**NoSQL**

**1. NoSQL ແມ່ນຫຍັງ?**

ຫຼາຍຄົນອາດເຄີຍໄດ້ຍິນນຳກັນ ກ່ຽວກັບເຕັກໂນໂລຢີການຄຸ້ມຄອງຂໍ້ມູນ ໃໝ່ ນີ້, NoSQL ເມື່ອເວົ້າເຖິງ NoSQL, ທ່ານສາມາດໄດ້ຍິນຊື່ຂອງເວບໄຊທ໌ໃຫຍ່. ສະຫນອງໃຫ້ກັບເຊັ່ນFacebook, Twitter, FourSquare, Digg ແລະອື່ນໆ, ພວກເຮົາຮູ້ວ່າ NoSQL ແມ່ນລະບົບຖານຂໍ້ມູນ ສຳ ລັບແອັບພລິເຄຊັນທີ່ຕ້ອງການສະ ໜັບ ສະ ໜູນ ຂໍ້ມູນໃຫຍ່. ຮອງຮັບການຂະຫຍາຍລະບົບງ່າຍແລະອື່ນໆ.

ເຊິ່ງເປັນວຽກທີ່ນ້ອຍ ວິທີການເຮັດມັນ ມັນສາມາດເຮັດວຽກໄດ້ບໍ, ມັນຄຸ້ມຄ່າທີ່ຈະເອົາ NoSQL ໄປປະຕິບັດງານຂະໜາດນ້ອຍ ຫລື Relational Database ຄຳຕອບກໍ່ຄືຂຶ້ນຢູ່ກັບລັກສະນະໃນການນຳມາໃຊ້ງານ ກ່ອນທີ່ຈະຕອບຄຳຖາມວ່າ NoSQL ແມ່ນຄຳຕອບຂອງລະບົບເກັບຂໍ້ມູນຂອງຫຼືບໍ່. ດັ່ງນັ້ນ ພິຈາລະນາຫົວຂໍ້ຕໍ່ໄປນີ້.

**2. ຜູ້ໃຊ້ນັບມື້ນັບຫຼາຍ (BigUsers)**

ມັນສາມາດເຫັນໄດ້ວ່າໃນຊ່ວງເວລາທີ່ຜ່ານມານີ້ ແລະ ດຽວນີ້ຄົນທີ່ໃຊ້ອິນເຕີເນັດມີຫລາຍຂື້ນ. ບໍ່ວ່າມັນຈະຖືກນຳໃຊ້ຜ່ານ Desktop PC ຫລື Smartphone, ເຕັກໂນໂລຢີຂອງອຸປະກອນ (Devices) ມີຫລາຍຢ່າງ ແລະ ງ່າຍຕໍ່ການໃຊ້

ການພັດທະນາລະບົບເພື່ອຮອງຮັບການຈະລາຈອນຂອງແຕ່ລະອຸປະກອນ (Devices) ແມ່ນສິ່ງ ໜຶ່ງ ທີ່ຕ້ອງໄດ້ພິຈາລະນາ ແລະ ບໍ່ພຽງແຕ່ຕ້ອງການຮອງຮັບການເຂົ້າເຖິງຜູ້ໃຊ້ເທົ່ານັ້ນ ແຕ່ພວກເຮົາຕ້ອງໄດ້ຮອງຮັບວິທີການປ້ອນຂໍ້ມູນໃໝ່, ແຕ່ກ່ອນຜູ້ຈັດການເນື້ອຫາຕ່າງໆຄື Web Master, Web Editor, Administrator ເປັນຕົນ, ແຕ່ປະຈຸບັນ ຜູ້ທີ່ປ້ອນຂໍ້ມູນກໍຄືຜູ້ໃຊ້ບໍລິການ (User) ໂດຍກົງຜ່ານອຸປະກອນຕ່າງໆທີ່ມີຫຼາຫຫຼາຍ ແລະ ການປ້ອນຂໍ້ມູນກໍງ່າຍກວ່າແຕ່ກ່ອນ.

