

Week05_Lab05: Arduino Shield and Data Communication

1. เนื้อหา

1.1 Shield

- มีหลากหลายชิ้นกับการใช้งาน
- ตัวอย่างผู้จำหน่ายในประเทศไทย
 - <http://www.adaline.co.th/category/38/board/arduino-shield>
 - <https://www.arduitronics.com/category/10/arduino-shields>
 - <https://www.arduinoall.com/category/4/sensors-modules-shield>
 - <https://www.thaielecs.com/category/13/sensors-modules-shield>
 - <https://www.arduino108.com/category/3/arduino-shields>
 - <http://www.arduino2robot.com/>
 - <https://www.9arduino.com/>

1.2 Add Lib

- ไบบรารี (Libraries) คือ คือชุดของโค้ดโปรแกรมที่มีคนอื่นเขียนไว้แล้ว เช่น คำสั่งสำหรับการใช้งานเซ็นเซอร์ต่างๆ, การใช้งานหน้าจอ LCD, Wifi Shield, GPS โมดูล และส่วนขยายต่างๆ โดยที่เราไม่ต้องมาเขียนขึ้นใหม่ทำให้สะดวกในการใช้งาน
- สามารถเพิ่ม Library ให้ Arduino IDE ได้ 3 วิธี

1.3 Data Communication

- OSI – Open System Interconnection
- อ่าน <https://www.jobpub.com/articles/showarticle.asp?id=3474>
- อ่าน <http://netprime-system.com/osi-model-7-layers/>
- อ่าน <http://datascienceth.com/manufacturingdatacsc/>
- อ่าน <http://www.theeleader.com/turning-big-data-data-analytics/>

1/3: ระบบบัสสำหรับงานอุตสาหกรรม

เนื่องจากการขยายตัวของงานอัตโนมัติชันมากในช่วง 10 ปีที่ผ่านมา ทำให้เทคโนโลยีการสื่อสารข้อมูลทางด้านอุตสาหกรรม (Industrial communication) มีความสำคัญมากขึ้น โดยจุดประสงค์เพื่อจัดการการสื่อสารข้อมูลที่มากและซับซ้อนของแต่ละลำดับชั้น ให้มีประสิทธิภาพมากที่สุด ซึ่งจากแนวโน้มการพัฒนาเทคโนโลยีที่ได้กล่าวมาแล้วนั้น คือ ได้มีการนำเทคโนโลยีระบบบัส (Bus system) มาใช้เพื่อให้บรรลุจุดประสงค์ที่ต้องการ

โดยทั่วไประบบบัสสำหรับงานอุตสาหกรรม (Industrial bus system) มีมากมายหลายชนิด ซึ่งในที่นี้จะขอจำแนกตามลำดับชั้นการติดต่อสื่อสารข้อมูลเป็น 3 ระดับ ดังนี้

1. ระดับ Factory level ซึ่งจะครอบคลุมระดับอัตโนมัติชันระดับ Factory manage และ coordinating
 2. ระดับ Cell level ซึ่งจะครอบคลุมระดับอัตโนมัติชันระดับ System และ Control
 3. ระดับ Field level ซึ่งจะครอบคลุมระดับอัตโนมัติชัน ระดับ Sensor actuator
- ซึ่งเราสามารถสรุปเป็นตารางได้ดังนี้

ลำดับชั้นของ INDUSTRIAL COMMUNICATION			ชนิดของ INDUSTRIAL BUS ในท้องตลาด
Factory level			EtherNet TCP/IP
Control level			ARCNET, ControlNet, INTERBUS, PROFIBUS-FMS
Field level	Process Bus Network (Analog) (Up to 1000 Bytes)		FOUNDATION field bus, HART, INTERBUS, LON, PROFIBUS-FMS, PROFIBUS-PA
	Device Bus Network (Discrete)	Byte-wide data (8-256 Bytes)	BITSBUS, CAN, CANopen, DeviceNet, FOUNDATION field bus, INTERBUS-S, PROFIBUS-DP, Smart Distributed System(SDS), Modbus RTU/ASII
		Bit-wide data (น้อยกว่า 8 bits)	AS Interface, INTERBUS LOOP, Seriplex

ลำดับชั้นของการควบคุมระบบอัตโนมัติชัน

คำว่า “Automation hierarchy” นั้นกล่าวถึงระดับของการควบคุมงานอัตโนมัติชัน พื้นที่ที่ใช้ความสามารถในการทำงานและทิศทางการไหลของข้อมูล ทั้งนี้เพื่อประโยชน์ในการจัดการข้อมูลที่ซับซ้อนของระบบอัตโนมัติชัน ซึ่ง Automation hierarchy จะถูกแบ่งเป็น 5 ระดับชั้นด้วยกันดังนี้

ระดับที่ 1 : Factory management level เป็นระดับที่สูงสุดโดยมีความสำคัญในการตัดสินใจเกี่ยวกับการวางแผนและความคุมการผลิต (Planning and Production Control : PPC) ซึ่งจำเป็นที่จะต้องนำข้อมูลจากทุกฝ่าย เช่น ฝ่ายขาย ฝ่ายบัญชี ฝ่ายต้นทุน ฝ่ายวัสดุ เป็นต้น มาทำการประมวลผลเพื่อการตัดสินใจ

ระดับที่ 2: Coordinating level เป็นระดับที่มีหน้าที่รับคำสั่งมาจากระดับที่ 1 จากนั้นก็จะทำการแจกจ่ายงานไปยังหน่วยการผลิต (work cell) เช่น การสั่งงานไปที่หน่วยประกอบ (Assembly cell) หน่วยสต็อก (Store cell) หน่วยการขึ้นรูป (Machine tool cell) จากนั้นก็จะมีการรายงานผลไปยังระดับที่ 1 เพื่อใช้ในการประมวลผลต่อไป

ระดับที่ 3: System level เป็นระดับหน่วยการผลิต (Cell level) ซึ่งจะทำหน้าที่ดูแลหน่วยการผลิตนั้น ๆ ในทุก ๆ เรื่องอย่างเช่น การกำหนดขั้นตอนการผลิตการซ่อมบำรุง (Maintenance) การวิเคราะห์งาน (Diagnostic) การควบคุมคุณภาพ

ระดับที่ 4: Control level เป็นระดับของคอนโทรลเลอร์เช่น RC (Robotic Controller), CNC, PLC (Programmable logic Controller)

ระดับที่ 5: Sensor actuator level เป็นระดับของอุปกรณ์ทำงานและเซนเซอร์ซึ่งเป็นระดับล่างสุด

ลำดับการควบคุมระบบอัตโนมัติชั้นในงานต่าง ๆ

Factory level

- Ethernet TCP/IP เป็นโปรโตคอลที่คนส่วนใหญ่จะคุ้นเคยดีในเรื่องของ Internet อุปกรณ์ทั้งซอฟต์แวร์ และฮาร์ดแวร์ก็เป็นสิ่งที่ทุกคนคุ้นเคยอยู่แล้วเพราะว่าใช้กันทั่วไปในสำนักงาน โดยทั่วไปแล้วเราจะใช้เทคโนโลยีอื่นนี้เพื่อเป็นการควบคุมระดับสูงสุดโรงงาน
- ประวัติของ TCP/IP นั้นมีกำเนิดมาจากระบบเครือข่าย ชื่อ ARPANET ในปี 1970 ซึ่งได้รับการสนับสนุนการวิจัยจากกระทรวงกลาโหมประเทศสหรัฐอเมริกา ต่อมาได้มีการพัฒนามาเป็นมาตรฐานในการรับ-ส่งข้อมูลผ่านระบบเครือข่ายอย่างเป็นทางการเมื่อวันที่ 1 มกราคม ค.ศ. 1983 ซึ่งปรากฏจำนวนระบบเครือข่าย และผู้ใช้เพิ่มขึ้นอย่างรวดเร็วจนเป็นที่รู้จัก ซึ่งถูกเรียกว่าเป็นระบบอินเทอร์เน็ต (Internet)

Control level

- ARCNET (Attached Resource Computer Network) ARCNET ถูกพัฒนามาตั้งแต่ปี 1977 โดยบริษัท SMC ประเทศสหรัฐอเมริกา
- Control Net ถูกพัฒนาโดยบริษัท Allen-Bradley ในปี 1995 เพื่อใช้ในงานควบคุม
- PROFIBUS-FMS (Profibus Fieldbus Message Specification) ถูกพัฒนาโดยบริษัท Siemens ประเทศเยอรมนี ในปี 1987 ในชื่อของ PROFIBUS (Process Fieldbus) และนำออกสู่ตลาดในปี 1992

Field level ในงาน Process automation

- FOUNDATION field bus ได้ถูกพัฒนามาบนพื้นฐานของมาตรฐาน IEC โดยได้รับการร่วมมือกันของ Noen World FIP และ ISP Foundation โดยจุดประสงค์เพื่อสร้างมาตรฐานฟิลด์บัสเพื่อให้สามารถใช้งานได้กับทุกยี่ห้อ
- HART (Highway Addressable Remote Transducer) ถูกพัฒนาขึ้นในปี 1984 เป็นโปรโตคอลสำหรับอุปกรณ์ระดับฟิลด์ที่ใช้มาตรฐาน 4-20 mA
- LON (Local Operating Network) เป็นระบบเน็ตเวิร์คสำหรับงานอัตโนมัติโดยใช้พื้นฐานของชิป “neuron” ซึ่งถูกแนะนำออกสู่ตลาดในปี 1990 โดยบริษัท Echelon ของประเทศสหรัฐอเมริกา
- PROFIBUS-PA (Profibus Process Automation) เป็น PROFIBUS ที่ใช้สำหรับงานควบคุมกระบวนการผลิต (Process control) โดยเฉพาะซึ่งจำเป็นต้องมีความปลอดภัยสูงมาก
- World FIP (World Factory Instrumentation Protocol) เป็นมาตรฐานเปิด (Open system) ตามมาตรฐาน UTE 46 ซึ่งถูกเสนอให้เป็นมาตรฐานตาม IEC และ ISA สมาคมผู้ใช้

เทคโนโลยี FIP ถูกตั้งขึ้นในปี 1993 โดยแกนนำหลัก ๆ คือ บริษัท Honeywell และ Bailey จากนั้นอีก 2 ปี (ต.ค. 1994) ได้รวมตัวกับสมาคมผู้ใช้เทคโนโลยี ISP (Interoperable System Project) โดยมีแกนนำหลัก ๆ คือ บริษัท Fisher – Rosemount, Siemens และ Yokogawa) รวมกันเป็นสมาคมฟิวด์บัส (Fieldbus Foundation)

Field level ในงาน Factory automation

Byte data

- **BITBUS** ถูกพัฒนาโดยบริษัท INTEL และถูกใช้งานตั้งแต่ปี 1984 เป็นไปตามมาตรฐาน IEEE 1118
- **CAN (Controller Area Network)** เริ่มต้นในปี 1980 โดยบริษัท Bosch เพื่อลดปริมาณสายไฟในรถเมอร์เซดีส-เบนซ์ จากนั้นโปรโตคอลนี้ได้ถูกพัฒนาเพื่อการใช้งานที่มากขึ้น
- **CAN open** อยู่ในตระกูล CAN ถูกพัฒนาโดยองค์การ CiA (CAN in Automation) ในปี 1993 เพื่อพัฒนาให้โปรโตคอล CAN มีความสามารถมากขึ้น จุดเด่นของ CAN open อยู่ต่างกับระบบ Fieldbus อื่น ๆ ก็คือ แต่ละโหนด (node) สามารถติดต่อกับโหนดโดยตรงได้โดยไม่ต้องจำเป็นต้องติดต่อผ่านตัวแม่ Master เป็นการลดจำนวนการติดต่อสื่อสารได้มาก
- **Device Net** เป็นมาตรฐานที่ถูกพัฒนามาจาก CAN โดยบริษัท Allen-Bradley ในปี 1994
- **INTERBUS-S** ถูกพัฒนาโดยบริษัท Phoenix Contact ในปี 1984 เป็นระบบ Field bus ที่ได้รับความนิยมเนื่องจากความเร็ว การวิเคราะห์ จุดบกพร่อง การกำหนด address โดยอัตโนมัติ
- **PROFIBUS-DP (Profibus Distributed Peripheral)** เป็น PROFIBUS ที่ใช้สำหรับงานควบคุมเครื่องจักร (Factory automation) จากข้อมูลการตลาด PROFIBUS ถือได้ว่าเป็นผู้นำทางด้าน Field bus เลยก็ว่าได้เพราะมีคนนิยมใช้มากกว่าครึ่งหนึ่งของตลาด field bus
- **SDS (Smart Distributed System)** ถูกพัฒนาโดยบริษัท Honeywell ซึ่งมีพื้นฐานมาจาก CAN

