บทที่ 2
แนวคิดและทฤษฎี

ระบบรับ-ให้บริการเป็นระบบที่เครื่องรับบริการเข้าถึงข้อมูลอยู่ที่เครื่องให้บริการ โดยการที่เครื่องรับบริการจะส่งการร้องขอข้อมูลไปยังเครื่องให้บริการ เพื่อให้ทางเครื่องบริการซึ่งทำหน้าที่เป็นเครื่องให้บริการฐานข้อมูลและมีระบบบริหารฐานข้อมูล ทำการประมวลผล และจัดส่งข้อมูลผลลัพธ์กลับมาให้ ยังเครื่องรับบริการที่ร้องขอข้อมูลไป การกระทำเช่นนี้จะช่วยให้การทำงานไม่ขึ้นอยู่กับ ฮาร์ดแวร์ และระบบปฏิบัติการมากนัก และยังช่วยให้ความเร็วในการสื่อสารเพิ่มขึ้นด้วย เพราะจะใช้ข้อมูลในการสื่อสารลดลง เนื่องจากไม่จำเป็นต้องส่งข้อมูลทั้งหมด ไปให้เครื่องรับบริการทำการประมวลผล

2.1 ประเภทของเครื่องให้บริการ

เครื่องให้บริการแฟ้มข้อมูล (File Server) เป็นเครื่องให้บริการที่มีรูปแบบง่ายที่สุด โดยเครื่องให้บริการจะร้องขอแฟ้มข้อมูลร่วมไปยังเครื่องให้บริการจัดส่งไปให้ รูปแบบการบริการนี้ต้องการแบนวิธสูงมาก และทำให้การสื่อสารในระบบเครือข่ายช้าลง มักใช้ในระบบแลนซึ่งยอมให้ผู้ใช้ทรัพยากรร่วมกัน เช่นแฟ้มข้อมูล และอุปกรณ์ต่อพ่วงต่างๆ

เครื่องให้บริการฐานข้อมูล (Database Server) จะรับเมสเซจจากเครื่องรับบริการ ซึ่งส่งมาเป็นภาษาสอบถามเชิงโครงสร้าง และทำการประมวลผล เมื่อประมวลผลเสร็จแล้วก็จะส่งผลลัพธ์ที่ได้กลับไปยังเครื่องรับบริการ

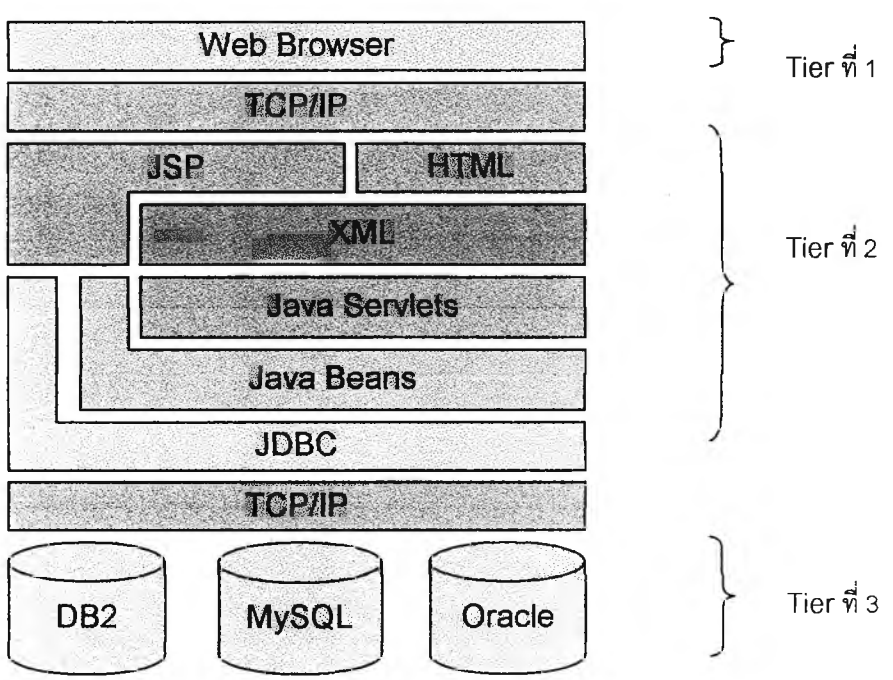
2.2 สถาปัตยกรรมของระบบ รับ-ให้บริการ

สถาปัตยกรรมแบบ 2 ไท (Two Tier Architectures) เป็นการติดตั้งระบบติดต่อกับผู้ใช้ไว้ที่เครื่องรับบริการ และติดตั้งส่วนของการให้บริการการจัดการฐานข้อมูลไว้ที่เครื่องให้บริการ ซึ่งโดยทั่วไปจะเป็นเครื่องที่มีประสิทธิภาพสูง และสามารถให้บริการเครื่องคอมพิวเตอร์รับบริการได้หลายเครื่อง การประมวลผลจะแยกกันระหว่างส่วนของการติดต่อกับผู้ใช้ที่เครื่องรับบริการ และส่วนของการจัดการฐานข้อมูลที่เครื่องให้บริการ สถาปัตยกรรมแบบนี้ส่วนมากจะนิยมใช้กับเครือข่ายที่มีเครื่องคอมพิวเตอร์จำนวนน้อย และมีการเข้าถึงระบบฐานข้อมูลพร้อมๆ กันไม่เกิน 100 ราย

