Tim's DIY NAS

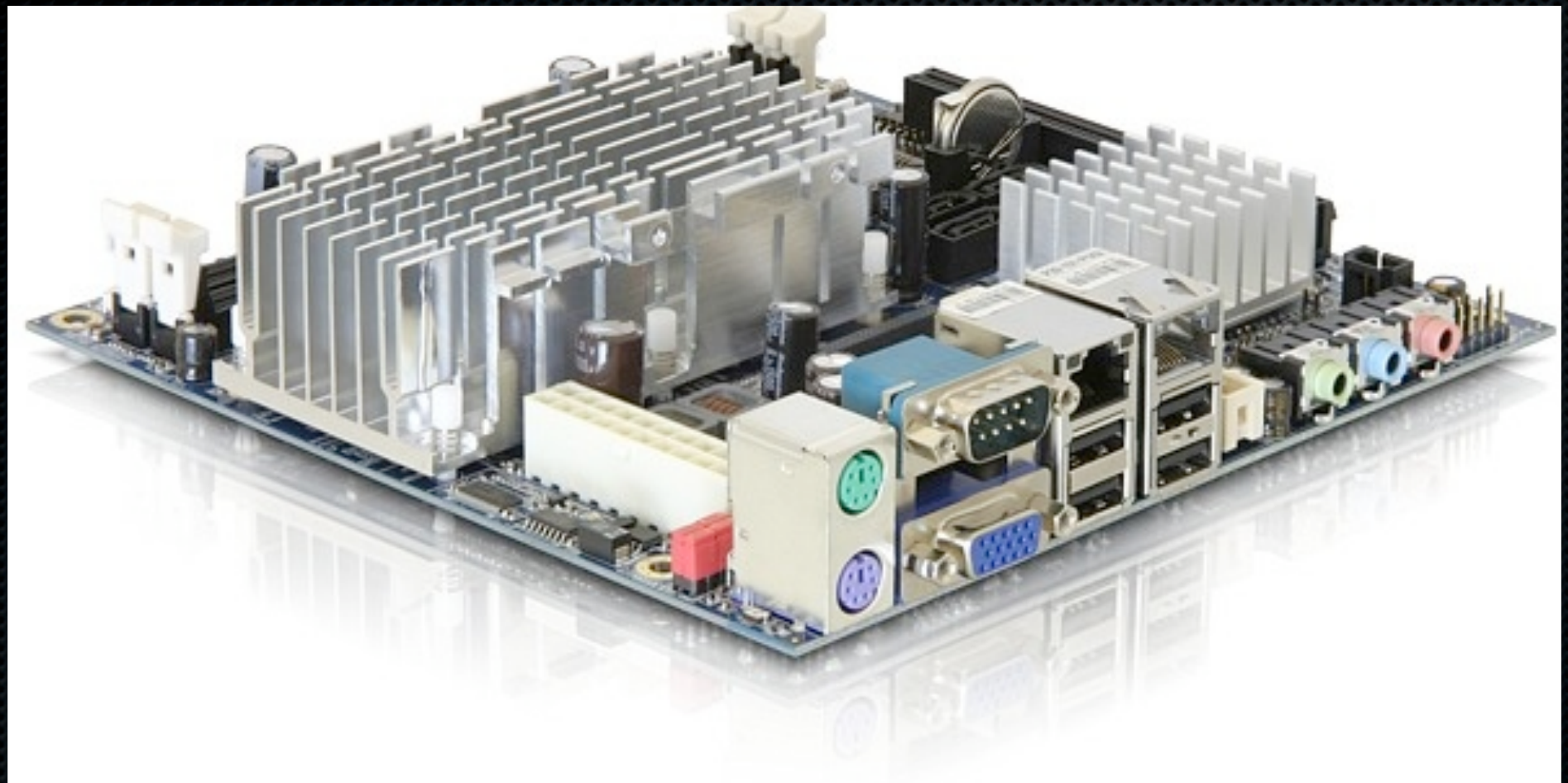
Because it's not like I already have plenty of hobbies already :)

Goals

- Full Linux Distribution Of Choice
- Low Power Consumption
- Large Disk Capacities
- Expandability
- Redundancy
- Configurability / Staggered Upgrades

Current Parts List

- VIA EPIA SN1000EG Motherboard
- 2x Seagate Barracuda LP 1TB Drives
- 1GB Crucial DDR2 Memory (1 stick)
- 2GB Kingston CompactFlash Card
- Antec Tower Case
- Antec 300W PSU



- VIA C7 1GHz CPU (Fanless)
- 4 SATA-II Ports
- Gigabit Ethernet
- CF Slot
- Full-Size PCI Express Slot
- Power & Fan Management

Why a VIA C7?