Bit data

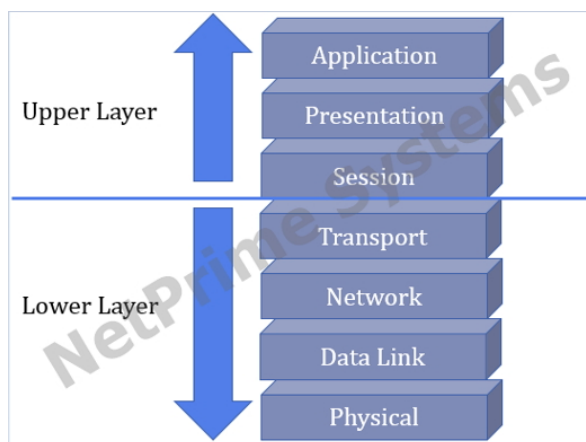
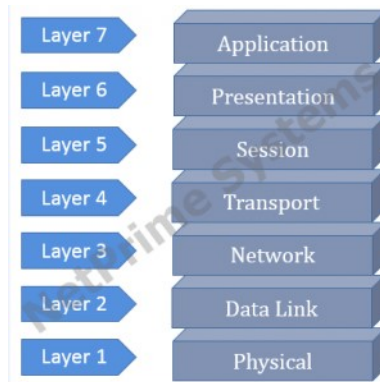
- **As interface**
- **INTERBUS LOOP** เป็น INTERBUS ที่ใช้งานระดับ Actuator sensor เพื่อติดต่อสื่อสารข้อมูลไม่กี่บิต
- **Seriplex** เป็นระบบบัสที่ใช้ควบคุมอุปกรณ์ระดับ Actuator sensor ถูกพัฒนาโดยบริษัท APC (Automated Process Control)

2/3: Open Systems Interconnection (OSI) Reference Model

องค์กรมาตรฐานระหว่างประเทศ (The International Organization for Standardization) หรือที่เราเรียกย่อๆว่า ISO (ไอโซ) ได้พัฒนา Model ของการทำงานบนระบบเครือข่ายขึ้นมาเป็นมาตรฐานกลาง ในปัจจุบันใช้เพื่ออ้างอิงการสื่อสารและเปรียบเทียบการทำงานบนเครือข่าย ผู้ผลิตหลายบริษัท ที่ผลิตอุปกรณ์หรือ software ต่างๆขึ้นมาก็ต้องผลิตตามมาตรฐานที่กำหนดขององค์ที่ดูแลในแต่ละส่วน ซึ่ง Model นี้ก็ถูกนำมาใช้มาเปรียบเทียบเพื่อให้เข้าใจและอธิบายการทำงานของแต่ละส่วนได้ง่ายขึ้น!!! ผมยกตัวอย่างของคำว่ามาตรฐานให้ดูง่ายๆครับ เช่น เวลาขับรถเกียร์ออโต้ เราเข้าไปที่เกียร์ D ก็คือไปข้างหน้า เข้าไปที่เกียร์ R ก็ถอยหลัง ไม่ว่าเราจะซื้อยี่ห้อไหนมา ก็จะถูกออกแบบให้เป็นมาตรฐานเดียวกัน และ Model ที่ผมพูดถึงนี้เรียกว่า Open System Interconnection (OSI) นั่นเอง !!!

อย่างที่ผมบอกนะครับ OSI Model เป็นเพียง Model ที่ใช้อ้างอิงในการสื่อสารเท่านั้น ปัจจุบันโลกเครือข่ายเราใช้งานบน TCP/IP เป็นหลักครับ แต่ที่ยังพูดถึง OSI Model เพราะว่าการแบ่งเป็น Layer ของมันสามารถมองเป็นภาพกว้างของการทำงานบนเครือข่ายในแต่ละส่วนได้ เพื่อมาใช้ในการสอนในการอธิบาย และในการแก้ปัญหาได้ง่ายขึ้นครับ

OSI Model ใช้อ้างอิงการสื่อสาร (Reference Model) แบ่งออกเป็นชั้น (Layer) โดยมีตั้งแต่ชั้นที่ 1 ถึงสวรรณชั้น 7 (Layer 1 – 7) โดย Layer 1 จะอยู่ด้านล่างสุด และเรียงขึ้นไปจนถึง Layer 7 แต่ละ Layer ก็มีชื่อเรียกตามรูปแบบการสื่อสารและการทำงานของมันในแต่ละชั้นนั่นเอง



บน OSI Model ก็จะแบ่งเป็น 2 กลุ่มใหญ่ คือ

- ตั้งแต่ Layer 1 – 4 เรียกว่า Lower Layer
- ตั้งแต่ Layer 5 – 7 เรียกว่า Upper Layer

โดยส่วนมากเราจะคุ้นเคยกันอยู่ที่ Lower Layer มากกว่า Upper Layer ครับ ใครที่ทำงานเป็น Network Engineer จะมองภาพออกครับว่า เราจะยุ่งอยู่กับ Lower Layer มากกว่า

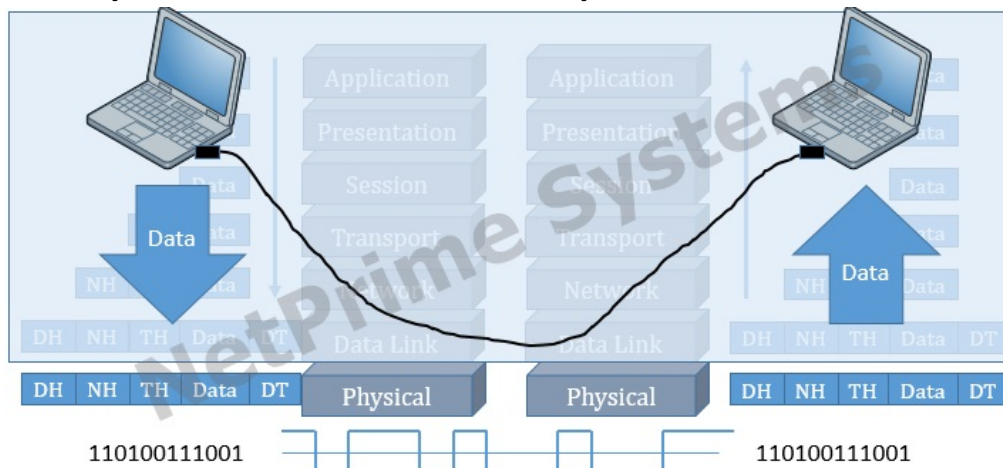
ที่นี้เรามาดูกันว่าในแต่ละ Layer มันมีหน้าที่อะไรบ้าง

Layer 1 (Physical Layer)

เป็น ชั้นล่างสุด จะมีการกำหนดคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้งสองระบบ เช่น

- สายที่ใช้รับส่งข้อมูลจะเป็นแบบไหน
- ข้อต่อที่ใช้ในการรับส่งข้อมูลมีมาตรฐานอย่างไร
- ความเร็วในการรับส่งข้อมูลเท่าใด
- สัญญาณที่ใช้ในการรับส่งข้อมูลมีรูปร่างอย่างไร
- ใช้แรงดันไฟฟ้าเท่าไร

ข้อมูลใน Layer ที่ 1 นี้จะมองเห็นเป็นการรับส่งข้อมูลที่ละบิตเรียงต่อกันไป



จากรูปแสดงถึงการส่งข้อมูลบน Physical layer ครบ แสดงให้เห็นว่า ข้อมูลจะมาเป็นอย่างไรรู้ก็ตาม ก็จะถูกแปลงเป็นสัญญาณเพื่อส่งไปยังปลายทาง แล้วฝั่งปลายทางก็จะนำสัญญาณที่รับมาแปลงกลับเป็นข้อมูลเพื่อส่งให้เครื่อง Client ต่อไป ยังไม่ได้พูดถึงการส่งข้อมูลบน OSI นะครับ เอาไว้รอบหน้าตรงนี้ผมยกขึ้นมาเพื่อให้เห็นภาพเบื้องต้น เพราะ ฉะนั้น อุปกรณ์ต่างๆที่มีความสามารถในการนำพาสัญญาณไป ก็พวก Card LAN (NIC), สาย UTP , สาย Fiber หรือพวก เต้าเสียบ หัวต่อต่างๆ RJ45 , RJ11 , RS323 ก็จัดอยู่ใน Physical Layer ครับ

Layer 2 (Data-Link Layer)

เป็นชั้นที่ทำหน้ากำหนดรูปแบบของการส่งข้อมูลข้าม Physical Network โดยใช้ Physical Address อ้างอิงที่อยู่ต้นทางและปลายทาง ซึ่งก็คือ MAC Address นั่นเอง รวมถึงทำการตรวจสอบและจัดการกับ error ในการรับส่งข้อมูล ข้อมูลที่ถูกส่งบน Layer 2 เราจะเรียกว่า Frame ซึ่งบน Layer 2 ก็ จะแบ่งเป็น LAN และ WAN ครับ ซึ่ง ปัจจุบัน บน Layer 2 LAN เรานิยมใช้เทคโนโลยีแบบ Ethernet มากที่สุด ส่วน WAN ก็จะมีหลายแบบแตกต่างกันไป เช่น Lease Line (HDLC , PPP) , MPLS , 3G และอื่นๆ

8 Bytes	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes
Preamble	Destination Address	Source Address	Type	Data	FCS

สำหรับ LAN ยังมีการแบ่งย่อยออกเป็น 2 sublayers คือ

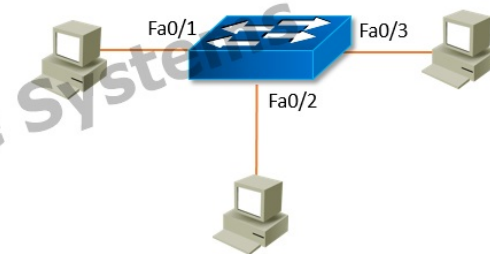
Logical Link Control (LLC)

IEEE 802.2 ซึ่งจะให้บริการกับ Layer ด้านบนในการเข้าใช้สัญญาณใน การรับ-ส่งข้อมูล ตามมาตรฐาน IEEE802 แล้ว จะอนุญาตให้สถาปัตยกรรมของ LAN ที่ต่างกันสามารถทำงานร่วมกันได้ หมายความว่า Layer ด้านบนไม่จำเป็นต้องทราบว่า Physical Layer ใช้สายสัญญาณประเภทใดในการรับ-ส่งข้อมูล เพราะ LLC จะรับผิดชอบในการปรับ Frame ข้อมูลให้สามารถส่งไปได้ในสายสัญญาณประเภทนั้นได้ และไม่จำเป็นต้องสนใจว่าข้อมูลจะส่งผ่านเครือข่ายแบบไหน เช่น Ethernet , Token Ring บลาๆ และไม่จำเป็นต้องรู้ว่าการส่งผ่านข้อมูลใน Physical Layer จะใช้การรับส่งข้อมูล แบบใด LLC จะเป็นผู้จัดการเรื่องเหล่านี้ได้ทั้งหมดครับ

Media Access Control (MAC)