สถาปัตยกรรมแบบ 3 ไท (There Tier Architectures) เป็นสถาปัตยกรรมที่มีการเพิ่มเครื่องที่ให้บริการ เพื่อทำหน้าที่เป็นตัวกลาง (Middle Tier) ในการติดต่อระหว่างระบบติดต่อกับผู้ใช้ที่เครื่องรับบริการ และระบบจัดการฐานข้อมูลที่เครื่องให้บริการ ในระบบเครือข่ายที่มีเครื่องให้บริการเป็นจำนวนมาก และมีผู้ใช้ที่ต้องการเข้าถึงฐานข้อมูลพร้อมๆกันมากกว่า 100 ราย โดยเครื่องบริการที่ทำหน้าที่เป็นตัวกลางนี้จะคอยตรวจสอบการขอใช้บริการจากเครื่องรับบริการ จากนั้นจึงจะขอบริการไปยังเครื่องให้บริการ เมื่อได้รับผลลัพธ์จากเครื่องให้บริการแล้ว จึงส่งผลลัพธ์ที่ได้นั้นกลับไปยังเครื่องรับบริการที่ร้องขอต่อไป สถาปัตยกรรมแบบนี้สามารถรองรับผู้ใช้งานได้พร้อมกันมากกว่า 1,000 ราย

2.3 รูปแบบของ สถาปัตยกรรมที่ใช้ในการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมของผู้วิจัย ได้ยึดถือรูปแบบ ในการพัฒนาโปรแกรม ตามมาตรฐานของ จาวาเทคโนโลยี โดยมี รูปแบบ ของโปรแกรมประยุกต์บนสถาปัตยกรรมแบบทรีไท ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงรูปแบบ ของโปรแกรมประยุกต์บนสถาปัตยกรรมแบบทรีไท

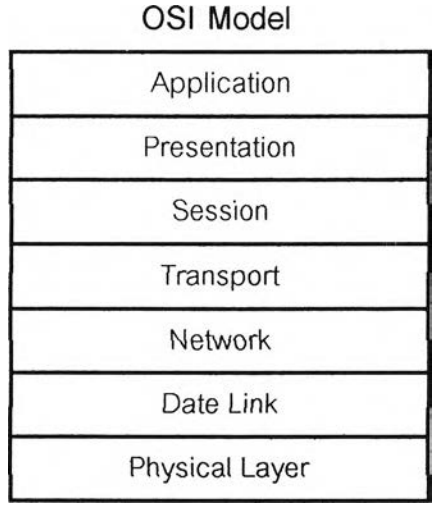
ส่วนของการทำงานในรูปแบบ ของโปรแกรมประยุกต์บนสถาปัตยกรรมแบบ 3 ไท ที่ผู้วิจัยได้นำเสนอ นั้นสามารถอธิบายลำดับในการทำงานได้ดังต่อไปนี้

หลังจากที่ เครื่องรับบริการได้ส่งความต้องการไปยังเครื่องให้บริการ โดยสามารถที่จะเลือกการติดต่อ ได้ 2 ทางคือ ติดต่อผ่านทางโปรแกรม JSP (Java Server Page) หรือติดต่อผ่านทาง HTML โดยถ้าเราเลือกการติดต่อผ่านทาง JSP เราก็ยังสามารถที่จะเลือกที่จะติดต่อไปยังเครื่องให้บริการฐานข้อมูลได้อีก 3 วิธีคือ 1. เข้าทางตรงโดยใช้ JDBC 2. เลือกติดต่อผ่านทาง จาวาบีน โดยที่ จาวา บีน จะทำหน้าที่ติดต่อกับ เครื่องให้บริการฐานข้อมูลให้อัตโนมัติ โดยผ่านทาง JDBC อีกครั้งหนึ่งหรือ 3. ติดต่อผ่านทาง XML ซึ่งจะเหมือนกับการติดต่อผ่านเข้ามาทาง HTML โดยที่ XML จะเป็นตัวที่กำหนดการทำงานในการติดต่อที่เข้ามาจากเครื่องรับบริการ โดยจะมีการเรียกการทำงานผ่านทาง จาวาเซอร์ฟเล็ต และเมื่อโปรแกรมต้องการข้อมูลจากเครื่องให้บริการฐานข้อมูลที่อยู่อีกเครื่องหนึ่ง โปรแกรมก็จะทำการติดต่อผ่านทาง จาวาบีน และเมื่อโปรแกรมทำงานเสร็จแล้ว จะให้ส่งผลลัพธ์ที่ได้ผ่านทางหน้าจอไหน หรือเมื่อไม่สามารถทำงานได้ จะให้ทำการแจ้งผู้ใช้งานผ่าทางหน้าจอไหน เป็นต้น