ແລະ ຍັງມີປັດໃຈອື່ນໆອີກເຊັ່ນ: ເທດສະການສຳຄັນຕ່າງໆທີ່ຄົນຈະມານຳໃຊ້ຫລາຍ ຫລື ຜູ້ໃຊ້ທີ່ບໍ່ແມ່ນພຽງແຕ່ປະເທດຂອງພວກເຮົາ ເພາະວ່າໂລກອິນເຕີເນັດເຖິງກັນ ອາດຈະຕ້ອງເບິ່ງວ່າລະບົບເຮົາມີຜູ້ເຂົ້າໃຊ້ງານຈາກຕ່າງປະເທດ ຫລື ທົ່ວໂລກບໍ່

ດັ່ງນັ້ນ, ພວກເຮົາຕ້ອງໄດ້ທົບທວນຄືນວິທີການຈັດການຖານຂໍ້ມູນ ທີ່ບໍ່ພຽງແຕ່ຮອງຮັບການເຂົ້າໃຊ້ງານຂອງຜູ້ໃຊ້ບໍລິການເທົ່ານັ້ນ ແຕ່ຕ້ອງຮອງຮັບການເກັບຂໍ້ມູນທີ່ເພີ່ມຂື້ນເລື້ອຍໆ

**3. ປະເພດຂໍ້ມູນຕ່າງໆ ແລະ ຂໍ້ມູນທີ່ຕ້ອງການຈັດເກັບຫຼາຍຂຶ້ນເລື້ອຍໆ (BigData)**

ຈາກຕົວແປຂອງຜູ້ໃຊ້ແມ່ນມີຫຼາຍຂື້ນ ອຸປະກອນຕ່າງໆໃນການເຂົ້າໃຊ້ງານກໍ່ຫຼາກຫຼາຍປະເພດຂອງຂໍ້ມູນທີ່ໄດ້ຮັບຈາກແຕ່ລະອຸປະກອນກໍ່ຫຼາຍປະເພດເຊັ່ນ: ຂໍ້ຄວາມ, ຮູບພາບ, ສຽງ, ວີດີໂອ, ສະຖານທີ່(GeoLocation) ແລະອື່ນໆ, ແລະ ການປ້ອນຂໍ້ມູນເຫຼົ່ານີີແມ່ນງ່າຍດາຍທີ່ສຸດ. ເພາະວ່າເຕັກໂນໂລຢີຂອງຮາດແວແລະຊອບແວ ມີການພັດທະນາຂື້ນເລື້ອຍໆ. ງ່າຍຕໍ່ການໃຊ້ງານ, ສະດວກ ແລະ ໄວຂຶ້ນ.

ດັ່ງນັ້ນ ການຈັດເກັບຮັກສາຂໍ້ມູນທີ່ໄຫຼມາຈາກອຸປະກອນເຫລົ່ານີ້. ພວກເຮົາອາດຈະຕ້ອງວິເຄາະພຶດຕິກຳຂອງຜູ້ໃຊ້. ການສົ່ງເສີມການຕະຫຼາດ ການຕັດສິນໃຈດ້ານການບໍລິຫານ ຂໍ້ມູນຄວາມສຳພັນຂອງລູກຄ້າ ແລະ ອີກຫລາຍໆຢ່າງ ການແນະນຳລະບົບຖານຂໍ້ມູນແບບດັ້ງເດີມ (Relation Database) ອາດຈະບໍເໝາະສົມກັບວຽກງານບາງປະເພດອີກຕໍ່ໄປ.