IEEE 802.3 ใช้ควบคุมการติดต่อสื่อสารกับ Layer 1 และรับผิดชอบในการรับ-ส่งข้อมูลให้สำเร็จ และถูกต้อง โดยมีการระบุ MAC Address ของอุปกรณ์เครือข่าย ซึ่งใช้อ้างอิงในการส่งข้อมูลจากต้นทางไปยังปลายทาง เช่น จาก ต้นทางส่งมาจาก MAC Address หมายเลข AAAA:AAAA:AAAA ส่งไปหาปลายทางหมายเลข BBBB:BBBB:BBBB เมื่อปลายทางได้รับข้อมูลก็จะรู้ว่าใครส่งมา เพื่อจะได้ตอบกลับไปถูกต้อง นั่นเอง

Switch# show mac-address-table			
Mac Address Table			
Vlan	Mac Address	Type	Ports
1	0001.6420.b2ab	DYNAMIC	Fa0/2
1	0060.5c7d.49d4	DYNAMIC	Fa0/1
1	00e0.f785.e801	DYNAMIC	Fa0/3

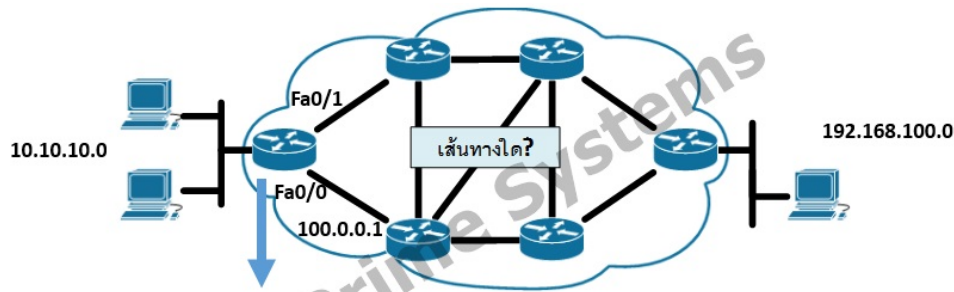


บน Ethernet (IEEE802.3) เมื่อมันมีหน้าที่ในการรับผิดชอบการรับ-ส่งข้อมูลให้สำเร็จและถูกต้อง มันจึงมีการตรวจสอบข้อผิดพลาดในการส่งข้อมูลด้วย ที่เราเรียกว่า Frame Check Sequence (FCS) และยังตรวจสอบกับ Physical ด้วยว่าช่องสัญญาณพร้อมสำหรับส่งข้อมูลไหม ถ้าว่างก็ส่งได้ ถ้าไม่ว่างก็ต้องรอ กลไกนี้เรารู้จักกันในชื่อ CSMA/CD นั่นเองครับ

CSMA/CD มันก็คือกลไกการตรวจสอบการชนกันของข้อมูล บน Ethernet ถ้าเกิดมีการชนกันเกิดขึ้น มันก็จะส่งสัญญาณ (jam signal) ออกไปเพื่อให้ทุกคนหยุดส่งข้อมูล แล้วสุ่มเวลา (back off) เพื่อส่งใหม่อีกครั้ง

Layer 3 (Network Layer)

ทำหน้าที่ส่งข้อมูลข้ามเครือข่าย หรือ ข้าม network โดยส่งข้อมูลผ่าน Internet Protocol (IP) โดยมีการสร้างที่อยู่ขึ้นมา (Logical Address) เพื่อใช้อ้างอิงเวลาส่งข้อมูล เราเรียกว่า IP address ข้อมูลที่ถูกส่งมาจากต้นทาง เพื่อไปยังปลายทาง ที่ไม่ได้อยู่บนเครือข่ายเดียวกัน จำเป็นจะต้องพึ่งพาอุปกรณ์ที่ทำงานบน Layer 3 นั่นก็คือ Router หรือ Switch Layer 3 โดยใช้ Routing Protocol (OSPF , EIGRP) เพื่อหาเส้นทางและส่งข้อมูลนั้น (IP) ข้ามเครือข่ายไป



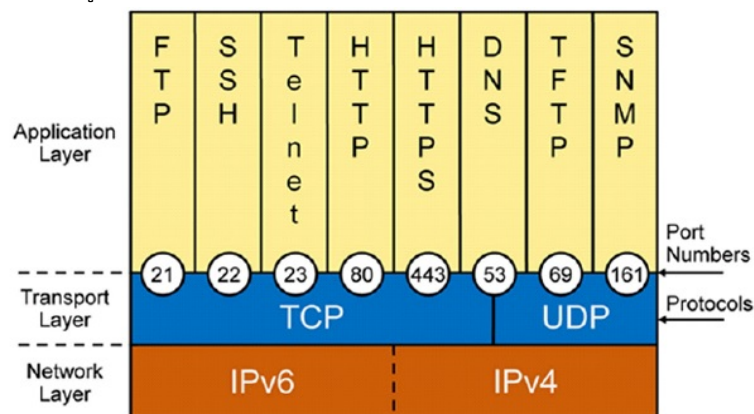
Network Protocol	Destination	Exit Interface	Next Hop
Connected	10.10.10.0	Fa0/1	
Learned	192.168.100.0	Fa0/0	100.0.0.1

โดยการทำงานของ Internet Protocol (IP) เป็นการทำงานแบบ Connection-less หมายความว่า IP ไม่มีการตรวจสอบข้อมูลว่าส่งไปถึงปลายทางไหม แต่มันจะพยายามส่งข้อมูลออกไปด้วยความพยายามที่ดีที่สุด (Best-Effort) เพราะฉะนั้น ข้อมูลที่ส่งออกไปแล้วไม่ถึงปลายทาง ต้นทางก็จะไม่รู้เลย ถ้าส่งไปแล้วข้อมูลไม่ถึงปลายทาง ฝั่งต้นทางจะต้องทำการส่งไปใหม่ บน Layer 3 จึงมี Protocol อีกตัวหนึ่งเพื่อใช้ตรวจสอบว่าปลายทางยังมีชีวิตอยู่ไหม ก่อนที่จะส่งข้อมูล นั่นคือ ICMP ครับ แต่ผู้ใช้งานจะต้องเป็นคนเรียกใช้ protocol ตัวนี้เองนะครับ

จริงๆ แล้ว ก็ยังมีรายละเอียดของ ICMP , ARP อีกที่ทำงานบน Layer 3 แต่เบื้องต้น เอาไว้เท่าที่ก่อนละกันครับ สำหรับ Layer 3

Layer 4 (Transport Layer)

ทำหน้าที่เชื่อมต่อกับ Upper Layer ในการใช้งาน network services ต่างๆ หรือ Application ต่าง จากต้นทางไปยังปลายทาง (end-to-end connection) ในแต่ละ services ได้ โดยใช้ port number ในการส่งข้อมูลของ Layer 4 จะใช้งานผ่าน protocol 2 ตัว คือ TCP และ UDP



เมื่อข้อมูลถูกส่งมาใช้งานผ่าน services Telnet ไปยังปลายทางถูกส่งลงมาที่ Layer 4 ก็จะทำให้การแยกว่า telnet คือ port number 23 เป็น port number ที่ใช้ติดต่อไปหาปลายทาง แล้วฝั่งต้นทางก็จะ random port number ขึ้นมา เพื่อให้ปลายทางสามารถตอบกลับมาได้เช่นเดียวกัน

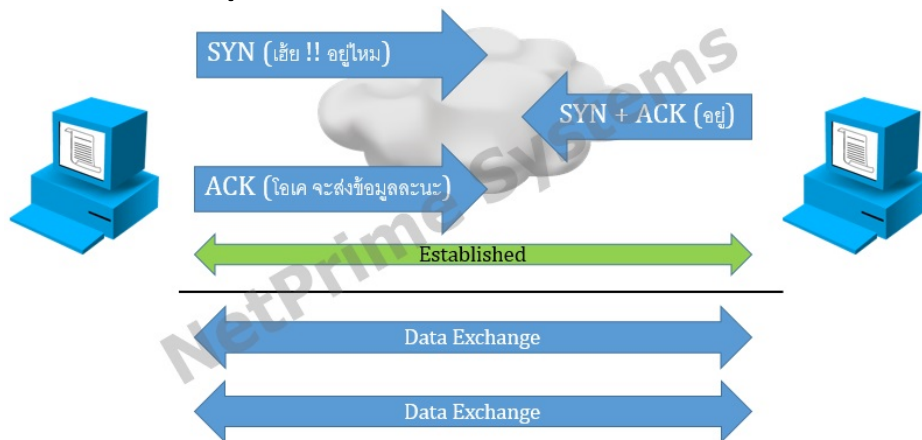


ที่นี้เรามาดู protocol 2 ตัว ใน Layer 4 กันต่อครับ

Transmission Control Protocol (TCP) มีคุณลักษณะที่สำคัญ ดังนี้

- จัดแบ่งข้อมูลจากระดับ Application ให้มีขนาดพอเหมาะที่จะส่งไปบนเครือข่าย (Segment)
- มีการสร้าง Connection กันก่อนที่จะมีการรับส่งข้อมูลกัน (Connection-oriented)
- มีการใช้ Sequence Number เพื่อจัดลำดับการส่งข้อมูล
- มีการตรวจสอบว่าข้อมูลที่ส่งไปถึงปลายทางหรือไม่ (Recovery)

บน TCP ก่อนจะส่งข้อมูลนั้นจะต้องทำการตรวจสอบก่อนว่า ปลายทางสามารถติดต่อได้ โดยจะทำการสร้างการเชื่อมต่อระหว่างผู้ส่งและผู้รับก่อน โดยใช้กลไก Three-Way Handshake เพื่อให้แน่ใจว่าข้อมูลที่ส่งจะสามารถส่งถึงผู้รับแน่นอน



นอกจาก Three-Way Handshake แล้ว TCP ยังมีกลไก Flow Control เพื่อควบคุมการส่งข้อมูลเมื่อเกิดปัญหาบนเครือข่ายระหว่างที่ส่งข้อมูลอยู่ หรือ กลไก Error Recovery ในกรณีที่มีข้อมูลบางส่วนหายไปขณะส่ง ก็ให้ทำการส่งมาใหม่ (Retransmission) แต่ผมขอพูดเรื่องกลไกต่างๆไว้เพียงเบื้องต้นละกันนะครับ

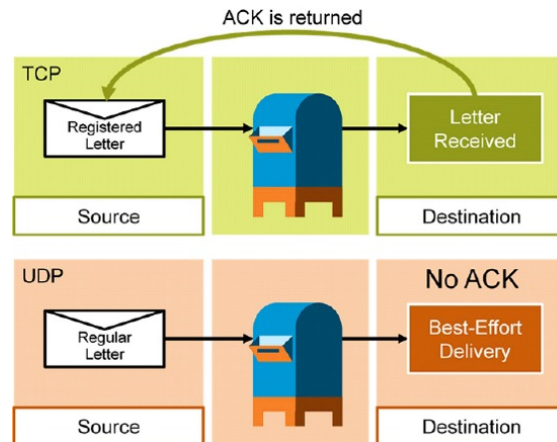
นอกจากนั้นยังสามารถทำการจัดสรรหรือแบ่งส่วนของข้อมูลออกเป็นส่วนๆ (Segmentation) ก่อนที่จะส่งลงไปที Layer 3 อีกด้วย และข้อมูลที่ถูกแบ่งออก ก็จะใส่ลำดับหมายเลขเข้าไป (Sequence number) เพื่อให้ปลายทางนำข้อมูลไปประกอบกันได้อย่างถูกต้อง

User Datagram Protocol (UDP) มีคุณลักษณะที่สำคัญ ดังนี้

- ไม่มีการสร้าง Connection กันก่อนที่จะมีการรับส่งข้อมูลกัน (Connectionless)
- ส่งข้อมูลด้วยความพยายามที่ดีที่สุด (Best-Effort)
- ไม่มีการตรวจสอบว่าข้อมูลที่ส่งไปถึงปลายทางหรือไม่ (No Recovery)