2.4 โพรโตคอล TCP/IP

การสื่อสารของเครือข่ายคอมพิวเตอร์มักจะเกี่ยวข้องกับเครื่องคอมพิวเตอร์หลากชนิด ทั้งพีซี เครื่องแมคอินทอช เมนเฟรม หรือ มินิ ซึ่งล้วนมีมาตราฐานการสื่อสารข้อมูลที่เป็นมาตราฐานของตนเองไม่เหมือนผู้อื่น ดังนั้น เพื่อให้เกิดความมั่นใจว่าเราจะสามารถส่งข้อมูลข้ามเครื่องได้จริง จึงได้มีการกำหนดมาตราฐานที่เครื่องทุกแบบเข้าใจเป็นสากลและสามารถปฏิบัติตามได้ จุดนี้จึงเป็นที่มาของโมเดล OSI (Open System Interconnection) ซึ่งเป็น โมเดล 7 ชั้น ที่ประกอบด้วยมาตราฐานแบบแผนที่ระบุถึงชนิด ลักษณะการทำงาน และวิธีการประยุกต์ใช้ OSI จะแบ่งความซับซ้อนของเครือข่ายออกเป็นชั้นๆ ดังนี้

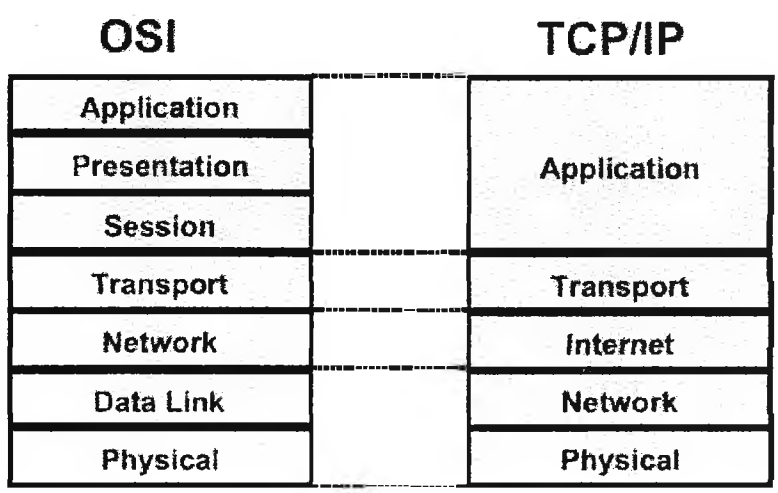
1. แอบพลิเคชั่นเลเยอร์ (Application Layer)
2. พรีเซนเตชันเลเยอร์ (Presentation Layer)
3. เซสชันเลเยอร์ (Session Layer)
4. ทรานสปอร์ตเลเยอร์ (Transport Layer)
5. เน็ตเวิร์กเลเยอร์ (Network Layer)
6. เดต้าลิงก์เลเยอร์ (Data Link Layer)
7. ฟิสิคัลเลเยอร์ (Physical Layer)



รูปที่ 2.2 แสดงลำดับชั้นของโมเดล OSI

Introduction to Transport Protocols

การให้บริการการเชื่อมต่อพื้นฐานมีอยู่ 2 ประเภท นั่นก็คือ Connection oriented และ Connectionless service (Datagram service) Connection-oriented service จะให้บริการการสร้างเส้นทางเชื่อมต่อระหว่าง Transport service (TS) user ตลอดจนบำรุงรักษา รวมทั้งยกเลิกการเชื่อมต่อดังกล่าว ซึ่งจัดได้ว่าเป็นโพรโตคอลที่ซับซ้อน แต่ก็มีประสิทธิภาพสูง และ มีการรับประกันว่าข้อมูลที่เราส่งไป จะไปถึงผู้รับได้อย่างถูกต้อง ตัวอย่างของโพรโตคอลแบบนี้ก็คือ TCP (Transmission control protocol) ส่วนการเชื่อมต่ออีกแบบหนึ่งคือ Connectionless service การเชื่อมต่อแบบนี้จะไม่มีการรับประกันถูกต้องว่าข้อมูลที่ส่งจะถึงผู้รับหรือไม่ ตัวอย่างของโพรโตคอลแบบนี้ก็คือ UDP



รูปที่ 2.3 แสดงการเทียบโมเดล TCP/IP กับโมเดล OSI

TCP Services

บริการ TCP เริ่มต้นจากการที่ทั้งฝ่ายผู้รับและผู้ส่งข้อมูลสร้างส่วนควบคุมการติดต่อระหว่างกันแบบที่เรียกว่า "ซ็อคเก็ต (socket)" โดยซ็อคเก็ตแต่ละตัวจะมีหมายเลขที่อยู่ประกอบด้วยสองส่วนคือหมายเลข IP ขอโฮสต์และหมายเลขขนาด 16 บิต ที่ใช้ภายในโฮสต์นั้นๆ เรียกว่า "พอร์ต (port)"

การเชื่อมต่อของเอ็นทีที TCP เป็นการสื่อสารสองทางแบบสมบูรณ์ (full-duplex) และเชื่อมต่อจุด-ต่อ-จุด (point-to-point) หมายความว่าการเชื่อมต่อจะมีโฮสต์อยู่เพียง 2 โฮสต์ที่ปลายสายแต่ละข้างซึ่งสามารถส่งข้อมูลส่วนทางกันได้ตลอดเวลา TCP ไม่สนับสนุนการเชื่อมต่อหลายจุด (multicasting) และแบบกระจายข่าว (broadcasting)