**4. ເຕັກໂນໂລຢີຮາດແວໄດ້ປ່ຽນແປງ ລາຄາກໍ່ຖືກຫຼຸດລົງແຕ່ວ່າປະສິດທິພາບດີຂື້ນ (Cloud Computing).**

ພວກເຮົາອາດຈະໄດ້ຍິນຄຳ Cloud Technology ຊຶ່ງມີຫລາຍປະເພດ, ແຕ່ນີ້ພວກເຮົາຈະເວົ້າໃນແງ່ຂອງການນຳໃຊ້ມັນ. ປະຈຸບັນນີ້ ຖ້າໃຜໄດ້ທົດລອງໃຊ້ EC2 ຂອງ Amazon, ມັນກໍ່ຈະເປັນທີ່ຮູ້ຈັກກັນດີວ່າ ການທີ່ຈະມີເຄຶ່ອງ Server ທີ່ມີປະສິດທິພາບສູງ ຫຼື ມີ Server 10 ຫຼື 20 ເຄື່ອງທີ່ຈະໃຊ້ເປັນ Database Cluster, ພຽງແຕ່ຄລິກສ້າງ Instance ກໍຈະໄດ້ Serverມາໃຊ້ງານແລ້ວ ແລະ ລາຄາກໍຖືກຫຼາຍ. ຖ້າພວກເຮົາບໍ່ໃຊ້ງານ ກໍ່ສາມາດຍົກເລີກແລະສົ່ງຄືນໄດ້ທັນທີ ເມື່ອປຽບທຽບກັບສະໄໝກ່ອນທີ່ພວກເຮົາຕ້ອງການຈະມີ Server 10 ເຄຶ່ອງ, ຈະຕ້ອງລົງທຶນຊື້ເຄື່ອງຫຼາຍ. ຖ້າບໍ່ໃຊ້ອີກຕໍ່ໄປ ຕ້ອງແບກຫາບພາລະຂອງ Server ເຫລົ່ານີ້ໄວ້ເຊິ່ງມັນເປັນຕົ້ນທຶນທີ່ແພງຫຼາຍ.

ຈາກທີ່ກ່າວມາຂ້າງເທິງນີ້ ພວກເຮົາບໍ່ໄດ້ກຳແໜ້ນ ຂໍ້ດີຂອງ EC2 ໃນທາງໃດທາງໜຶ່ງ, ແຕ່ພວກເຮົາກຳລັງຈະເນັ້ນໜັກເລື່ອງພາບລວມຂອງການນຳໃຊ້ Server ເລີ່ມປ່ຽນແປງໄປຄື ມັນງ່າຍຕໍ່ການໃຊ້, ລາຄາຖືກກວ່າ, ແຕ່ມີປະສິດຕິພາບດີຂື້ນ. ເຊິ່ງເປັນສິ່ງສຳຄັນທີ່ຕ້ອງຄຳນຶງເຖິງເຕັກໂນໂລຢີດ້ານຖານຂໍ້ມູນ ຖ້າຕ້ອງການຈັດເກັບຖານຂໍ້ມູນໃຫຍ່ ຫຼື ຮອງຮັບຜູ້ໃຊ້ງານຈຳນວນຫຼາຍ ການຂະຫຍາຍລະບົບຖານຂໍ້ມູນແມ່ນເປັນເລື່ອງທີ່ງ່າຍຂຶ້ນ. ຊຶ່ງເຮັດໂດຍການເອົາ Server ມາຕໍ່ກັນອອກໄປ ຫຼື ເຊິ່ງເອີ້ນກັນວ່າການຂະຫຍາຍອອກຕາມລວງນອນ (Scale Out), ບໍ່ແມ່ນການຂະຫຍາຍລະບົບຄືເກົ່າ ມັນຄືການຂະຫຍາຍອອກຕາມແນວຕັ້ງ (Scale Up) ແລະຕ້ອງໃຊ້ Server ທີ່ມີປະສິດຕິພາບສູງ. ຊຶ່ງຈະຕ້ອງມີລາຄາແພງຫຼາຍກ່ວາການຂະຫຍາຍອອກຕາມລວງນອນ.

ສະນັ້ນ, ການຂະຫຍາຍລະບົບໂດຍອີງໃສ່ NoSQL ແມ່ນການຮອງຮັບການຂະຫຍາຍອອກຕາມລວງນອນ (Scale Out), ເຊິ່ງຈະແຈກຢາຍຂໍ້ມູນທີ່ຈະເກັບໄວ້ໃນ Server ຫຼາຍຕົວ ແລະໃຊ້ Server ທົ່ວໄປທີ່ເອີ້ນວ່າ (Commodity Server) ບໍ່ຈຳເປັນໃຊ້ Server ທີ່ເອີ້ນວ່າ Enterprise Server ທີ່ມີລາຄາແພງຕາມ Spec ທີ່ສູງກວ່າ ແລະການບໍລິຫານຈັດການກໍມີຄວາມຫຍຸ້ງຍາກຂຶ້ນເຊັ່ນກັນ