บน UDP จะตรงข้ามกับ TCP เลยครับ เพราะ ไม่มีการสร้างการเชื่อมต่อกันก่อน หมายความว่า ถ้า services ใดๆ ใช้งานผ่าน UDP ก็จะถูกส่งออกไปทันทีด้วยความพยายามที่ดีที่สุด (Best-Effort) และไม่มีการส่งใหม่เมื่อข้อมูลสูญหาย (No Recovery) หรือส่งไม่ถึงปลายทางอีกด้วย ข้อดีของมันก็คือ มีความรวดเร็วในการส่งข้อมูล เพราะฉะนั้น services ที่ใช้งานผ่าน UDP ก็มีมากมาย เช่น TFTP , DHCP , VOIP และอื่นๆ เป็นต้น

ผมชอบภาพนี้ เพราะเปรียบเทียบการทำงานของ TCP และ UDP ได้เห็นภาพดีเหมือนกันครับ



อ้าวๆ เหมือนกันหรือยังครับ ผ่านไปแล้ว สำหรับ Lower Layer นะครับ คือ Layer 1 – 4 ซึ่ง ส่วนมากแล้วชาว Network Engineer ก็จะวุ่นวายกับ Lower Layer มากกว่า Upper Layer (5-7) ซะอีก แต่จะไม่พูดถึงก็ไม่ได้ใช่ไหมครับ ถ้าอย่างนั้นก็ไปดูกันต่อเลยครับ

Layer 5 (Session Layer)

ทำหน้าที่ควบคุมการเชื่อมต่อ session เพื่อติดต่อจากต้นทาง กับ ปลายทาง ลองดูจากรูป ตัวอย่างเพื่อจะทำความเข้าใจง่ายขึ้น (รีเปล่า) 55+



เมื่อฝั่งต้นทางต้องการติดต่อไปยังปลายทางด้วย port 80 (เปิด Internet Explorer) ฝั่งต้นทาง ก็จะมีการติดต่อไปยังปลายทาง โดยการสร้าง session ขึ้นมา เป็น session ที่ 1 ส่งผ่าน Layer 4 โดย random port ต้นทางขึ้นมาเป็น 1025 ส่งไปหาปลายทางด้วย port 80 ระหว่าง ที่ session ที่ 1 ใช้งานอยู่ เราติดต่อไปยังปลายทางอีกครั้งด้วย port 80 (เปิด Google Chrome) ฝั่งต้นทางก็จะทำการสร้าง session ที่ 2 ขึ้นมา ส่งผ่าน Layer 4 โดย random port ต้นทางขึ้นมาเป็น 1026 ส่งไปหาปลายทางด้วย port 80 แล้วแต่ละ session ฝั่งปลายทาง ก็จะตอบกลับมาจาก port ที่ฝั่งต้นทางส่งมา ทำให้สามารถแยก session ออกได้ เมื่อเราส่งข้อมูลบนเครือข่ายนั่นเองครับ

Layer 6 (Presentation Layer)

ทำหน้าที่ในการแปล หรือ นำเสนอ structure , format , coding ต่างๆของข้อมูลบน application ที่จะส่งจากต้นทางไปยังปลายทาง ให้อยู่ในรูปแบบที่ฝั่งต้นทางและปลายทาง สามารถเข้าใจได้ทั้ง 2 ฝั่ง

ผมอ่านจากเว็บนึงกล่าวไว้ก็เข้าใจง่ายขึ้นครับ

Most computers use the ASCII table for characters. If another computer would use another character like EBCDIC than the presentation layer needs to "reformat" the data so both computers agree on the same characters.

เค้าบอกว่า computer ส่วนมากใช้ ASCII format ถ้าจะติดต่อพวก computer mainframe จะใช้พวก EBCDIC format ซึ่ง Layer 6 ก็จะทำ reformat ข้อมูล ซึ่งสามารถทำให้ทั้ง 2 เครื่องสามารถเข้าใจ format ข้อมูลที่จะสื่อสารกันได้

Layer 7 (Application Layer)

ทำหน้าที่ติดต่อระหว่างผู้ใช้ (user) กับ application ที่ใช้งานบนเครือข่าย เช่น Web Browser (HTTP) , FTP , Telnet เป็นต้น สรุปแล้วมันก็คือพวก application ที่ใช้งานผ่าน network นั่นเองครับ

3/3 BIG DATA ในอุตสาหกรรมการผลิต (MANUFACTURING)

ภายในปี 2020 จะมี devices ที่ connected กัน มากกว่า 30,000 ล้านชิ้น

-> แสดงว่า IoT จะมีส่วนสำคัญในการทำ (Big) Data Analytics และ การทำ cloud/fog computing ก็จะเป็นพื้นฐานของการเก็บ ประมวลผลข้อมูล

ภายในปี 2018 ข้อมูล 65% ของโรงงานอุตสาหกรรม ที่มี มากกว่า 10 โรงงานจะลงทุนกับ Operational Intelligence

->แสดงว่ามีระบบที่สามารถ ลด ค่าใช้จ่าย จะเป็นที่นิยมมากๆ (IoT + Data Analytics + Predict waste)

ย้อนกลับไปราว 20 ปีก่อนใครจะไปเชื่อว่าแม้แต่ข้อมูลที่เกิดขึ้นที่จุดขายบนเครื่อง POS (Point of Sales System) ก็ไม่ได้ถูกจัดเก็บครบทั้งปี เนื่องจากต้นทุนของสื่อจัดเก็บข้อมูลยังแพงอยู่มาก เวลาผ่านไปอย่างรวดเร็ว จินตนาการแบบในภาพยนตร์ซีรีส์ Person of Interest ที่เครื่องจักรอัจฉริยะสามารถเข้าถึงข้อมูลทุกอย่าง และทำการประมวลผลพยากรณ์ความเสี่ยงด้านความมั่นคงล่วงหน้าเริ่มเข้าใกล้ความเป็นจริงมากขึ้น แม้จะไม่ได้ไปไกลถึงขนาดนั้นแต่องค์กรขนาดใหญ่ ทั้งภาครัฐและเอกชนในหลายวงการก็ได้ใช้ทั้งโมเดลและเทคโนโลยี Big Data มาระยะหนึ่งแล้ว

Big Data ได้เชื่อมรวมเอางานข้อมูล สถิติ การวิจัย และไอซีที่เข้ามาไว้ด้วยกัน ซึ่งองค์กรขนาดใหญ่ในหลายวงการที่การพยากรณ์มีความสำคัญยิ่งยวดในเชิงพันธกิจ (Mission Critical) ต่อความสำเร็จและล้มเหลวของธุรกิจ (เช่น ธุรกิจประกัน การค้าหลักทรัพย์ และการค้าเงินตราต่างประเทศ) รวมถึงองค์กรภาครัฐที่การพยากรณ์อาจมีผลต่อความเป็นความตายของประชาชน (เช่น การพยากรณ์อากาศ) ได้นำมาใช้มาระยะหนึ่งแล้ว

แต่ด้วยการพัฒนาขององค์ประกอบที่จำเป็นสำหรับ Big Data หลายๆ ด้านทั้งเรื่องสื่อจัดเก็บข้อมูลที่มีขนาดใหญ่ขึ้นแต่ราคาถูกลง และความสามารถในการประมวลผลข้อมูลปริมาณมหาศาลได้ด้วยต้นทุนที่องค์กรต่างๆ สามารถเข้าถึงได้ เช่น การใช้สถาปัตยกรรมการประมวลผลแบบกระจายตัว (Distributed Computing) และการใช้ทรัพยากรประมวลผลในฐานะบริการบนคลาวด์ (Cloud Computing) ซึ่งองค์ประกอบที่สนับสนุนการก้าวสู่ยุคของ Big Data เหล่านี้เป็นสัญญาณที่บอกองค์กรทั้งหลายว่า ถ้าคุณไม่รีบนำเอาโซลูชัน Big Data มาใช้กับธุรกิจของคุณ ความสามารถทางการแข่งขันของ

องค์กรคุณน่าจะไม่มีทางตามทันองค์กรที่นำเอา Big Data มาใช้ และคุณอาจจะแปลกใจว่าพวกเขาเหล่านั้นช่างรู้กระแสตลาด และความต้องการของลูกค้าอย่างรวดเร็วเหมือนมีเทพพยากรณ์เป็นพนักงานหรือกรรมการบริหารองค์กรอย่างไรง่ายๆจนเหลือที่เดียว

องค์ประกอบขององค์กรซึ่งพร้อมที่จะดำเนินกลยุทธ์ Big Data

1. มีการวางแผนในการติดตามตรวจสอบ รวบรวม จัดเก็บ และบริหารจัดการข้อมูลในระดับ Big Data
2. มีการใช้เทคโนโลยีใหม่ๆ ที่จะจัดการติดตามตรวจสอบและจัดเก็บข้อมูลระดับ Big Data ได้
3. มีโซลูชัน Big Data ที่มีความพร้อมทั้งพื้นที่จัดเก็บข้อมูล พลังการประมวลผล การประยุกต์ใช้งาน และระบบรายงานถูกต้องเหมาะสม
4. มีบุคลากรที่มีความเชี่ยวชาญในวิทยาศาสตร์และคณิตศาสตร์การพยากรณ์ เพื่อการบริหารที่เหมาะสมกับงานโซลูชัน Big Data

เตรียมพร้อม Big Data & Data Analytics

แน่นอนว่าเมื่อได้ยิ้องค์กรมากมายต่างต้องการใช้โซลูชัน Big Data เพื่อเพิ่มความสามารถทางการแข่งขันของตนเอง คุณก็คงจะเป็นคนหนึ่งที่อยากจะนำเอาเทคโนโลยี Big Data มาใช้บ้าง แต่ Big Data ไม่ได้เหมือนเรื่องพลังการประมวลผลของเซิร์ฟเวอร์ หรือขนาดพื้นที่จัดเก็บข้อมูล การจะใช้โซลูชัน Big Data ให้ได้ผลจริงๆ อยู่ที่ปริมาณและธรรมชาติข้อมูลขององค์กรคุณอยู่ในระดับ Big Data แล้วหรือยัง ปริมาณข้อมูลที่มีการจัดเก็บ ประเภทของข้อมูล ความครบถ้วนและหลากหลายของข้อมูลที่เกี่ยวข้องกับธุรกิจ ซึ่งความสมบูรณ์พร้อมนี้เท่านั้นที่จะสามารถนำเอาโซลูชัน Big Data มาแยกเอาข้อมูลออกเป็นกลุ่มตามความเหมือนและต่าง และหาความสัมพันธ์ระหว่างกันจนสามารถหาความหมายหรือคำอธิบายที่อาจบ่งบอกถึงทิศทางหรือเทรนด์ความต้องการ ความชอบ หรือพฤติกรรมของลูกค้าที่กำลังเปลี่ยนไปได้

ถ้าจะให้คำอธิบายหรือขยายความง่าย ๆ เกี่ยวกับ Big Data ก็คือ การวิเคราะห์ข้อมูลใหม่เพื่อหาข้อสรุปได้อย่างทันทีทันใด ด้วยเทคโนโลยีสารสนเทศยุคใหม่ที่มีความสามารถอย่างเหลือเชื่อ การวิเคราะห์ข้อมูลนั้นเริ่มต้นจากการต่อยอดของสถิติเชิงวิเคราะห์ (Analytical Statistics) ที่เดิมก็ถูกนำมาใช้ในองค์กรขนาดใหญ่หรือเอนเตอร์ไพรส์อยู่บ้างแล้ว ในรูปของ Data Analytics ซึ่งข้อแตกต่างกับสถิติเชิงวิเคราะห์ก็คือ เราไม่เน้นการตั้งสมมติฐานก่อนแบบงานวิจัยเชิงสถิติ แต่ดูจากรูปแบบและความสัมพันธ์ของชุดข้อมูลที่ออกมา โดยปล่อยให้งานทางด้านการสร้างโมเดลเพื่อหาคำตอบของสมมติฐาน และการใช้เครื่องมือทางสถิติอื่นๆ เป็นงานของนักสร้างโมเดลคณิตศาสตร์ และหลายครั้งที่จริงๆ ที่ทีมงานอาจไม่มีคำถามอะไรก่อนการวิเคราะห์ชุดข้อมูลเลยด้วยซ้ำ แต่อาศัยความสามารถของโซลูชัน Big Data และนักวิเคราะห์ข้อมูลที่จะเห็นรูปแบบความสัมพันธ์บางอย่างจากชุดข้อมูลที่มีปริมาณมหาศาลและมีหลากหลายของ Big Data และหาข้อสรุปให้แก่ผู้บริหารในองค์กรต่อไป