การส่งข้อมูลผ่าน TCP โพรโตคอลเป็นไปในลักษณะชุดหรือกระแสไบต์ (byte stream) ไม่ใช่กลุ่มของข่าวสาร (message) ขอบเขตของข่าวสารแต่ละชุดจึงไม่มีการกำหนดใช้งาน เช่น การส่งข่าวสารออกไป 4 ชุดชุดละ 512 ไบต์ผ่าน TCP โพรโตคอล ข้อมูลทั้งหมดอาจถูกส่งไปยังผู้รับในลักษณะ เป็นชุด 512 ไบต์ 4 ชุด ชุดละ 1024 ไบต์ 2 ชุด ชุดละ 2048 ไบต์ 1 ชุด หรืออย่างไรก็ได้ ทางผู้รับจึงไม่สามารถทราบได้เลยว่าข้อมูลที่ส่งมานั้นประกอบด้วยข่าวสาร (message) กี่ชุด

แฟ้มข้อมูลในระบบปฏิบัติการ ยูนิกส์ ก็ใช้วิธีการเดียวกันนี้ในการเก็บข้อมูล นั่นคือผู้ใช้ไม่อาจทราบได้เลยว่าข้อมูลในแฟ้มที่กำลังอ่านอยู่นั้นถูกอ่านขึ้นมาครั้งละบล็อก, ครั้งละไบต์, หรือทั้งแฟ้มพร้อมกัน TCP เอ็นทีที จึงไม่มีความสนใจในโครงสร้างของข้อมูลที่กำลังทำการรับ-ส่งอยู่เพียงแต่ให้ความสนใจในการรับ-ส่งให้ครบจำนวนไบต์อย่างถูกต้องเท่านั้น

เมื่อโพรเซสผู้ใช้ส่งข้อมูลมายัง TCP เอ็นทีที ข้อมูลนั้นอาจถูกส่งออกไปในทันที หรืออาจถูกเก็บรวบรวมไว้ก่อน (เพื่อรวบรวมส่งเป็นกลุ่มขนาดใหญ่พร้อมกัน) ในบางครั้งโพรเซสที่ส่งข้อมูลเข้ามาอาจต้องการให้ข้อมูลนั้นถูกส่งออกไปในทันที เช่น การติดต่อเข้ากับระบบโต้ตอบทันที (Interactive) เมื่อผู้ใช้พิมพ์คำสั่งเสร็จข้อมูลนั้นจะต้องถูกส่งออกไปในทันทีที่ผู้ใช้กดแป้นพิมพ์ "Enter/Return" ในการทำงานลักษณะนี้ผู้ใช้สามารถบังคับให้ TCP เอ็นทีที ส่งข้อมูลออกไปทันทีโดยใช้คำสั่ง "Push"

คุณลักษณะอีกประการหนึ่งของบริการแบบ TCP คือ "ข้อมูลด่วน (urgent data)" เมื่อผู้ใช้ระบบโต้ตอบทันทีกดแป้นพิมพ์ "Del" หรือ "Ctrl-C" เพื่อหยุดการทำงานของโปรแกรมในเครื่องที่กำลังติดต่อด้วยนั้น ข้อมูลจากแป้นพิมพ์นี้จะถูกส่งไปยังเครื่องผู้รับผ่าน TCP พร้อมกับสัญญาณข้อมูลด่วนซึ่งจะส่งผลให้ TCP เอ็นทีที หยุดการรวบรวมข้อมูลและจัดการส่งข้อมูลที่เก็บไว้ออกไปในทันที

ข้อมูลที่ส่งไปพร้อมกับสัญญาณข้อมูลด่วนจะถูกส่งไปยังผู้รับ TCP เอ็นทิตีไม่มีหน้าที่ในการจัดการข้อมูลนี้ จึงเป็นหน้าที่ของโปรแกรมทางผู้รับที่จะต้องแปลความหมายของข้อมูลนี้เอง กระบวนการนี้ แม้ว่าจะไม่ได้อำนวยความสะดวกมากนัก แต่ก็ช่วยให้โปรแกรมของผู้ใช้สามารถส่งข่าวสารด่วนพิเศษไป-มาระหว่างกันได้

UDP (User Datagram Protocol)