**5. ບັນຫາ​ຂອງ Relational Database**

    ຖ້າ​ໃຜ​ທີ່​ໃຊ້​ງານ Relation Database ທີ່​ຕ້ອງ​ການ​ຮອງ​ຮັບ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຂະໜາດ​ໃຫຍ່​ໆ ຄົງ​ຫຼີກລ້ຽງບໍ່ໄດ້ເລື່ອງ​ການເຮັດ Sharding ແລະ Distributed Cache ເພາະ​ເປັນ​ຕັວ​ຫຼັກ​ທີ່​ຕ້ອງ​ເຮັດ ເພື່ອ​ຂະຫຍາຍ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຂອງ Relational Database ໃຫ້​ສາມາດ​ຮອງ​ຮັບ​ຂໍ້​ມູນ​ທີ່​ຫລາຍ​ຂຶ້ນ ແລະ​ຮອງ​ຮັບ​ຈຳນວນ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ລະບົບ​ໄດ້​ຫລາຍ​ຂຶ້ນ

**6. Manual Sharding**

    ການ​ແບ່ງ​ຕາ​ຕະລາງ​ຖານ​ຂໍ້​ມູນ (Table) ອອກ​ເປັນ​ສ່ວນ​ໆ ແລ້ວ​ກໍ່ການ​ກະ​ຈາຍ​ໄປ​ຈັດ​ເກັບ​ໃນ​ຫຼາຍ​ໆ Server  ເພື່ອ​ໃຫ້​ແຕ່​ລະ​ຕາ​ຕະລາງ (Table) ຂອງ​ຖານ​ຂໍ້​ມູນ​ບໍ່​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ຫຼາຍ​ເກີນ​ໄປ ເພາະ​ຖ້າ​ຂໍ້​ມູນ​ໃນ​ແຕ່​ລະ​ຖານ​ຂໍ້​ມູນ​ຫູາຍ​ເກີນ​ໄປ ຈະ​ເຮັດໃຫ້​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຊ້າ​ ແຕ່​ບັນຫາ​ກໍ​ຈະ​ຕາມ​ມາ​ອີກ​ຄື ເມື່ອ​ຕ້ອງ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ອອກ​ໄປ​ໃນ​ແຕ່​ລະ Server ການ​ຈະ​ ​ກັບ​ຂໍ້​ມູນ ເຊັ່ນ ເພີ່ມ, ແກ້​ໄຂ, ລົບ, ດຶງ​ຂໍ້​ມູນ​ມາ​ສະແດງ​ຕ່າງ​ໆ ຈະ​ຕ້ອງ​ເຮັດ​ຜ່ານ Application ຫຼື ​ຕ້ອງ​ມີ Server ບາງ​ໂຕ​ທີ່​​ຖ້າ​ດຶງ​ຂໍ້​ມູນ​ແຕ່​ລະ Server ມາ​ທັງໝົດ​ເປັນ​ກ້ອນ​ດຽວ ນັ້ນ​ໝາຍ​ຄວາມ​ວ່າ ເຮົາ​ຕ້ອງ​ເຮັດ​ດ້ວຍ​ໂຕ​ເຮົາ​ເອງ ບໍ່​ແມ່ນ​ລະບົບ​ຖານ​ຂໍ້​ມູນ​ຈັດການ​ໃຫ້ (Manual Sharding)