และเมื่อถึงยุคของ Big Data ปริมาณ ประเภท ความหลากหลายของข้อมูลทุกอย่างที่องค์กรมี ไม่ใช่เพียงแค่เฉพาะที่ได้จากการจัดเก็บโดยส่วนงานที่เกี่ยวข้อง เช่น ส่วนงานขาย การผลิต การเงินและบัญชี หรือระบบการบริหารการซื้อที่จุดขาย (POS) แต่อาจรวมถึงการใช้เทคโนโลยีด้านข้อมูลใหม่ๆ เช่น RFID, Geo Tag, Semantic Web, ฯลฯ นั่นหมายถึงรากฐานของแหล่งกำเนิดข้อมูลในองค์กรนั้นๆ มีความเป็นข้อมูลระดับ Big Data อย่างแท้จริง และด้วยการใส่ความพยายามที่จะตรวจจับ ติดตาม จัดเก็บ บริหารและวิเคราะห์ข้อมูลอย่างครอบคลุมมากที่สุดด้วยเครื่องมือต่างๆ ที่กล่าวมา นั้นจะทำให้ความพร้อมในการวิเคราะห์ข้อมูลขององค์กรคุณขึ้นสู่ระดับ Big Data และแน่นอนว่าที่จริงแล้วก่อนจะไปสนใจกับปลายทางของการวิเคราะห์ข้อมูลระดับ Big Data องค์กรของคุณจึงต้องใส่ใจกับการติดตาม จัดเก็บ และบริหารจัดการข้อมูลให้เป็น Big Data อย่างแท้จริงเสียก่อนด้วย เพื่อที่จะมีข้อมูลที่เกี่ยวข้องและจำเป็นไว้ครอบคลุมครบถ้วน และมีปริมาณมากพอที่จะนำมาใช้ด้วย

แหล่งข้อมูลสำหรับ Big Data มีอะไรบ้าง?

1. ข้อมูลการใช้งานเว็บไซต์
2. ข้อมูลการซื้อเนื้อหา บริการ และสินค้าผ่านอีคอมเมิร์ซ
3. ข้อมูล POS
4. ข้อมูลการขนส่ง
5. ข้อมูลคลังสินค้า
6. พฤติกรรมการในร้านค้า / จุดจำหน่ายสินค้าของลูกค้า
7. ข้อมูลการซื้อของลูกค้า + ข้อมูลรายละเอียดเกี่ยวกับองค์กร/ตัวลูกค้า
8. ฯลฯ

ตัวอย่างสิ่งที่จะได้จาก Big Data

1. แนวโน้มการตลาด
2. พฤติกรรมลูกค้าทั้งในร้านค้าและออนไลน์ (Buying Pattern, Internet Behavior, ฯลฯ)
3. ความสัมพันธ์ระหว่างรายละเอียดลูกค้าและความต้องการสินค้า/บริการ
4. การเปลี่ยนแปลงของอัตราส่วนลูกค้าแต่ละกลุ่มแบ่งตามอายุ และความชอบของกลุ่มอายุต่างๆ
ที่จะมีผลกระทบต่อคุณสมบัติสำคัญที่ผลิตภัณฑ์หรือบริการจะต้องมี หรือต้องมุ่งเน้นเป็นพิเศษ
5. ความสัมพันธ์เชิงภูมิศาสตร์กับคุณสมบัติของสินค้า / บริการในความต้องการ
6. อิทธิพลของผู้ใช้ในเครือข่าย Social Media
7. ความรู้สึกของลูกค้าต่อแบรนด์บน Social Media (Customer Sentiment)
8. ผลกระทบยอดขายที่ลดลงของลูกค้ารายที่กำหนดต่อความเสี่ยงทางการเงินของบริษัท
9. การเพิ่มหรือลดสินค้าในแต่ละกลุ่มสินค้า (Categories) และการเพิ่มลดกลุ่มย่อยกับผลต่อรายได้
และกำไรของบริษัท
10. การวิเคราะห์ความเสี่ยง (Risk Analysis) และการตรวจจับการฉ้อโกง (Fraud Detection)

ก่อนจะมาเป็น Big Data

ที่จริงแล้วการคิดเอาโซลูชัน Big Data มาใช้งานก็ไม่ได้เป็นสิ่งที่องค์กรต้องเริ่มจากศูนย์ เพราะที่จริงแล้วองค์กรมีการบริหารจัดการข้อมูลในระดับใดระดับหนึ่งอยู่แล้วไม่มากนัก ซึ่งถ้าจะอ้างอิงถึงเราคงคุ้นเคยกับคำศัพท์ เช่น Database, Data Warehouse และ Data Mining กันอยู่บ้างแล้ว

ในอดีตการบริหารจัดการข้อมูลในองค์กรมักเป็นการบริหารเชิงรับ คือนำเอาข้อมูลที่ต้องใช้งานเข้ามาจัดเก็บ และบริหารจัดการให้ค้นหาได้ง่ายเมื่อต้องการ และมีการวิเคราะห์โดยพื้นฐานในระดับหนึ่ง นั่นคือสิ่งที่ Database ทำในยุคแรกๆ ซึ่งการวิเคราะห์ส่วนใหญ่ก็ขึ้นอยู่กับข้อมูลที่มีโครงสร้าง (Structured Data) และการวิเคราะห์หลายครั้งก็อิงกับสมมติฐานและการคาดการณ์ที่ผู้บริหารในองค์กรพอจะทราบหรือคาดเดาได้อยู่บ้างแล้ว

แต่เมื่อมาถึงยุคของ Data Mining เป้าหมายของการใช้งาน Data Mining คือการค้นหารูปแบบหรือความสัมพันธ์ของข้อมูลที่องค์กรไม่เคยรู้มาก่อน และจากแนวคิดของ Data Mining ก็ได้มีการต่อยอดมาถึงจุดที่องค์กรพยายามที่จะทำให้ความสามารถในการวิเคราะห์ดังกล่าวเกิดขึ้นกับข้อมูลที่มีปริมาณมหาศาล และไม่เพียงแค่นำมาดำเนินการกับข้อมูลที่มีการรวบรวม คัดเลือกและจัดการจนเป็นกลุ่มข้อมูลที่ต้องการ (Batch Processing) แต่ต้องสามารถดำเนินการกับข้อมูลที่มีการใช้งานในเวลาจริงได้ด้วย เช่น Streaming Data เป็นต้น

ความก้าวหน้าสูงสุดก็คือ Big Data ให้ความสำคัญและความสนใจที่ครอบคลุมเรื่องกลยุทธ์ข้อมูล และสารสนเทศเชิงกลยุทธ์ที่มากกว่า ทุกแนวคิดด้านการบริหารจัดการข้อมูลที่ผ่านมา Big Data สนใจแม้กระทั่งว่า คุณจะหาวิธีได้มาซึ่งข้อมูลเพื่อนำมาวิเคราะห์ได้อย่างไร เช่น หาวิธีเข้าไปดึงข้อมูลจาก

แหล่งข้อมูลบนอินเทอร์เน็ตของหน่วยงานที่เกี่ยวข้องมาด้วยสคริปต์ที่เขียนขึ้นอย่างชาญฉลาด หรือหา RFID หรืออุปกรณ์ในการวัดค่าที่ต้องการในแบบอื่นๆ เพื่อให้ได้ข้อมูลที่ต้องการ และนำมาประมวลผลเพื่อตอบโจทย์ Big Data ที่ต้องการ

ผู้สังเกตการณ์และผู้เชี่ยวชาญหลายท่านกล่าวว่า แนวคิดหนึ่งที่คนที่จะดูแลส่วนงาน Big Data ต้องมีคือ การรับรู้ถึงเป้าหมายเชิงธุรกิจที่เป็นที่มาของกลยุทธ์ Big Data แล้วพัฒนากลยุทธ์ Big Data ที่เหมาะสมสำหรับการตอบเจอนั้นๆ ขึ้นมา โดยมุ่งเน้นที่การตอบสนองการตอบเจอยทางธุรกิจ ไม่ว่าจะเป็นกลยุทธ์ที่เกิดขึ้นในระยะสั้น หรือเป็นการดำเนินการต่อเนื่องในระยะยาวก็ตาม ตามแนวคิดนี้เท่ากับว่า Big Data ไม่ใช่แค่โซลูชันด้านไอทีและสารสนเทศขององค์กร แต่ก้าวข้ามไปเป็นกลยุทธ์เชิงรุกของธุรกิจ ซึ่งหากไปได้ถึงจุดนั้นก็เท่ากับองค์กรสามารถเพิ่มความสามารถทางการแข่งขันด้วยกลยุทธ์ Big Data เชิงรุกได้ ซึ่งมากกว่าแค่การรอให้ข้อมูลในระบบขององค์กรแสดงแนวโน้มอะไรบางอย่างที่น่าสนใจขึ้นมา เพราะนั่นอาจจะช้าเกินไปสำหรับการแข่งขันก็ได้

วิทยาศาสตร์-คณิตศาสตร์

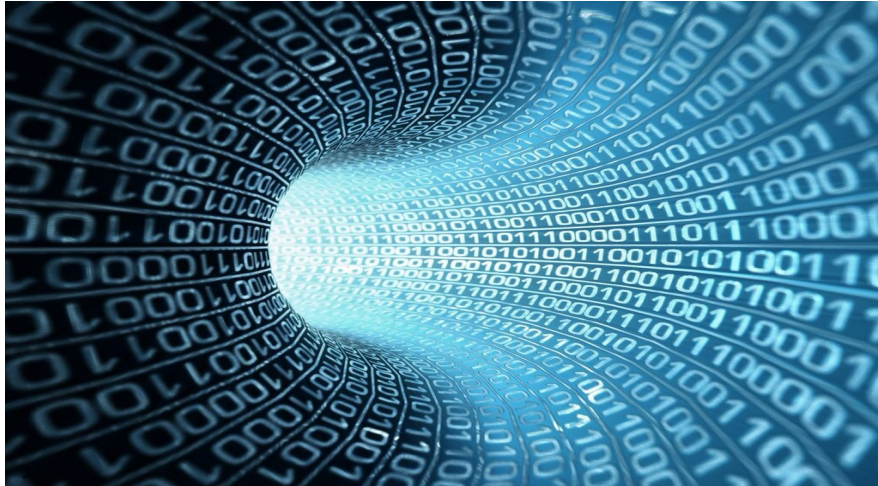
กลจักรสำคัญทำ Big Data

ถึงแม้เทคโนโลยีการจัดเก็บและวิเคราะห์ข้อมูล Big Data จะมีความก้าวหน้าเป็นอย่างมาก แต่คงไม่ใช่เรื่องง่ายที่จะให้ระบบสามารถเข้าไปวิเคราะห์ข้อมูลมหาศาลในแต่ละองค์กรแล้วแจ้งผลที่น่าสนใจต่างๆ ได้ทันที เหตุผลหลักๆ ก็คือ ความแตกต่างของแต่ละองค์กรในเรื่องของข้อมูลที่มีการติดตามจัดเก็บและบริหารจัดการ รวมถึงประเภทและธรรมชาติของธุรกิจที่องค์กรนั้นๆ ดำเนินการด้วย