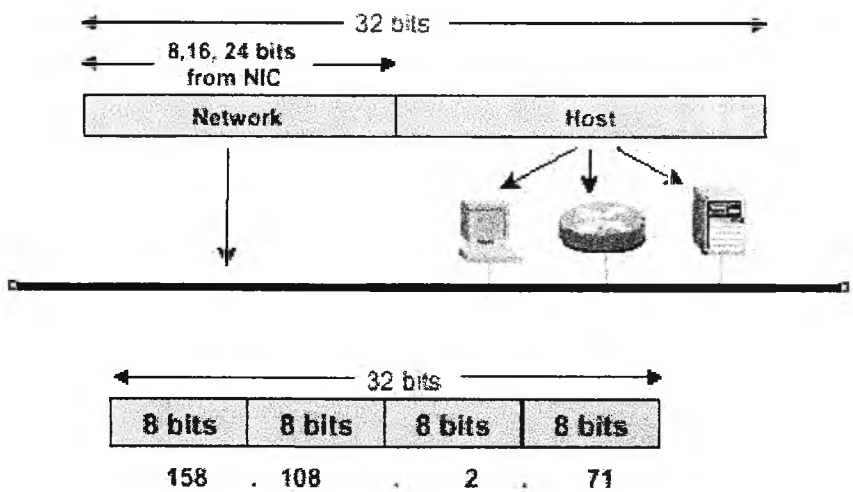
คือ อินเทอร์เน็ตโพรโตคอล ที่สามารถให้การสนับสนุนการเชื่อมต่อแบบไม่ต่อเนื่อง (Connectionless) ซึ่งเป็นการใส่ข้อมูลเข้าไปในแพ็กเก็ต IP แล้วส่งออกไปโดยไม่มีการสร้างการเชื่อมต่อจากผู้ส่งมายังผู้รับโดยตรง แต่อาศัยการเชื่อมต่อเป็นช่วงสั้นๆ ในลักษณะการฝากส่งจากผู้ส่งจนกระทั่งไปถึงผู้รับ เป็นการสื่อสารแบบผู้ให้และผู้รับบริการ (Client/Server System) ซึ่งโดยปกติจะมีการขอใช้บริการจากผู้ใช้ส่งไปหนึ่งแพ็กเก็ต และมีผลลัพธ์จากบริการนั้นๆ ส่งกลับมาหนึ่งแพ็กเก็ต จึงไม่เหมาะสมที่จะสร้างการเชื่อมต่อขึ้นมาใช้งาน โพรโตคอล UDP ได้รับการอธิบายมาตรฐานไว้ใน RFC 768

ตัวอย่างของงานที่ไม่เหมาะสมกับการใช้ TCP ซึ่งทำให้ UDP เป็นทางเลือกที่ดีกว่า เช่น

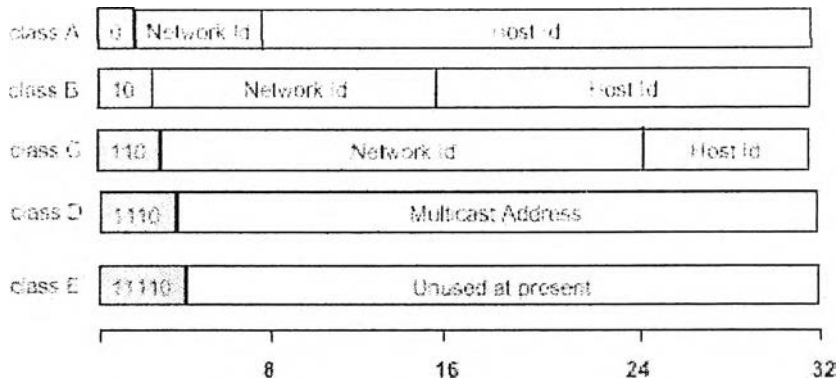
- Inward data collection
  งานที่เกี่ยวข้องกับการเก็บตัวอย่างข้อมูลภายในคาบเวลา เช่น Sensors, รายงานผลการทดสอบตัวเองจากอุปกรณ์รักษาความปลอดภัย หรือส่วนประกอบต่างๆของเครือข่าย ซึ่งจะต้องมีการรายงานผลแบบ real-time การสูญหายของข้อมูลจะต้องไม่ทำให้เกิดความล่าช้าขึ้น เนื่องจากข้อมูลชุดต่อไปจะต้องถูกส่งมาในระยะเวลาอันสั้น
- Outward data dissemination
  เกี่ยวข้องกับการกระจายข่าว (Broadcast) ไปยังเครือข่ายผู้ใช้งาน, การประกาศ node ใหม่ หรือ การเปลี่ยนแปลงที่อยู่ของบริการ และ การแจกจ่ายค่าของ real-time clock
- Real-time applications
  เช่นการส่งผ่านข้อมูลภาพและเสียง ซึ่งเป็นข้อมูลในลักษณะ real-time ดังนั้นจึงไม่มีความจำเป็นจะต้องให้บริการ Retransmission

Application		FTP TELNET SMTP TFTP NFS NTP SNMP NNTP DNS BOOTP DHCP HTTP X-windows
Transport		TCP / UDP
Internet		IP
Network		Network Driver Software
Physical		