**7. Distributed Cache**

    ເມື່ອ​ເຮົາ​ຕ້ອງ​ການ​ໃຫ້​ລະບົບ​ຮອງ​ຮັບ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ຫລາຍ​ໆ​ໄດ້​ນັ້ນ ຖ້າ​ຈະ​ຕ້ອງ​ເຂົ້າ​ມາ​ອ່ານ​ຂໍ້​ມູນ​ຜ່ານ Database ໂດຍ​ກົງ​ມັນອາດຈະ​ຮອງ​ຮັບ​ບໍ່​ໄຫວ ຫຼື ​ເຮັດ​ໄດ້​ຊ້າ ດັ່ງ​ນັ້ນ​ຈະ​ຕ້ອງ​ມີ​ການເຮັດ Cache Layer ຂຶ້ນ​ມາ ຄື​ແທນ​ທີ່​ຈະ​ເຂົ້າໄປ​ອ່ານ​ຈາກ​ຖານ​ຂໍ້​ມູນ​ໂດຍ​ກົງ ກໍ​ໃຫ້​ອ່ານ​ຜ່ານ Cache ກ່ອນ ດັ່ງ​ນັ້ນ​ການ​ອ່ານ​ຂໍ້​ມູນ​ຈາກ Cache ເປັນ​ການ​ອ່ານ​ຈາກ Memory ໂດຍ​ກົງ ເຮັດໃຫ້​ຮອງ​ຮັບ​ປະລິມານ​ການ​ເຂົ້າ​ມາ​ໃຊ້​ງານ​ໄດ້​ຫຼາຍ​ຂຶ້ນ

    ແຕ່​ບັນຫາ​ຄື ການເຮັດ Cache Layer ນີ້ ຮອງ​ຮັບ​ສະເພາະ​ການ​ອ່ານ​ຂໍ້​ມູນ​ເທົ່າ​ນັ້ນ ບໍ່​ຮອງ​ຮັບ​ການ​ຂຽນ​ຂໍ້​ມູນ​ໄດ້ ຖ້າ​ຕ້ອງ​ການ​ຮອງ​ຮັບ​ການ​ຂຽນ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ແລະ​ອ່ານ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ຈຶ່ງ​ເປັນ​ສິ່ງ​ທີ່ Relational Database ບໍ່​ສາມາດ​ຮອງ​ຮັບ​ງານ​ໃນ​ລັກສະນະ ອ່ານ​,​ຂຽນ ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆໄດ້​ດີ​ ແລະ​ ສິ່ງ​ສຳຄັນການເຮັດ Cache Layer ຈະ​ຕ້ອງ​ມີ​ການ​ດູແລຮັກສາ ແລະ​ ໃຊ້ Server ແຍກ​ອອກ​ໄປ​ຕ່າງ​ຫາກ​ອີກ​

    ຈາກ​ຈຸດ​ນີ້​ເອງ ທັງ​ການ​ເຮັດ Sharding ແລະ Caching ເປັນ​ສິ່ງ​ທີ່​ຖືກ​ພັດທະນາ​ຂຶ້ນ​ໃນ NoSQL ເທ​​ກ​ໂນ​ໂລ​​ຢີ  ໂດຍ​ຮອງ​ຮັບ Auto-Sharding ແລະ Integrated Caching ໃນ​​ຕັວ​ເອງ ດັ່ງນັ້ນ​ເຮົາ​ຈຶ່ງ​ໄດ້​ເຫັນ NoSQL ຖືກ​ນຳໄປ​ໃຊ້​ງານ​ກັບ​ລະບົບ​ໃຫຍ່ໆ ເຊັ່ນ Facebook,Twitter, FourSquare, Digg ແລະ​ອື່ນໆ ເພ​​າະ NoSQL ອອກ​ແບບ​ມາ​ເພື່ອ​ຮອງ​ຮັບ​ຄວາມ​ຕ້ອງການ​ງານ​ໃຫຍ່ໆ​ໄດ້​ດີ​ໂດຍ​ສະເພາະ​ຢູ່​ແລ້ວ ແຕ່​ເຖິງ​ຢ່າງ​ໃດກໍ​ຕາມ​ຍັງ​ມີ​ຄຸນສົມບັດ​ອື່ນໆ ທີ່​ໜ້າ​ສົນໃຈ​ໃນ NoSQL ເທ​​ກ​ໂນ​ໂລ​​ຢີ