ถ้าเริ่มต้นจากสมมติฐานที่ว่า ฝ่ายบริหารของทุกองค์กรสามารถตั้งคำถาม หรือโจทย์ทางธุรกิจที่ต้องการทราบจากข้อมูลที่บริษัทมีได้ (ดูตัวอย่างจากลอมกรอบ “ตัวอย่างของสิ่งที่ได้จาก Big Data”) หรือยิ่งไปกว่านั้นคือ สามารถตั้งคำถามทางธุรกิจเพื่อการทำกลยุทธ์ Big Data เชิงรุกได้ตามที่ได้นำเสนอไปในหัวข้อก่อนนี้แล้ว จากนั้นทีมงานได้หาวิธีที่จะได้มาซึ่งข้อมูลที่ต้องการแล้ว แต่การนำชุดข้อมูลดังกล่าวใส่เข้าไปในระบบยังไม่สามารถทำให้ได้คำตอบที่ฝ่ายธุรกิจต้องการออกมาได้ หากขาดสิ่งที่เป็สมองของโซลูชัน Big Data ไป

สิ่งที่องค์กรต้องมีเพิ่มเติมนอกเหนือจากข้อมูลที่หลากหลายและมีปริมาณมากพอ กับโซลูชัน Big Data ที่เหมาะสมก็คือผู้เชี่ยวชาญที่มีความรู้ในศาสตร์ที่เกี่ยวข้องกับการทำ Big Data ซึ่งไม่ได้หมายถึงผู้บริหาร ผู้เชี่ยวชาญทางธุรกิจ-เศรษฐกิจ และผู้เชี่ยวชาญด้านไอทีที่องค์กรน่าจะมีอยู่แล้ว แต่เป็นผู้เชี่ยวชาญทางด้านศาสตร์สถิติและคณิตศาสตร์เพื่อการบริหาร ซึ่งจะเป็นกลุ่มบุคคลที่มีความจำเป็นอย่างมากในการทำงานด้าน Big Data

เช่นเดียวกับงานของผู้เชี่ยวชาญสาขาวิทยาศาสตร์อื่นๆ เช่น วิศวกรรม แพทย์ เกษษกรรม ฯลฯ ที่อธิบายให้กับคนอื่นเข้าใจได้ยาก แต่หลายคนก็คงมีคำถามว่างานของนักวิเคราะห์ข้อมูล นักคณิตศาสตร์ สถิติ หรือผู้เชี่ยวชาญในเชิงข้อมูลอื่นๆ ที่จะทำให้กับโซลูชัน Big Data นี่คืออะไรบ้าง อาจสรุปได้ว่างานเชิงสถิติ วิจัยและวิเคราะห์ข้อมูลทุกประเภทคืองานที่เหล่าผู้เชี่ยวชาญด้านนี้สามารถทำให้ธุรกิจผ่านโซลูชัน Big Data ได้ โดยหลักการทางสถิติที่สำคัญที่ต้องนำมาใช้ในงาน Big Data คือสถิติอนุมาน (Inductive Statistics) สำหรับการบ่งชี้ระบบที่ไม่เชิงเส้น (Nonlinear system identification) หากท่านใดสนใจที่จะศึกษาในเชิงลึกก็สามารถหาอ่านตามหัวข้อดังกล่าวได้



ทั้งนี้งานของนักคณิตศาสตร์สถิติ นักคณิตศาสตร์สารสนเทศ หรือนักวิเคราะห์ข้อมูลสามารถทำให้องค์กรด้วยโซลูชัน Big Data ได้มีรายละเอียดดังต่อไปนี้

1. การตั้งสมมติฐานแล้วใช้เครื่องมือสถิติพื้นฐาน อย่าง Z-test หรือ Chi-Square ในการพิสูจน์
2. การวิเคราะห์จากกราฟหรือตาราง เพื่อดูรูปแบบ เช่น Seasonal, Cycle, Pattern อื่นๆ และความสัมพันธ์ ซึ่งเป็นเครื่องมือที่พื้นฐานที่สุด แต่ใช้การได้มากมายที่สุดตัวหนึ่งในการวิเคราะห์ข้อมูล โดยมีคำศัพท์เรียกกันอย่างไม่เป็นทางการว่า “การพลิกข้อมูลดูในหลายๆ มุมที่แตกต่าง”
3. การตั้งโมเดลด้วยเทคนิคทางปัญญาประดิษฐ์ (AI) หรือ Machine Learning เช่น Bayesian Network หรือ K-Means Clustering
4. การทำบทสรุปการวิเคราะห์ ซึ่งไม่ได้หมายถึงการทำกราฟและข้อสรุปเชิงบรรยาย เช่น การทำงานวิจัยทั่วไป แต่ต้องอาศัยความสามารถในการแสดงผลข้อมูลออกมาเป็นภาพ (Visualization) ของโซลูชัน Big Data เพื่อที่จะทำให้ได้ผลลัพธ์ที่ต้องการอย่างเป็นอัตโนมัติและสมบูรณ์แบบยิ่งขึ้น

อย่างไรก็ตาม ความเป็นไปได้สูงที่บริษัทซึ่งมีความสนใจในกลยุทธ์ Big Data จะก้าวจากองค์กรที่ใช้การวิจัยทางการตลาดทั่วไปไปสู่ Big Data โดยไม่เคยทำการวิเคราะห์ข้อมูลด้วยโซลูชันก้าวหน้าอื่นๆ เช่น Data Mining หรือ Data Warehouse แต่ก้าวสู่ Big Data เลย และสิ่งที่โซลูชัน Big Data สามารถให้ผลลัพธ์ได้จะรวดเร็วกว่าการตั้งสมมติฐานหรือคำถามทางธุรกิจของผู้บริหารจะตามทัน จนกลายเป็นหน้าที่ของผู้เชี่ยวชาญด้านการวิเคราะห์ข้อมูลที่จะต้องวิเคราะห์หาความสัมพันธ์ของชุดข้อมูลต่างๆ ที่เข้าสู่ระบบ เช่นเดียวกับต้องวางแผนให้องค์กรในการติดตามและจัดเก็บข้อมูลใหม่ๆ หรือด้วยวิธีการใหม่ๆ ที่จะสามารถนำมาใช้กับโซลูชัน Big Data ได้ในเวลาต่อมาด้วย

เทคโนโลยี Big Data

คำถามที่หลายคนอยากรู้อยากจะเป็นเรื่องของเทคโนโลยีว่ามีอะไรบ้างที่องค์กรจะต้องมีเพื่อที่จะแน่ใจได้ว่าจะสามารถดำเนินกลยุทธ์ Big Data ให้ได้ผลสัมฤทธิ์ตามหัวใจของกลยุทธ์อย่างควรจะเป็นอย่างแท้จริง

หัวใจหลักน่าจะเป็นส่วนของโครงสร้างพื้นฐานที่เกี่ยวข้อง ได้แก่ ส่วนจัดเก็บข้อมูลที่รองรับได้ทั้งปริมาณข้อมูลมหาศาล และมีความเร็วในการเข้าถึงที่สูงพอ รวมทั้งพลังการประมวลผลที่ทรงพลังพอจะรับการประมวลผลข้อมูลปริมาณมหาศาลของ Big Data ได้ ซึ่งแต่เดิมนั้นยากที่องค์กรทั่วไปจะสามารถจัดหา หรือเข้าถึงทรัพยากรที่จำเป็นเหล่านั้นได้ แม้แต่องค์กรระดับเอนเตอร์ไพรส์หลายแห่งก็เชื่อว่าได้ทรัพยากรเหล่านั้นมาในระดับที่มีความคุ้มค่าในเชิงต้นทุน

ปัจจัยหนึ่งในเรื่องเทคโนโลยีที่ทำให้โซลูชัน Big Data สามารถนำมาใช้ในองค์กรต่างๆ ได้อย่างกว้างขวางมากขึ้น คือเทคโนโลยีทั้งในเรื่องการประมวลผลและการจัดเก็บข้อมูลผ่านคลาวด์ (Cloud Service) เช่นเดียวกับการสื่อสารอินเทอร์เน็ตความเร็วสูง ซึ่งพัฒนาขึ้นมาจากถึงระดับที่พร้อมใช้งานกับ Big Data มาระยะหนึ่งแล้ว ซึ่งนั่นทำให้โซลูชันที่เป็นองค์กรประกอบต่างๆ ทั้งหมดถูกพัฒนาขึ้นมาให้ตอบสนององค์กรทั่วไปได้มากยิ่งขึ้น พุดง่าย ๆ องค์กรที่เดิมไม่มีทางจะเข้าถึงโครงสร้างพื้นฐาน และทรัพยากรที่จำเป็นสำหรับการทำ Big Data ก็จะสามารถดำเนินการได้ในยุคปัจจุบันนี้

สำหรับรายละเอียดการพัฒนาของโซลูชันที่ทำให้ Big Data น่าจะสามารถกลายมาเป็นโซลูชันกระแสหลักสำหรับองค์กรชั้นนำนั้นก็เช่น ส่วนได้ตอบกับผู้ใช้ (UI) ซึ่งเริ่มมีความง่ายในการใช้ตั้งแต่ขั้นตอนการติดตั้งและกำหนดค่าต่างๆ แม้แต่ในเรื่องการตั้งค่าในการวิจัยและวิเคราะห์ข้อมูลใน Big Data โดยใช้ UI แบบกราฟิกและการใช้งานแบบลากและวาง (Drag and Drop) มาเพิ่มความสะดวกให้กับผู้ใช้งานยิ่งขึ้น ไปจนกระทั่งขั้นตอนการจัดทำรายงาน และแสดงผลการวิเคราะห์โดยเน้นการแสดงผลด้วยภาพ (Visualization) ระดับ Dashboard เพื่อให้ได้ผลเชิงภาพรวมที่เข้าใจได้ง่ายและตอบสนองความต้องการทางธุรกิจได้อย่างรวดเร็วด้วย

รายละเอียดของเทคโนโลยีที่สำคัญเพื่อการใช้งานโซลูชัน Big Data

1. ฐานข้อมูล NoSQL ซึ่งเป็นฐานข้อมูลยุคถัดจาก RDBMS มุ่งเน้นการจัดการข้อมูลที่มีพลวัต (Dynamic) มากกว่าข้อมูลที่มีโครงสร้างและมีการจัดเก็บเป็นระบบ ซึ่งก็คือฐานข้อมูลสำหรับ Big Data อย่างแท้จริง เช่น Apache Cassandra ซึ่งมีความสามารถในการอ่านเขียนข้อมูลได้อย่างรวดเร็ว ทำให้สามารถทำงานกับข้อมูลปริมาณมหาศาลของ Big Data ได้ และยังเป็น Open source ด้วย
2. การจัดเก็บข้อมูลปริมาณมหาศาลโดยใช้บริการ Cloud Storage ซึ่งผู้ให้บริการตั้งแต่ระดับนานาชาติอย่าง Amazon หรือ Google ไปจนกระทั่ง ISP และ Vendor ในแต่ละประเทศก็ขยายธุรกิจของตนออกมาครอบคลุมบริการในส่วนนี้ด้วย เพราะในยุคของ Cloud service เช่นนี้ ผู้ที่ให้บริการหนึ่งรายการมักจะได้เป็นผู้ให้บริการรวมทั้งโซลูชัน โดยสามารถได้รายได้จากบริการอื่นๆ ที่เกี่ยวข้องกับ Big Data ติดตามมาทั้งหมดในที่สุดด้วย
3. เทคโนโลยีการประมวลผลแบบ Parallel Processing ที่มาแรงอย่าง Hadoop ซึ่งมีความสามารถในการบริหารจัดการงานประมวลผลโดยกระจายงานออกไปยัง Cloud Processing, ประมวลผล และรวบรวมผลลัพธ์กลับมาให้ผู้ใช้ในระดับที่สามารถรองรับการทำงานด้าน Big Data ได้อย่างสบาย อย่างไรก็ตามด้วยต้นกำเนิดของ Hadoop ที่เป็น Open source ทำให้มีข้อจำกัดในหลายๆ ประเด็น บริษัทผู้ค้าซอฟต์แวร์และโซลูชันหลายรายจึงนำเอาแนวทางของ Hadoop ไปพัฒนาให้มีความเชื่อถือได้ (Reliability) ในการทำงานมากขึ้น เช่น GPFS (General Parallel File System) ของ IBM โดยเพิ่มความยืดหยุ่น, ความปลอดภัย, การสำรองข้อมูล, ความสอดคล้องกันของข้อมูล (Integrity) ฯลฯ ซึ่งจะทำให้การบริหารจัดการ Parallel Processing ที่มีความซับซ้อนและต้องทำงานกับข้อมูลมหาศาลมีความสะดวก และลดความเสี่ยงลงให้สูงที่สุดได้อีกด้วย
4. ระบบที่มีความสามารถในการทำ Load Balancing และป้องกัน fail over ในสเกลระดับ Big Data
5. ความสามารถในการลบข้อมูลซ้ำ (De-duplication), การติดตามโดยอ้างอิง KPI ที่กำหนดและการประมวลผลแบบ Real time
6. ความสามารถในการจัดตารางงาน (Job Scheduling) โดยอิงตามอีเวนต์และเวลา