รูปที่ 2.4 แสดงการเทียบโมเดล OSI กับโมเดลของโปรแกรมที่ใช้งาน



รูปที่ 2.5 แสดงการกำหนดหมายเลข IP



รูปที่ 2.6 แสดงการกำหนดหมายเลข IP ของแต่ละ Class

ตารางที่ 2.1 แสดงการจัดแบ่ง IP Address ของแต่ละ Class

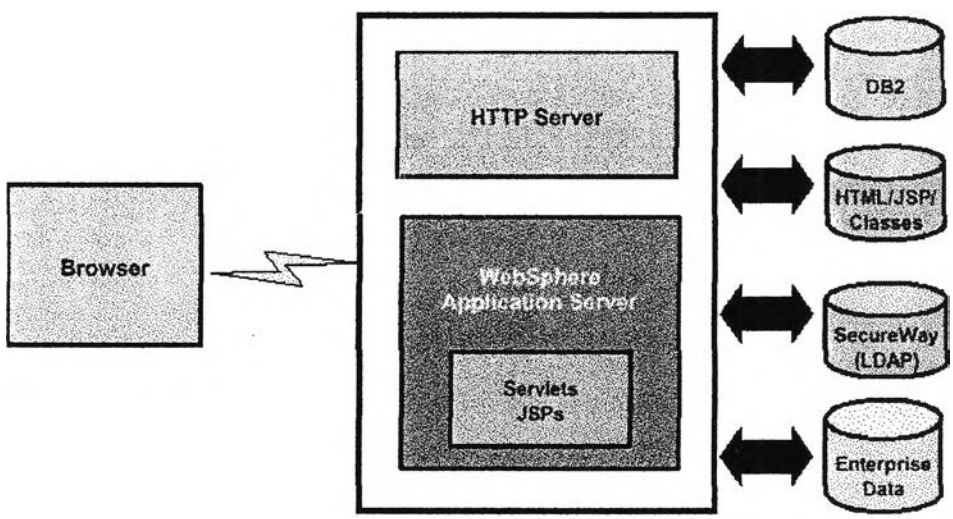
Class	Initial bits	#bit net	#bit host	range	
A	0	7	24	0.0.0.0	127.255.255.255
B	10	14	16	128.0.0.0	191.255.255.255
C	110	21	8	192.0.0.0	223.255.255.255
D	1110	28	--	224.0.0.0	239.255.255.255
E	11110	27	--	240.0.0.0	247.255.255.255

ตารางที่ 2.2 แสดงจำนวน Address ที่สามารถใช้งานได้ในแต่ละ Class

Class	Address spaces	useable
A	2^24=16677216	166777214
B	2^16=65536	65534
C	2^8=256	254

2.5 โครงสร้างระบบงานบนอินเทอร์เน็ต (Web Application Model)

เป็นโครงสร้างที่ใช้สถาปัตยกรรมแบบ 3 ไท (There Tier Architectures) ในการทำงาน แต่จะแตกต่างกันตรงที่เครื่องรับบริการจะไม่มีระบบงานติดตั้งไว้ จะมีแต่เพียงโปรแกรมที่ใช้ในการอ่านข้อมูล (Browser Program) ที่ทำหน้าที่คอยนำเสนอข้อมูลที่รับมาจากเครื่องให้บริการตัวกลางที่ทำหน้าที่เป็น ผู้ให้บริการข้อมูล



รูปที่ 2.7 สถาปัตยกรรมแบบทรีไท (Thee Tier Architectures)

2.6 ภาษาจาวา (Java Language)

กำเนิดภาษา จาวา มีสาเหตุเริ่มต้นจากความยุ่งยากในการพัฒนาโปรแกรม ใช้งาน (Application) สำหรับระบบอินเทอร์เน็ต ทั้งนี้เพราะระบบอินเทอร์เน็ต เป็นระบบเปิดที่สามารถใช้งานจากเครื่องคอมพิวเตอร์ประเภทใดๆ ก็ได้ ไม่ว่าจะเป็นเครื่องแบบ พีซี, แมคอินทอช, ซัน เครื่องมินิคอมพิวเตอร์ไปจนถึงระดับซูเปอร์คอมพิวเตอร์ จึงนับว่าเป็นข้อดีคือ ทำให้เราสามารถเข้าสู่ระบบอินเทอร์เน็ตได้ ไม่ว่าเราจะใช้เครื่องประเภทใดอยู่ก็ตาม

แต่ปัญหาของข้อดีดังกล่าวก็คือ ยังไม่มีโปรแกรมใช้งานอินเทอร์เน็ตโปรแกรมใดเลย ที่สามารถนำมาใช้งานได้กับเครื่องหลายประเภทเหล่านี้ ได้เหมือนๆกัน โปรแกรมที่พัฒนาขึ้นมาใช้กับเครื่องพีซีไม่สามารถนำมาใช้กับเครื่องเมคอินทอชหรือเครื่องซันได้ หรืออาจใช้งานได้แต่ก็ไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพ ดังนั้นในปี พ.ศ.2534 บริษัท ซัน ไมโครซิสเต็มส์ (Sun

Microsystems, Inc.) จึงได้พัฒนาภาษาคอมพิวเตอร์ใหม่ที่มีประสิทธิภาพในการทำงานชนิดไม่ยึดติดกับแพลตฟอร์มขึ้นมา

จาวา นับเป็นภาษาคอมพิวเตอร์ภาษาแรก ที่ถูกออกแบบให้มุคุณสมบัติพิเศษด้านเทคนิค สามารถสร้างโปรแกรมใช้งานจากเครื่องคอมพิวเตอร์ประเภทหนึ่ง แล้วนำไปใช้ได้กับเครื่องอีกประเภทหนึ่งที่อยู่ภายในเครือข่ายเดียวกัน