**8. ຄຸນສົມບັດ​ຂອງ NoSQL Database**

**9. Dynamic Schemas**

    ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຕ່າງ​ໆ ໃນ​ຖານ​ຂໍ້​ມູນ​ແບບ Relational Database ເຮົາ​ຈະ​ຕ້ອງ​ມີ​ການສ້າງ Schema ຫຼື ຮູບ​ແບບ​ຂອງ​ໂຄງ​ສ້າງ​ຕາ​ຕະລາງວ່າ​ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຫຍັງ​ ເມື່ອ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ເພິ່ມເຕີມ​ຕ້ອງ​ປ່ຽນ Schema ພາຍ​ຫຼັງ (Alter-Table) ກ່ອນ​ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຮູບ​ແບບ​ໃໝ່​ໄດ້

    ແຕ່​ໃນ​ປະຈຸບັນ ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ມີ​ການ​ປ່ຽນ​ແປງ​ຕະຫລອດ​ເວລາ ເພາະ​ຄວາມ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຕ່າງ​ໆ ມີ​ຫຼາກ​ຫຼາຍ​​ຂຶ້ນ​ເລື້ອຍ​ໆ ການ​ກຳນົດ​ໂຄງ​ສ້າງ​ຂອງ​ຕາ​ຕະລາງ​ຖານ​ຂໍ້​ມູນ ຫຼື ການ​ຕ້ອງ​ປ່ຽນ​ໂຄງ​ສ້າງ​ຖານ​ຂໍ້​ມູນ​ເລື້ອຍໆ ໂດຍ​ທີ່​ຂໍ້​ມູນ​ຍັງມີ​ຢູ່​ແລ້ວ ເປັນ​ເລື່ອງ​ທີ່​ຍາກ​ຫຼາຍ ​ຫຼື ເຮັດ​ບໍ່​ໄດ້​ເລີຍ ວິທີ​ການ​ຄື​ອາດ​ຕ້ອງ​ແຍກ​ອອກ​ເປັນ​ຕາ​ຕະລາງ​ໃໝ່ ຊຶ່ງ​ເປັນ​ວິທີ​ແກ້​ບັນຫາ​ຊົ່ວ​ຄາວ​ເທົ່າ​ນັ້ນ

    ລະບົບ​ຖານ​ຂໍ້​ມູນ​ແບບ NoSQL  ເຮົາ​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ມີ Schema ທີ່​ຕາຍ​ຕົວ ຫຼື ​ບໍ່​ຕ້ອງ​ມີ Schema ກ່ອນທີ່ຈະ​ຈັດ​ເກັບ​ຂໍ້​ມູນ ຂໍ້​ມູນ​ແຕ່​ລະ​ແຖວ ສາມາດ​ຈັດ​ເກັບ​ໄດ້​ຕາມ​ຕ້ອງ​ການ ຈະ​ເພີ່ມ​ ຫຼື ​ຫຼຸດ ກໍ​ບໍ່​ມີ​ບັນຫາ​ກັບ​ລະບົບ  ເຮັດໃຫ້​ເຮົາ​ສາມາດ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ໄດ້​ຕາມ​ທີ່​ຕ້ອງ​ການ ປ່ຽນ​ແປງ​ໄດ້​ຕະຫລອດ​ເວລາ ສະ​ດວກ​ ແລະ ວ່ອງໄວ

**10. Auto-Sharding**

    ເມື່ອ​ຂໍ້​ມູນ​ມີ​ຂະໜາດ​ໃຫຍ່ ຫຼື ​ເຮົາ​ຕ້ອງ​ການ​ເພີ່ມ​ປະ​ສິດ​ທິ​ພາບ​ການ ອ່ານ​ ແລະ ຂຽນ​ຂໍ້​ມູນ​ປະລິມານ​ຫຼາຍ​ໆ ການເຮັດ Sharding ໃນ​ລະບົບ NoSQL Database ຈະ​ກໍ່ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ໄປ​ຍັງ Server ຕ່າງ​ໆ​ອັດ​ໂນ​ມັດ​​ (Auto-Sharding) ຜູ້​ພັດທະນາ (Developer) ບໍ່​ຕ້ອງ​ຂຽນ​ໂປຣ​ແກຣມ​ໃນ​ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ເອງຄືກັບ Relational Database  
    ການ​ກະ​ຈາຍ​ຂໍ້​ມູນ​ອອກ​ໄປ​ຫຼາຍ​ໆ Server ນີ້​ຍັງ​ເຮັດໃຫ້​ມີ​ຂໍ້​ດີ​ຄື ປະ​ຢັດ​ຕົ້ນ​ທຶນ​ໃນ​ການ​ຂະຫຍາຍ​ລະບົບ ເພາະ​ເປັນ​ການ​ຂະຫຍາຍ​ແບບ​ແນວ​ນອນ (Scale Out) ຊຶ່ງ​ສາມາດ​ນຳ Server ປົກກະຕິ​ທົ່ວ​ໄປ​ມາ​ໃຊ້​ງານ​ໄດ້ ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ເປັນ Enterprise Server

