

BOREASZ. JAK Z NIEGO KORZYSTAC?

- **OBOWIĄZUJE DAWNY LOGIN I DAWNE HASŁO.** Logowanie z delty analogicznie jak logowanie na notosa: ssh boreasz
- Jeden węzeł = 32 procesory (128 procesorów). Lepiej używać 32 procesory/węzeł
- System operacyjny **AIX** podobny do linux'a, jednak nie taki sam. To ma pewne konsekwencje:
 - ssh -Y nie działa, używamy ssh. Nie potrzeba **module load ...**
 - kompilowanie programów przy pomocy **Makefile**. Jak "zrobić" **Makefile**: patrz niżej
- System kolejkowy: **loadleveller**. Skrypt kolejkowy nieco inny niż dla notosa. Patrz niżej. Obsługa systemu kolejkowego - taka sama jak dla notosa:
 - llsubmit [nazwa skryptu]
 - llq i llq - u [nazwa użytkownika]
 - llcancel [numer w kolejce]

- **SKRYPT KOLEJKOWY dla programu PROGRAM.f**

```
# !/usr/bin/csh
#@ class                =kdm (lub test)
#@ account_no          =G59-2
#@ job_type            =parallel
#@ network.MPI         =sn_all,not_shared,US,HIGH
#@ job_name            =[NĄZWA SKRYPTU]
#@ tasks_per_node     =[liczba procesorow w węźle]
#@ node                =[liczba węzłów]
#@ wall_clock_limit    =hh:mm:ss
#@ environment         =COPY_ALL
#@ output              =OUT
#@ error               =ERR
#@ notification        =never
#@ queue
mpiexec -n [liczba użytych procesorów] ./PROGRAM (>PLIK)
```

- **MAKEFILE**

W katalogu, w którym uruchamia się program **PROGRAM.f** należy utworzyć plik o nazwie **Makefile**, o takiej postaci:

```
FC=mpxlf_r  
OPTS=-q64 -qfixed -qarch=pwr7 -g -O2
```

```
PROGRAM: PROGRAM.o
```

```
$(FC) $(OPTS) -o PROGRAM PROGRAM.o
```

```
PROGRAM.o: PROGRAM.f
```

```
$(FC) $(OPTS) -c PROGRAM.f
```

- Aby skompilować program należy wykonać:

```
gmake<ENTER>
```

- Program wstawiamy do kolejki: `lsubmit [NAZWA SKRYPTU KOLEJKOWEGO]`

- **WAŻNE OSTREŻENIE:** program daje się skompilować również bez Makefile:

```
mpixlf_r -qfixed PROGRAM.f
```

Nawet gdy otrzymamy informacje 'successful' i uruchomimy program, może on działać błędnie!