จาวา เป็นภาษาแบบอินเตอร์แอคทีฟ ที่มีข้อแตกต่างจากภาษาอื่นคือ จาวา จะทำการคอมไพล์(Compile) คำสั่งจาก ซอร์ซโค้ด(Source code) ให้กลายเป็นภาษารหัสเครื่อง ที่เรียกว่า ไบต์โค้ด(Byte code) มีคุณลักษณะเด่นคือ มีขนาดเล็ก สามารถนำไปประยุกต์ใช้งานได้สะดวกและรวดเร็ว นอกจากนี้ ภาษา จาวา ยังถูกจัดให้เป็นภาษาคอมพิวเตอร์ระดับสูงเช่นเดียวกับภาษา Fortran, Cobol, C, Pascal หรือ Basic เป็นภาษาที่มีเสถียรภาพการทำงานสูง ใช้โครงสร้างการเขียนโปรแกรมเป็นแบบ โปรแกรมเชิงวัตถุ(Object Oriented Programming หรือ OOP)

ภาษาจาวาสามารถทำงานได้หลายลักษณะ และการเรียกชื่อก็แตกต่างกัน เช่น เมื่อทำงานที่เครื่องรับบริการเราจะเรียกมันว่า จาวาแอบเพล็ต (Java Applet) แต่ถ้านำมาทำงานอยู่ที่เครื่องให้บริการเราจะเรียกว่า จาวาเซอร์ฟเล็ต (Java Servlet) โปรแกรมที่เขียนขึ้นมาด้วยภาษาจาวา ที่มีลักษณะของส่วนประกอบซอฟต์แวร์ (Software Component) เพื่อที่จะสามารถนำมาประกอบให้เข้ากันกับโปรแกรมที่เราเขียนเราจะเรียกมันว่า จาวาบีน (Java Bean)
ซึ่งเราสามารถจำแนกการทำงานของภาษา จาวา ตามลักษณะการใช้งานได้ ดังนี้

1.) จาวาสคริปต์ มีลักษณะเป็น อินเตอร์พรีเตอร์(interpleter) สามารถเขียนให้ฝังตัวอยู่ในเว็บเพจได้เลย โดยไม่ต้องคอมไพล์เป็น เอ็กซ์คิวเทเบิล ไฟล์(executable file)

2.) จาวา มีลักษณะเป็น คอมไพเลอร์(compiler) คำสั่งในภาษา จาวา ต้องเขียนเก็บไว้ในเท็กไฟล์ ที่มีส่วนขยายเป็น .java แล้วจึงคอมไพล์เป็น ไบต์โค้ด เพื่อนำมาใช้งานต่อไป ซึ่งการใช้งานภาษา จาวา ที่เป็น ไบต์โค้ด ก็สามารถจำแนกการทำงานได้เป็น 2 ลักษณะคือ

2.1) จาวาแอ็ปเพล็ต คือการเตรียมโปรแกรมไว้ที่เครื่องเซิร์ฟเวอร์ เมื่อมีการเรียกใช้งานจาก เว็บเบราเซอร์ เครื่องเซิร์ฟเวอร์จะทำการส่งข้อมูลโปรแกรมดังกล่าวกลับ เพื่อให้เว็บเบราเซอร์สั่งให้ทำงานต่อไป

2.2) จาวาเซอร์ฟเล็ต คือการเตรียมโปรแกรมไว้ที่เครื่องเซิร์ฟเวอร์ เมื่อมีการเรียกใช้งานจาก เว็บเบราเซอร์ เครื่องเซิร์ฟเวอร์จะทำงานโดยทันทีและจะส่งผลที่ได้กลับมาแสดงผลยัง เว็บเบราเซอร์

2.3) จาวาบีน คือโปรแกรมจาวาที่ทำหน้าที่เฉพาะอย่าง เมื่อต้องการใช้งานก็จะต้องนำมา รวมกับโปรแกรมหลัก เพื่อทำงานร่วมกัน

ในส่วนของการพัฒนาระบบงาน เราจะใช้ Java Servlet ในการทำงานในเครื่องที่ให้บริการที่เป็นตัวกลาง(Tier ที่ 2) เนื่องจากเทคโนโลยีของภาษาจาวานั้น เป็นที่ยอมรับกันทั่วไปว่ามีความยืดหยุ่นในการใช้งานสูง ซึ่งสามารถสรุปข้อดีที่เด่นชัด ได้คร่าวๆดังนี้

1) สามารถทำงานได้บนระบบปฏิบัติการหลายๆ ชนิดเช่น วินโดวส์, ยูนิกส์ และหรือ Mac

2) โปรแกรมที่เขียนด้วยภาษา จาวา ไม่จำเป็นต้องนำไปเขียนหรือคอมไพลใหม่ เมื่อมีการเปลี่ยนระบบปฏิบัติการ หรือมีการขยายระบบจากระบบเดิมสู่ระบบใหญ่กว่า

3) เมื่อมีการออกแบบในส่วนของโปรแกรมที่ดีแล้ว ยังสามารถนำ ส่วนที่ได้เขียนแล้วนำกลับมาใช้ได้ใหม่อีก ซึ่งเป็นการลดเวลาในการพัฒนาระบบงานใหม่ได้มาก เนื่องจากภาษา จาวา เป็นภาษาที่สนับสนุนกับ เทคโนโลยีเชิงวัตถุ (Object Oriented) ที่จะสามารถพัฒนา โปรแกรมเป็นส่วนๆ ก่อน แล้วนำมาประกอบกันในภายหลังได้

2.7 การใช้งานโปรแกรม WAS (WebSphere Application Server)