**11. Replication**

    ການ​ສຳ​ເນົາ​ຂໍ້​ມູນ​ຈາກ​ເຄື່ອງ​ໜຶ່ງ​ໄປ​ຢັງ​ອີກ​ເຄື່ອງ​ໜຶ່ງ (Replication) ເມື່ອ Server ໜຶ່ງ​ເສຍ​ຫາຍ ອີກ​ເຄື່ອງ​ໜຶ່ງ​ຈະ​ຂຶ້ນ​ມາ​ເຮັດວຽກ​ແທນ​ທັນ​ທີ ໂດຍ​ຂໍ້​ມູນ​ຂອງ​ແຕ່​ລະ​ເຄື່ອງ​ຈະ​ມີ​ຂໍ້​ມູນ​ຄືກັນ ດັ່ງ​ນັ້ນ Replication ເປັນ​ໜຶ່ງ​ຄຸນສົມບັດ​ທີ່​ຕອບສະໜອງ​ຕໍ່​ການ​ໃຊ້​ງານ​ທີ່​ຕ້ອງ​ການ​ຄວາມ​ຕໍ່​ເນື່ອງ​ໄດ້​ຕະຫລອດ​ເວລາ (High Availability)

**12. Integrated Caching**

    ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ໃຊ້​ງານ​ເລື້ອຍໆ ເຂົ້າ​ໄວ້​ໃນ Memory (RAM) ຊຶ່ງ​ເປັນ​ຄຸນສົມບັດ​ເດັ່ນ​ຂອງ NoSQL ທີ່​ທັງຫມົດ Caching ໄວ້​ໃນ​ຕົວ​ເອງ​ຢູ່​ແລ້ວ ເຮົາ​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ເຮັດ Cache Layer ຄືກັບ Relational Database ອີກ​ຕໍ່ໄປ ທີ່​ຕ້ອງ​ເຮັດ Cache Layer ແຍກ​ຕ່າງ​ຫາກ ​ແລະ​ ເບິ່ງແຍງຮັກສາ​ແຍກ​ອອກ​ໄປ​ຕ່າງ​ຫາກ​ອີກ​ດ້ວຍ

**13. ປະ​ເພດ​ຂອງ​ຖານ​ຂໍ້​ມູນ NoSQL**

    NoSQL ຖືກ​ແບ່ງ​ປະ​ເພດ​ຕາມ​ລັກສະນະ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ແຕກ​ຕ່າງ​ກັນ ດັ່ງ​ນັ້ນ​ການ​ຈະ​ເລືອກ NoSQL Database ໂຕໃດໂຕໜຶ່ງ ຈະ​ຕ້ອງ​ເບິ່ງ​ອີກ​ວ່າ ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ຂອງ​ຖານ​ຂໍ້​ມູນ​ເປັນ​ແບບໃດ ເຊັ່ນ

- Document databases ເຊັ່ນ MongoDB, CouchDB, Elasticsearch

- Graph stores ເຊັ່ນ Neo4J, Infinite Graph, InfoGrid

- Key-value stores ເຊັ່ນ DynamoDB, Redis, MemcacheDB

- Wide-column stores ເຊັ່ນ Cassandra, Amazon SimpleDB, Hadoop / HBase

**14. Open source License**

    ໂດຍ​ສ່ວນ​ໃຫຍ່​ແລ້ວ NoSQL ຈະ​ເປັນ​ລິຂະ​ສິດ​ແບບ Open source ຊຶ່ງ​ບໍ່​ຕ້ອງ​ເສຍ​ຄ່າ​ໃຊ້​ຈ່າຍ​ໃນ​ການ​ນຳ​ມາ​ໃຊ້​ງານ ດັ່ງ​ນັ້ນ​ເຮົາ​ສາມາດ​ນຳ NoSQL Database ແຕ່​ລະ​ຕົວ​ມາ​ຕິດ​ຕັ້ງ​ໃຊ້​ງານ​ໄດ້​ໂດຍ​ບໍ່​ເສຍ​ຄ່າ​ໃຊ້​ຈ່າຍ​ໃດ​ໆ (ຟ​ຣີ)