โดยสรุปการทำ Big Data ที่จะก้าวไปสู่การวิเคราะห์ข้อมูลเชิงลึกอย่าง Data Analytics เพื่อให้ได้ข้อมูลคุณภาพที่พร้อมนำไปใช้ในการวางแผน และดำเนินธุรกิจได้นั้น องค์กรต้องมีความพร้อมตั้งแต่รากฐานหรือโครงสร้างของเทคโนโลยี เช่น ระบบประมวลที่รวดเร็ว เครือข่ายที่รองรับ ไปจนถึงเครื่องมือหรือทูลด้าน Big Data หรือข้อมูลที่จะนำมาใช้ในการวิเคราะห์ และสิ่งสำคัญคือ การมีบุคลากรที่จะต้อง

ชำนาญด้านคณิตศาสตร์ และวิทยาศาสตร์ข้อมูล เพื่อให้การวิเคราะห์ข้อมูลได้มาซึ่งข้อมูลที่มีประสิทธิภาพ และตรงกับความต้องการมากที่สุด

เครื่องมือที่เกี่ยวข้องอื่นๆ ของ Hadoop

1. YARN เทคโนโลยีในการบริหารคลัสเตอร์ ใน Hadoop รุ่นที่ 2 โดยมีนิยามล่าสุดคือ ระบบปฏิบัติการแบบกระจายตัวขนาดใหญ่ (Large-scale Distributed Operating System) สำหรับ แอปพลิเคชันด้าน Big Data บางกรณีก็ถูกเรียกว่า MapReduce 2.0 โดย YARN แยกเอาความสามารถ เรื่องการทำ scheduling และการบริหารทรัพยากรออกมาจากคอมโพเนนต์ด้านการประมวลผลข้อมูล ซึ่งทำให้ Hadoop รองรับการประมวลผลที่หลากหลายและรองรับแอปพลิเคชันได้กว้างขวางมากขึ้น เช่น ทำให้ Hadoop clusters รันการคิวรีแบบอินเทอร์แอคทีฟ และใช้งานแอปพลิเคชันสตรีมมิ่งติดต่อกันไปพร้อมกับแบตช์จ็อบของ MapReduce ได้ด้วย

2. MapReduce ซอฟต์แวร์เฟรมเวิร์กเพื่อใช้ในการเขียนโปรแกรมประมวลผลข้อมูลที่ไม่มีโครงสร้าง ในปริมาณมหาศาลระดับ Big Data โดยผู้พัฒนาจะมีความสะดวกในการเรียกใช้รูทีนไลบรารีของฟังก์ชัน Intra cluster ต่างๆ โดยไม่ต้องเขียนขึ้นมาใหม่ MapReduce รองรับการใช้งานหลายภาษา ซึ่งรวมถึง Java, C++, Python, Perl, Ruby และ C อย่างไรก็ตาม MapReduce รองรับเฉพาะแอปพลิเคชันแบบ Batch Processing เท่านั้น

3. Spark เฟรมเวิร์กสำหรับการวิเคราะห์ข้อมูลด้วยการทำ Parallel processing ของข้อมูลขนาดใหญ่ข้ามคอมพิวเตอร์คลัสเตอร์ ซึ่งด้วยวิธีการดังกล่าวทำให้ Spark สามารถรันจ็อบแบบ In-memory ได้เร็วกว่า MapReduce 100 เท่าและรันจ็อบบนดิสก์เร็วกว่า 10 เท่า นอกจากนี้ฟังก์ชันหลักของเอนจิน ส่วนหนึ่งของ Spark ยังทำหน้าที่ส่วนหนึ่งเป็น API และ Spark ยังมี툴ในการบริหารและวิเคราะห์ ข้อมูลเช่น เอนจิน SQL query, อัลกอริธึมด้าน Machine Learning, ระบบประมวลผลกราฟ และ ซอฟต์แวร์ประมวลผลข้อมูลแบบสตรีมมิ่งด้วย

4. HIVE ซอฟต์แวร์ Data Warehouse แบบโอเพ่นซอร์สสำหรับใช้เพื่อคิวรีและวิเคราะห์ข้อมูล ขนาดใหญ่ที่เก็บใน Hadoop โดย HIVE มีอินเทอร์เฟซที่คล้าย SQL สำหรับทำงานบน Hadoop :ซึ่งช่วยให้ผู้ใช้ไม่ต้องทำโปรแกรมเพิ่มเติมบน MapReduce สำหรับการใช้พีเจอร์ดังกล่าวด้วย

5. PIG เทคโนโลยีที่ทำให้ผู้ใช้มีกลไกระดับสูงสำหรับการทำ Parallel Programming ในการนำจ็อบ ของ MapReduce ไปทำงานบนคลัสเตอร์ Hadoop ซึ่ง PIG ทำให้ผู้ใช้สามารถกำหนดเอ็กซีคิวชันรูทีน สำหรับการคิวรีงานวิเคราะห์ชุดข้อมูลขนาดใหญ่ที่ถูกจัดเก็บกระจายกันอยู่ได้โดยไม่ต้องทำงานระดับโลว์ สีเวลบน MapReduce

ตัวอย่างแนวคิดกลยุทธ์ Big Data เพื่อสำรวจความนิยมของแบรนด์บน Social Media

เป้าหมาย

เพื่อทราบถึงผลตอบรับในแคมเปญการตลาดเพื่อโปรโมตความนิยมในแบรนด์สินค้า/บริการของบริษัทต่อกลุ่มเป้าหมายทางการตลาด

ตัวชี้วัด

1. จำนวน Like บน Facebook Page
2. จำนวนการทวีตที่มี Hashtag ที่กำหนดใน Twitter
3. จำนวนการแชร์ภาพตามกติกาที่กำหนดบน Instagram โดยใช้ Hashtag
4. ความสัมพันธ์ระหว่างคำสำคัญ (Keyword) ที่มีความหมายทางบวกกับแบรนด์ของบริษัท

แนวทางการดำเนินการ

1. ประสานกับส่วนงานการตลาด สื่อสารการตลาดและเอเยนซี
2. นำเอาตัวชี้วัดในข้อ 1-3 มากำหนดการติดตามตรวจสอบในระบบ
3. สร้างโมเดลความสัมพันธ์ระหว่างคำสำคัญ (Keyword) ที่ต้องการและแบรนด์ของบริษัท
4. พัฒนาเครื่องมือหรือใช้เครื่องมือในโซลูชัน Big Data ในการเก็บข้อมูล
5. นำเอาข้อมูลที่ได้มาวิเคราะห์โดยโมเดลที่กำหนดไว้
6. วิเคราะห์หาข้อสรุป และสร้างรายงานจากระบบ

2. การทดลอง

2.1 Big Number Library

- Add Library <http://www.gammon.com.au/Arduino/BigNumber.zip>
- ทดลองรัน File → Example → Big Number → Factorial
- ทดลองรัน File → Example → Big Number → Multiplication
- ทดสอบความเข้าใจด้วยโจทย์ Q1: ปีปีแอบเลี้ยงแมวในโรงเรียน

T0 ลูก = 0 รวม = 6 ตัวเมีย = รวม / 2

T1 ลูก = 6 * ตัวเมีย รวม = รวม + ลูก ตัวเมีย = รวม / 2

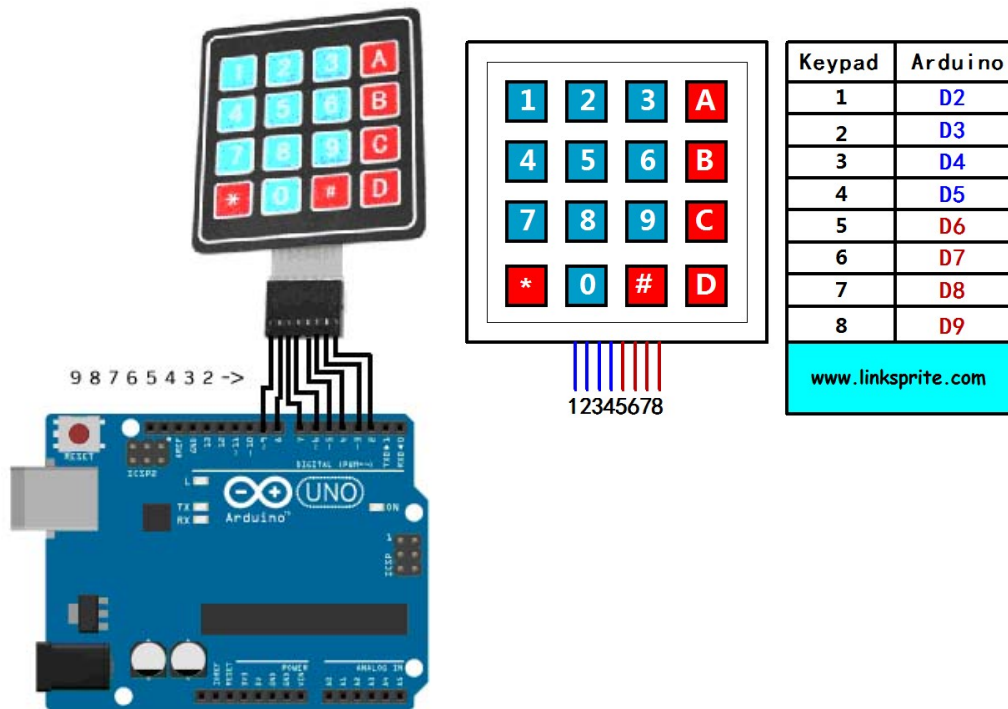
T2 ลูก = 6 * ตัวเมีย รวม = รวม + ลูก ตัวเมีย = รวม / 2

...