- Far better power efficiency than typical desktop CPU
- 9W TDP (Pretty good for x86)
  - Mobile variants as low as 3.5W TDP
- Fanless
- I like going for the underdog
  - Intel Atom boards would work just as well if not better

What about ARM?

- Far superior power usage to anyone
- Was unable to find in a standard form-factor
- If you <3 for ARM, look at the Linux-friendly commercial NAS devices
  - qNap, Buffalo, DroboShare

Seagate Barracuda LP

- 1-2TB Capacity
- 3W Avg / 5.7W Operating (1TB)
- 5900 RPM
- 5.1ms Average Latency
- 95MB/Sec Sustained Transfer Rate

Software Configuration

- Ubuntu Linux 9.04
- OS Boots & Runs off CompactFlash
- Software RAID / LVM For Hard-Drives
  - LVM volumes for /home, swap, Time Machine
- File Systems
  - ext2 for OS
  - XFS for everything else
- Backups via Samba and SSH/rsync

Challenges

- Migrating from old system
  - Hardware RAID & SIi3112
- Installing OS onto CF card
- Preventing non-essential writes to CF media
- CPU frequency scaling

Minimizing Writes

The Quick And Dirty Way

- Large Flash Device (2GB+)
  - Avoids needing to put OS on a ramdisk
  - Better Wear Leveling?
- Disable Unnecessary Services
  - apt-get install rcconf
- Mount High Write Directories Onto tmpfs
- Verify Setup By Looking At iostat/dstat
- Probably No Need To Do This With Modern Flash
  - (But I Did Anyway)

Configuring Time Machine

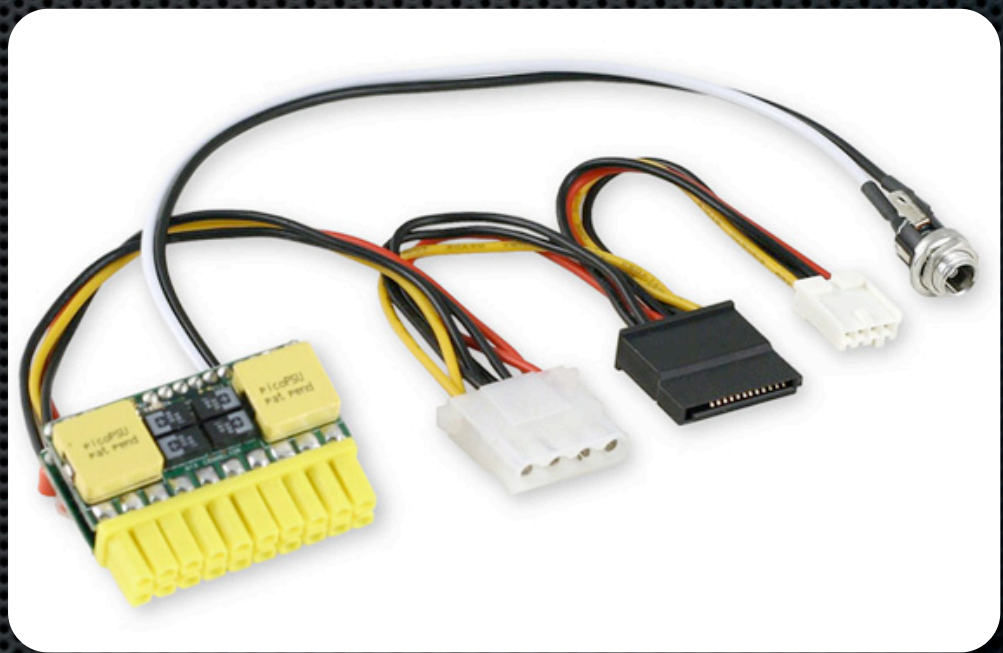
- Create a file-system just for Time Machine
- Create a network share (samba or NFS)
- Enable Unsupported Backups In Time Machine
  - `defaults write com.apple.systempreferences TMShowUnsupportedNetworkVolumes 1`
- Manually Create a Sparse Image & Copy to Share
  - machinename_macaddress.sparsebundle
- Mount Share, Open Time Machine, Select Share

Problems with Time Machine

- There may still be a bug which, if the share gets full, can blow away your backups (yay!)
- Bare metal restores require installing and updating OS X and then using Migration Assistant
- I have not (yet) tried a full-restore
- **Always test your backups**
  - I recommend doing other backups in addition to Time Machine

To-Do

- Setup JungleDisk for Offsite Backups of Critical Data
- External Drive Backups
- VIA C7 Optimized Kernel
- Temperature Monitoring (lm-sensors)
- Multimedia Streaming (mDNS, Bonjour, PS3)
- 4-drive Hot Swap Bay
- Low Power PSU (PicoPSU?)
- Smart UPS/Power Strip with Power Reporting



Demo