การควบคุมและการใช้งาน เราจะใช้โปรแกรม Administrative Console ที่มาพร้อมกันกับ WAS ซึ่งจะมี 3 ส่วนคือ Navigation, Content และ Message ซึ่งแต่ละส่วนก็จะทำหน้าที่แตกต่างกันคือ

Message: เป็นส่วนที่แสดงข้อความการทำงานที่สำคัญของระบบ

Content: เป็นส่วนที่แสดงข้อมูลซึ่งขึ้นอยู่กับการเลือกการทำงานในส่วน Navigation

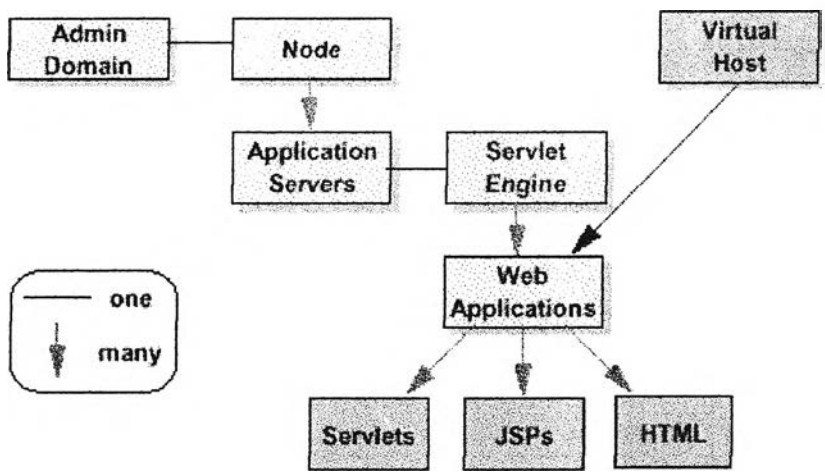
Navigation: ประกอบด้วย 3 ส่วนย่อยคือ

- Task ใช้ในการสร้าง เว็บแอบพลิเคชั่น
- Type ใช้ในการกำหนดค่าเริ่มต้นและส่วนประกอบต่างๆ
- Topology เป็นส่วนที่ใช้งานหลัก เพราะเป็นส่วนที่ใช้จัดการส่วนประกอบต่างๆ ของ เว็บแอบพลิเคชั่น

สถาปัตยกรรมของโปรแกรม WAS

WAS Topology

Topology ใน WebSphere มีโครงสร้าง ดังรูป



รูปที่ 2.8 โครงสร้างใน WAS

Node: เป็นตัวแทนของเครื่องคอมพิวเตอร์ หลังจากติดตั้ง WAS จะทำการสร้าง Node ขึ้นมาให้ โดย Node ที่สร้างจะใช้ชื่อเดียวกับชื่อเครื่อง

Appliaction Servers: ใช้ขยายความสามารถของ Web Server เพื่อที่จะให้บริการ การทำงานของ servlet, enterprise beans และ เว็บแอบพลิเคชั่น โดยที่ Application Server จะมีส่วนประกอบหลักที่สำคัญ 2 ส่วนคือ Java vertual machine และ Servlet engine

Servlet Engine: เป็นโปรแกรมที่ทำงานอยู่ภายใต้ Application Server คอยให้บริการการทำงานของ Servlet, JavaServer Page (JSP) และบริการอื่นๆ ที่ทำงานในลักษณะ Server-side includes โดยที่ WAS จะยอมให้มี 1 Servlet engine ต่อ 1 Application Server

Web Application: เป็นตัวแทนกลุ่มของ servlets, JSP และทรัพยากรที่เกี่ยวข้อง การจัดการในแต่ละส่วนสามารถสั่งให้ เริ่ม หรือ หยุดทำงาน ได้ง่าย ภายใต้ Servlet engine เราสามารถแบ่ง Web Application ออกมาได้หลายส่วน เพื่อที่จะจัดการเอกสารที่แตกต่างกัน ซึ่งโดยปรกติหลังจากการติดตั้ง WAS แล้ว Web Application จะถูกสร้างมา 3 ส่วนด้วยกันคือ

- *default_app* ในส่วนนี้เราสามารถจะใช้ในการทดสอบการทำงานของ Servlets ได้ ซึ่งส่วนนี้ได้ถูกออกแบบมาสำหรับการใช้งานในส่วนของ WAS ในรุ่นก่อนหน้านี้
- *admin* ส่วนนี้จะถูกใช้โดย WAS เราไม่ควรเข้าไปแก้ไข
- *examples* ส่วนนี้จะเก็บตัวอย่างของ Servlets เอาไว้ให้เราได้เห็นแนวคิดในการทำงานของระบบ เราสามารถทดสอบโดยการพิมพ์ http://localhost/webapp/examples/

Virtual host: เป็นกลไกที่จะยอมให้ เครื่อง 1 เครื่อง มีสภาพแวดล้อมเหมือนมีหลายเครื่อง โดยที่การใช้ ทรัพยากรนั้นจะแยกจากกัน แต่อยู่ในเครื่องเดียวกัน หลังจากติดตั้ง WAS จะสร้าง Virtual host เริ่มต้นที่ชื่อ *default_host* ให้