	Look	Mae	Total
0		6.00	6.00
1	18.00	6.00	24.00
2	72.00	24.00	96.00
3	288.00	96.00	384.00
4	1,152.00	384.00	1,536.00
5	4,608.00	1,536.00	6,144.00
6	18,432.00	6,144.00	24,576.00
7	73,728.00	24,576.00	98,304.00
8	294,912.00	98,304.00	393,216.00
9	1,179,648.00	393,216.00	1,572,864.00
10	4,718,592.00	1,572,864.00	6,291,456.00
11	18,874,368.00	6,291,456.00	25,165,824.00
12	75,497,472.00	25,165,824.00	100,663,296.00
13	301,989,888.00	100,663,296.00	402,653,184.00
14	1,207,959,552.00	402,653,184.00	1,610,612,736.00
15	4,831,838,208.00	1,610,612,736.00	6,442,450,944.00
16	19,327,352,832.00	6,442,450,944.00	25,769,803,776.00
17	77,309,411,328.00	25,769,803,776.00	103,079,215,104.00
18	309,237,645,312.00	103,079,215,104.00	412,316,860,416.00

2.2 Scan Keypad

- Add Library <http://playground.arduino.cc/uploads/Code/keypad.zip>
- อ่าน <http://www.learningaboutelectronics.com/Articles/Arduino-keypad-circuit.php>
- อ่าน <https://www.arduitronics.com/article/31/arduino-with-keypad-and-4-channel-relay>
- ตัวอย่างจรงดังรูป



- ทดลองรัน File → Example → Keypad → CustomKeypad
- ทดลองรันโปรแกรม

```
// Program 11 - 1 of 3 Keypad4x4
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup() {
  Serial.begin(9600);
}

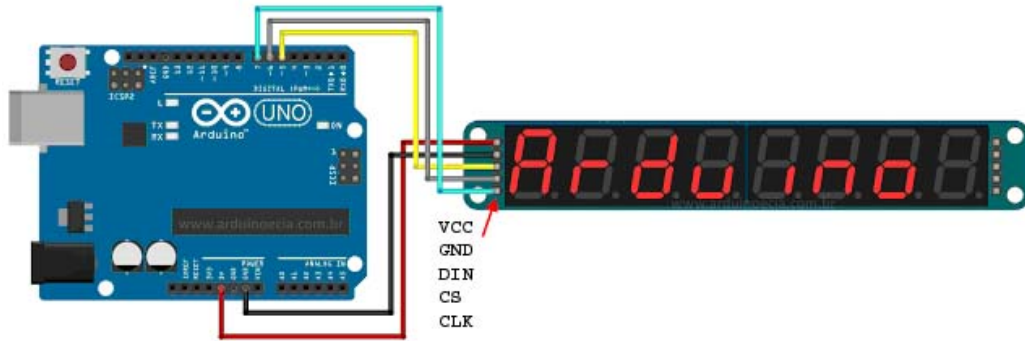
void loop() {
  char key = keypad.getKey();

  if (key) {
    Serial.println(key);
  }
}
```

- ทดสอบความเข้าใจด้วยโจทย์ Q2:

2.3 Scan Display with MAX-7219

- Add Library <https://github.com/wayoda/LedControl/archive/master.zip>
- อ่าน <http://playground.arduino.cc/Main/LedControl>
- อ่าน <http://www.adaline.co.th/product/227/7-segment-8-digi-max7219-module>
- อ่าน <http://blog.circuits4you.com/2016/04/arduino-display-module-7-segment-8.html>
- ตัวอย่างรูป



- ทดลองรันโปรแกรม

```
//
// Program 12 - 2 of 3 Display
//
#include "LedControl.h"

LedControl lc = LedControl(12, 11, 10, 1);

void setup() {
  lc.shutdown(0, false);
  lc.setIntensity(0, 8);
  lc.clearDisplay(0);
}

void loop() {
  { lc.setDigit(0, 7, 1, false);
    lc.setDigit(0, 6, 2, false);
    lc.setDigit(0, 5, 3, false);
    lc.setDigit(0, 4, 4, false);
    lc.setDigit(0, 3, 5, false);
    lc.setDigit(0, 2, 6, false);
    lc.setDigit(0, 1, 7, false);
    lc.setDigit(0, 0, 8, false);
  }
}
```

- จะปรับแก้โปรแกรมอย่างไรเมื่อเราย้ายจากขา 5,6,7 มาเป็นขา 10,11,12

- ต่อวงจร Keypad เพิ่มเข้าไป

- ทดลองรันโปรแกรม

```
//
// Program 13 - 3 of 3 Key4x4 with Display
//
#include "LedControl.h"
#include <Keypad.h>

int Box8 = 0, Box7 = 0, Box6 = 0, Box5 = 0, Box4 = 0, Box3 = 0, Box2 = 0, Box1 = 0;
long unsigned int Result;

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
LedControl lc = LedControl(12, 11, 10, 1); // 12DataIn, 11CLK, 10LOAD

void setup()
{ Serial.begin(9600);
  pinMode(13, OUTPUT);
  lc.shutdown(0, false);
  lc.setIntensity(0, 8);
  lc.clearDisplay(0);
  MAX7219_Display();
}

void loop()
{ char key = keypad.getKey();
  if (key) {
    Serial.println(key);
    if ((key >= '0') && (key <= '9'))
    { Box8 = Box7;      Box7 = Box6;   Box6 = Box5;   Box5 = Box4;
      Box4 = Box3;      Box3 = Box2;   Box2 = Box1;   Box1 = key & 0x0f;
    }

    if (key == '*')
    { Box1 = Box2;      Box2 = Box3;   Box3 = Box4;   Box4 = Box5;
      Box5 = Box6;      Box6 = Box7;   Box7 = Box8;   Box8 = 0;
    }
    lc.clearDisplay(0);

    if (key == '#')
    { Serial.println(Result);
    }

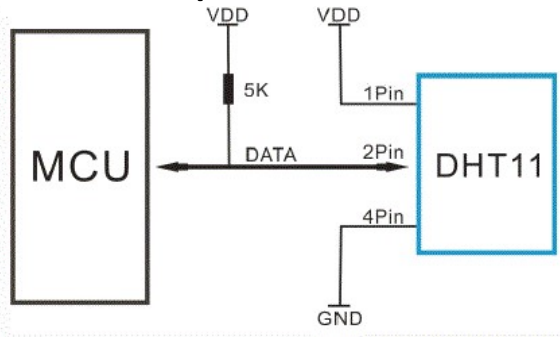
    Result = Box8;
    Result = Result * 10 + Box6;      Result = Result * 10 + Box5;      Result = Result * 10 + Box7;
    Result = Result * 10 + Box4;      Result = Result * 10 + Box3;
    Result = Result * 10 + Box2;      Result = Result * 10 + Box1;
    MAX7219_Display();
  }
}

void MAX7219_Display(void)
{ if (Result > 10000000) lc.setDigit(0, 7, Box8, false);
  if (Result > 1000000) lc.setDigit(0, 6, Box7, false);
  if (Result > 100000) lc.setDigit(0, 5, Box6, false);
  if (Result > 10000) lc.setDigit(0, 4, Box5, false);
  if (Result > 1000) lc.setDigit(0, 3, Box4, false);
  if (Result > 100) lc.setDigit(0, 2, Box3, false);
  if (Result > 10) lc.setDigit(0, 1, Box2, false);
  lc.setDigit(0, 0, Box1, false);
}
```

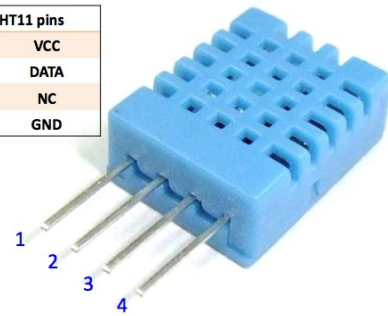
- ทดสอบความเข้าใจด้วยโจทย์ Q3

2.4 DHT-11 Sensors

- Add Library <https://github.com/adafruit/DHT-sensor-library/archive/master.zip>
- อ่าน <https://www.wemos.cc/product/dht-shield.html>
- อ่าน <https://www.arduitronics.com/article/13/การใช้งาน-dht11-humidty-and-temperature-sensor-กับบอร์ด-arduino>
- อ่าน http://www.thaieasyelec.com/information_support/EADN057/EADN057-i%20Module%20-%20SHT11%20Humidity%20and%20Temp%20Sensor%20Kit%20V1.0-QuickStart%20Guide.pdf
- ตัวอย่างจรงดังรูป



DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Wemos DHT Shields {+3.3V, D4, GND} ↔ Arduino {+3.3V, D4, DNG}

- ทดลองรัน File → Example → Big Number → Multiplication
- ทดลองรันโปรแกรมและทดสอบความเข้าใจด้วยโจทย์ Q4

```
//
// Test Progrm 21-1 of 2 DHT11
//
#include "DHT.h"
#define DHTPIN 3 // what digital pin we're connected to
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
}

void loop() {
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\n");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C\n");
  Serial.print(f);
  Serial.print(" *F\n");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C\n");
  Serial.print(hif);
  Serial.print(" *F\n");
  delay(2000);
}
```

- ต่อ MAX7219 Display เพิ่มในวงจร

- ทดลองรันโปรแกรมและทดสอบความเข้าใจด้วยโจทย์ Q5

```
//
// Test Progrsm 22-2 of 2 DHT11 with Display
//
#include "LedControl.h"
#include "DHT.h"
#define DHTPIN 3 // what digital pin we're connected to
#define DHTTYPE DHT11 // DHT 11
int Tempp2 = 7, Tempp1 = 6, Tempp0 = 5;
int Humi2 = 4, Humi1 = 3, Humi0 = 2;
long unsigned int Result;

LedControl lc = LedControl(12, 11, 10, 1); // 12DataIn, 11CLK, 10LOAD
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
  lc.shutdown(0, false);
  lc.setIntensity(0, 8);
  lc.clearDisplay(0);
}

void loop() {
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print("Humidity: ");
  Serial.print(h); Serial.print(" %\n");
  Serial.print("Temperature: ");
  Serial.print(t); Serial.print(" *C\n");
  Serial.print(f); Serial.print(" *F\n");
  Serial.print("Heat index: ");
  Serial.print(hic); Serial.print(" *C\n");
  Serial.print(hif); Serial.print(" *F\n");

  lc.setDigit(0, 7, Tempp2, false);
  lc.setDigit(0, 6, Tempp1, true);
  lc.setDigit(0, 5, Tempp0, false);
  lc.setRow(0, 4, B00001101); // Code=tabcdefgh
  lc.setDigit(0, 3, Humi2, false);
  lc.setDigit(0, 2, Humi1, true);
  lc.setDigit(0, 1, Humi0, false);
  lc.setRow(0, 0, B00010111); // Code=tabcdefgh

  delay(2000);
}
```


3. คำถามท้าทายการทดลอง

Q1: ปิปปิแอบเลี้ยงแมวในโรงเรียนตอนเริ่มเรียน ปี 1 เริ่มแรกมี แมว 6 ตัวครึ่งหนึ่งเป็นตัวเมีย ทุกเทอมแมวตัวเมียทุกตัวจะคลอดลูก 1 ครอก ครอกละ 6 ตัวเสมอ และครึ่งหนึ่งเป็นตัวผู้ ในเวลา 10 ปีที่เรียน ปิปปิจะมีแมวกี่ตัว ให้เขียนโปรแกรมเพื่อหาคำตอบ

Q2: จากการทดลอง Keypad 4x4 อยากได้ชุดอุปกรณ์ ที่ทำการรับค่า 2 จำนวนหลังสุด จากคีย์แพด 4x4 เมื่อกด # ให้นำค่าที่ได้นี้กำหนดการกระพริบของ LED 13 เช่น

- กด 5# ให้กระพริบจำนวน 5 ครั้ง
- กด 123456# ให้กระพริบจำนวน 56 ครั้ง
- กด 13234# ให้กระพริบจำนวน 34 ครั้ง

Q3: จากการทดลอง Keypad 4x4 ให้ต่อ Max-7219 7_Segment จากนั้นนำค่าที่กดจากคีย์แพด 4x4 แสดงที่ Segment หากกด

- หากกด 0-9 ให้ดับเลขไปทางซ้าย
- หากกด * ให้ลบตัวล่าสุดดับเลขไปทางขวา
- หากกด # ให้นำค่าที่ได้นี้กำหนดการกระพริบของ LED 13

Q4: จากการทดลอง DTH-11 อยากได้การแสดงผลแบบนี้

Temperature: 23.73999C / 74.78599F, Humidity: 24.39%
Temperature: 23.72999C / 74.76799F, Humidity: 24.39%
Temperature: 23.71999C / 74.78599F, Humidity: 24.39%

Q5: จากการทดลอง SHT-11 อยากให้แสดงผลอุณหภูมิ (°C) กับความชื้น ที่ MAX-7219 7_Segment