**15. ນຳ NoSQL ມາ​ໃຊ້​ງານ​ຂະໜາດ​ນ້ອຍ​ໄດ້​ ຫຼື​ ບໍ່**

    ຈາກ​ທີ່​ກ່າວ​ມາ​ແລ້ວ ຄົງ​ພໍ​ຈະ​ຕອບ​ຄຳ​ຖາມ​ນີ້​ໄດ້​ວ່າ ການ​ນຳ NoSQL Database ​ເມື່ອ​ນຳ​ມາ​ໃຊ້​ໃນ​ງານ​ຂະໜາດ​ໃຫຍ່​ນັ້ນ ເໝາະ​ສົມ​ຢ່າງ​ແນ່ນອນ ແຕ່​ຖ້າ​ເປັນ​ລະບົບ​ທົ່ວ​ໆ​ໄປ​ ຄວນ​ຈະ​ນຳ NoSQL ມາ​ໃຊ້​ງານ ​ຫຼື ​ບໍ່

    ຄຳ​ຕອບ​ຄື ຂຶ້ນ​ຢູ່​ກັບ​ລັກສະນະ​ງານ​ວ່າ ເຮົາ​ຈະ​ໃຊ້​ຄຸນສົມບັດ​ຫຍັງ​ຂອງ NoSQL ຖ້າ​ເຮົາ​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ບໍ່​ຕ້ອງ​ຢຶດ​ຕິດ​ກັບ​ໂຄງ​ສ້າງ (Dynamic Schema) ແລະ ​ຕ້ອງ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ລະບົບ​ທີ່​ວ່ອງໄວ (Integrated Caching) ຂໍ້​ມູນ​ອາດຈະ​ຢັງ​ບໍ່​ຫຼາຍ​ເທົ່າ​ໃດ ເຈົ້າ​ອາດຈະ​ໃຊ້ NoSQL ໄດ້​ຢ່າງ​ແນ່ນອນ

    ແຕ່​ຖ້າ​ບໍ່​ຕ້ອງ​ການ​ໃຊ້​ງານ (Dynamic Schema) ບໍ່​ຕ້ອງ​ການ​ເຂົ້າ​ໃຊ້​ງານ​ທີ່​ວ່ອງໄວ (Integrated Caching) ເພາະ​ໃຊ້ Relational Database ກໍ​ເຮັດ​ໄດ້​ດີ​ຢູ່​ແລ້ວ Database Server ກັບ Web Server ກໍ​ຢູ່​ທີ່ Server ດຽວ​ກັນ ຂໍ້​ມູນ​ບໍ່​ຫລາຍ​ນັ້ນ ບໍ່​ຕ້ອງ​ການ​ຈັດ​ເກັບ​ຂໍ້​ມູນ​ທີ່​ເພີ່ມ​ຂະຫຍາຍ​ຂຶ້ນ​ທຸກ​ມື້​ໆ ຜູ້​ເຂົ້າ​ໃຊ້​ງານ​ກໍ​ບໍ່​ໄດ້​ຫລາຍ ເບິ່ງ​ແລ້ວ​ວ່າ​ລະບົບ​​ບໍ່​ມີ​ແນວ​ໂນ້ມ​ຈະ​ຕ້ອງ​ຂະຫຍາຍ​ລະບົບ​ໃນ​ອະນາຄົດ​ອັນ​ໃກ້ ເຈົ້າ​ສາມາດ​ໃຊ້​ງານ Relational Database ໄດ້​ດີ​ຢູ່​ແລ້ວ ຢ່າງ​ບໍ່​ມີ​ບັນຫາ