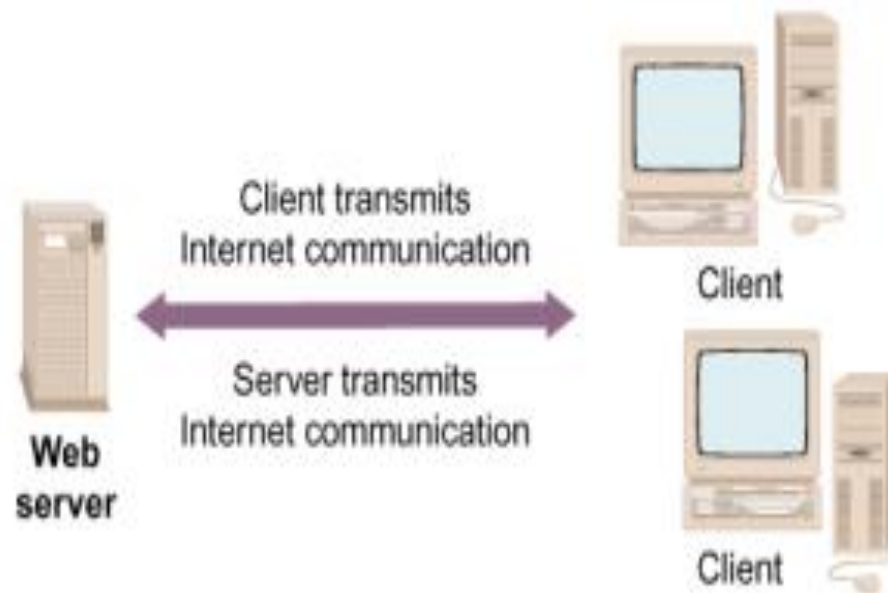




ชนิด Sever ที่ใช้ในระบบ Client/Sever

6. Web Application Server

- www เป็น Application Level Protocol ซึ่งเดิมเป็นระบบ Web-Client เท่านั้น กล่าวคือ Client จะมี browser ที่จะ request ไปยัง host ที่ต้องการ แล้ว host ก็จะทำการส่ง html กลับมายัง client เพื่อ process แล้วแสดงผลออกมา
- ปัจจุบันพัฒนาเป็น Front-End Application Client





Client-Sever Architecture

ข้อดี

- ▶ มีระบบความปลอดภัยสูง
- ▶ มีประสิทธิภาพสูง
- ▶ สำรองข้อมูลได้ง่าย
- ▶ มีความน่าเชื่อถือสูง เนื่องจากรองรับระบบ Fault-Tolerance เช่น รองรับการจัดตั้งฮาร์ดดิสก์หลายตัวด้วยเทคโนโลยี Raid
- ▶ มีอุปกรณ์และยูทิลิตี้โปรแกรม รวมถึงเครื่องมือต่างๆสนับสนุนการใช้งานค่อนข้างมาก

ข้อเสีย

- ▶ จำเป็นต้องพึ่งพาผู้มีความรู้ความเชี่ยวชาญในการจัดการดูแลระบบทั้งหมด
- ▶ การลงทุนค่อนข้างสูง เมื่อเทียบกับเครือข่ายแบบ Peer-to-Peer เนื่องจากอุปกรณ์บางชิ้นเป็นอุปกรณ์เฉพาะของเครื่องเซิร์ฟเวอร์ ซึ่งส่วนใหญ่มักมีราคาแพง

